MATCHING ALGORITHMS

AND FEATURE MATCH QUALITY MEASURES

FOR MODEL-BASED OBJECT RECOGNITION

WITH APPLICATIONS TO

AUTOMATIC TARGET RECOGNITION

MATCHING ALGORITHMS AND FEATURE MATCH QUALITY MEASURES FOR MODEL-BASED OBJECT RECOGNITION WITH APPLICATIONS TO AUTOMATIC TARGET RECOGNITION

Martin Garcia Keller

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science
Courant Institute of Mathematical Sciences
Graduate School of Arts and Science
New York University

May 1999

Approved ________________________

Robert Hummel

Acknowledgements

Support for this research has been provided by the Graduate School of Arts and Science, New York University, and by the Information Systems Office, Defense Advanced Research Projects Agency.

Preface

Needless to say, this work would not have been possible without the continuing support of Robert Hummel and Benjamin Goldberg. To them goes my deepest gratitude.

Table of Contents

Chapter 1. Overview

1.1 Motivation: The Role of Image Understanding

In the fields of computational vision and image understanding, the object recognition problem can often be formulated as a problem of matching a collection of model features to features extracted from an observed scene. This dissertation is concerned with the use of feature-based match similarity measures and feature match algorithms in object detection and classification in the context of image understanding from complex signature data. Our applications are in the domains of target vehicle recognition from radar imagery, and binocular stereopsis.

In what follows, we will consider “image understanding” to encompass the set of activities necessary to identify objects in visual imagery and to establish meaningful three-dimensional relationships between the objects themselves, or between the object and the viewer. The main goal in image understanding then involves the transformation of images to symbolic representation, effectively providing a high-level description of an image in terms of objects, object attributes, and relationships between known objects. As such, image understanding subsumes the capabilities traditionally associated with image processing, object recognition and artificial vision [Crevier and Lepage 1997].

In human and/or biological vision systems, the task of object recognition is a natural and spontaneous one. Humans can recognize immediately and without effort a huge variety of objects from diverse perceptual cues and multiple sensorial inputs. The operations involved are complex and inconspicuous psychophysical and biological processes, including the use of properties such as shape, color, texture, pattern, motion, context, as well as considerations based on contextual information, prior knowledge, expectations, functionality hypothesis, and temporal continuity. These operations and their relation to machine object recognition and artificial vision are discussed in detail elsewhere [Marr 1982], [Biederman 1985], but they are not our concern in this thesis.

In this research, we consider only the simpler problem of model-based vision, where the objects to be recognized come from a library of three-dimensional models known in advance, and the problem is constrained using context and domain-specific knowledge.

The relevance of this work resides in its potential to support state-of-the-art developments in both civilian and military applications including knowledge-based image analysis, sensors exploitation, intelligence gathering, evolving databases, interactive environments, etc. A large number of applications are reviewed below in section 1.4. Experimental results are presented in Chapters 5, 6, and 7.

1.2 Object Recognition as Feature Matching

1.1.1 Introduction

Model-based object recognition techniques involve finding patterns of features within an image, guided by *a priori* models. That is, we wish to identify modeled objects in a scene by relating stored geometric model properties and projection parameters to salient features extracted from sensor data.

The choice of model ***representation*** is thus essential to the process. The representation should be rich enough to allow reliable discrimination and to account for complex variability, and yet terse enough to enable efficient matching. In the end, this question reduces to the selection of stable and discriminative features for use in the matching process. In practice, objects and models are represented as abstract "patterns," or collections of "locally defined" features (see the discussions about features in subsection 1.1.2. and about models in section 1.2.)

Matching is one of the central issues of model-based recognition and an important component of most object recognition systems. A common goal is to project a three-dimensional model in a scene at roughly the correct position, with a similar scale and similar orientation to the object in the scene. In that way, the accuracy of the match between the object model and the image data can be measured and compared, and candidate hypotheses can be validated in a verification stage. Accomplishing this task requires that relevant and comparable information be extracted from both the sensor data as well as the model database.

Traditional object recognition systems comprise the following five stages: (1) model construction, (2) data acquisition, (3) feature extraction, (4) feature matching, and (5) verification. For the most part in this work, we will only be concerned with the matching stage but will briefly review the general paradigm. The discussion about models is deferred to section 1.2.

1.1.2 Data and Features

The meaning of the word “features” is in general highly application-dependent. In this dissertation, a feature is considered to represent the outcome of an event, which is conditioned on some test, which in turn depends on the input data stream. The fact that the test is satisfied means that the feature is present, and feature parameters can then be estimated accordingly from local data. When the data stream is an image, the output parameters will normally include a “location” for the feature. As an example, an edge detector might compute the local gradient magnitude in an image at every pixel. We might declare the feature to be present whenever the gradient magnitude exceeds some threshold. In that case, the feature will consist of the location and the actual gradient magnitude, as well as a discretized angle measure determining the direction of the gradient at that location. The collection of features from the image would then be a listing of all pixels and associated attributes where the gradient magnitude exceeds the threshold.

Since the determination of a feature generally requires that some functional be applied to the input data stream, and that functional is typically stationary (i.e., it is reapplied at each individual "location" in the input stream), the output of such a functional is sometimes called the "feature" information. Indeed, the output information is a feature of the image in the previous sense, where the test is always true. However, we can also view the output information in a different way.

Specifically, if the functional is applied to the entire data stream and no test is performed, then the output is also a data stream. The output data stream can have multiple components at each input datum and so it can be viewed as a vector-valued output. A "feature map" usually refers to a slice of that output data stream, where one component of the vector data has been selected. Clearly, the notions of a feature map, and the notion of a feature as an attributed event that has passed some test are two different things. Unfortunately, the term "feature" can be applied to both of them, and thus leads to confusion. In this work, a feature generally refers to a collection of attributes together with a location, and it is based upon a functional that has passed a test. This is consistent, for example, with the concept of features as “interest operators” [Moravec 1980, 1981].

Therefore, in the realm of image processing, features are typically thought of as a discrete collection of prominent locations in the image supplemented with attribute information. More generally, a feature can be augmented with a vector of components from some heterogeneous multidimensional space describing feature properties and parameters. Hereafter, this is what we mean by a “feature vector” or simply a “feature.”

When viewed as a problem of matching collections of feature vectors, we identify the following three sub-problems within the object recognition problem:

1. Finding correspondences between extracted and model features, after the model has been suitably transformed into the image domain;

2. Computing a score similarity measure based on those correspondences, by means of a measure for the distance between sets of features;

3. Finding the transformation that best superimposes a model view into the image in order to optimize the ultimate score.

We will initially focus on the scoring problem (item 2 above), since the problem of determining the best correspondences depends on the score, and the problem of finding the optimal transformation can be handled through an iterative process of approximation and refinement, or by other transformation-space approaches (subsection 1.1.3.) All of them rely ultimately on a sensible score function.

1.1.3 Match and Search

In general, the matching process in object recognition involves some type of search since there are many unknown factors governing the appearance of an object in a given scene. Search is interesting and important because the potential complexity can be very high. From the algorithmic point of view, most object recognition systems fall into one of two approaches: those that search in the space of features (or correspondences) and those that search in the space of transformations (or parameters.)

When the search takes place over the set of extracted features and recognizable models, the system attempts to determine correspondences between subsets of model and image features and these correspondences should be consistent with the permissible transformations.

Alternatively, if the search takes place in the space of allowed transformations, the system attempts to find a transformation that brings subsets of model and image features in close correspondence with one another. The classical example representative of this approach is the generalized Hough transform, in which object recognition is achieved by recovering the transformation that brings a large number of model features in correspondence with image features. Each transformation is described in terms of a number of transformation parameters, and votes for these parameters are accumulated by hypothesizing matches between subsets of model and image features [Illingworth and Kittler 1988].

Related to the transformation approach is the algorithm known as geometric hashing [Lamdan, Schwartz and Wolfson 1988], [Wolfson and Hummel 1988], [Rigoutsos and Hummel 1992], [Tsai 1993], [Liu and Hummel 1995] which performs a Hough-like transform and an indexing search in a quantized transformation space. The potential transformations are precomputed (in an offline phase) and encoded by transformation-invariant features; and putative correspondences are then used to index (in the online phase) into an appropriate data structure containing the models and transformations, and other (score) information. The indexing keys used are geometric invariants such as coordinates of scene points computed in the coordinate system defined by ordered groups of scene points.

We consider a model as a three-dimensional representation of a physical object that can give rise to an image realization by applying a parametric transformation from some set of allowable transformations followed by a projection in image or feature space. A particular realization of a model can then be viewed as a two-dimensional image. The process of obtaining a model instantiation or realization from a 3D model is known as model prediction.

We will show in Chapter 5 that a simple implementation of geometric hashing can handle a variety of match score functions and correspondence algorithms and can be used to select the best-fit pair of model and transformation.

1.3 The Meaning of Models in Model-Based Object Recognition

Most object recognition research is model-based or model-driven, in that it relies heavily on the use of known geometrical models for the objects of interest. Model vision systems depend on the building of three-dimensional descriptions and representations, and the prediction, description, and interpretation take place largely in three dimensions [Binford 1981]. Experience has shown that for realistic applications, it is not possible to rely on simple image comparison or template matching schemes, but instead it is necessary to incorporate three-dimensional properties of the objects and the scene.

Models are built from the objects themselves with precise geometric primitives, volumetric descriptions and well-defined transformation properties. These geometric entities and transformations are generally used to predict the appearance of features in 2D images based on photometry of the 3D models and the reflectance and illumination properties of the scene, the sensor, and the imaging geometry. For example, at least the following sources of variability in the image must be taken into account by the modeling process: viewing position, illumination, lighting and photometric effects, object setting and context, occlusion and self-occlusion, object articulation, structural composition, texture and shape.

The terminology is confusing due to the fact that sometimes the model patterns in image or feature space are often referred to as “models,” as opposed to model realizations, instantiations, or exemplars.

1.4 Related Work

The problem of object recognition in machine vision has a long history and is extensively reviewed in [Besl and Jain 1985], [Chin and Dyer 1986], [Fischler and Firschein 1987], [Suetens et al 1992] and more recently, in the online bibliographic surveys of Rosenfeld [Rosenfeld 1998]. [Binford 1982] provides a comprehensive documentation of many earlier object recognition systems.

In the remainder of this section, we briefly survey a number of representative systems for object recognition, in order to present a broad picture of the research, as well as to illustrate diverse paradigms for recognition, and the breadth of techniques that are available. We only consider approaches that are related to the research presented in this thesis. While some of these descriptions are limited to specific techniques and implementations, most of them are representative of generic paradigms for image understanding. For example, many of the following strategies fall in the category of hypothesis generation, testing, verification, and refinement. Much of this material is paraphrased from [Suetens 1992] and from [Wolfson 1990]. Finally, we show various specific areas of application in the next section.

Hierarchical Models and Symbolic Constraints. One of the first object recognition systems was ACRONYM [Brooks 1983], [Binford 1982]. The models in ACRONYM were volumetric 3D models based on generalized cones and generalized cylinders, which represent object classes and their spatial relationships. ACRONYM used symbolic constraints to control and effectively prune the search space. Interpretation proceeds by combining local matches of shapes to individual generalized cylinders into more global matches for more complete objects, exploiting local invariance of features and requiring consistency among related families of constraints. ACRONYM incorporates an effective constraint manipulation system, and an online prediction process that finds viewpoint-invariant characteristic features and builds composite object shapes from different generalized cylinders in a manner consistent with the constraints.

Heuristic Tree Pruning and Hill Climbing. The HYPER system [Ayache and Faugeras 1988] is an example of a robust tree-pruning approach. The system is able to identify and accurately locate touching and overlapping flat industrial parts in an image. Object models and segmented image patterns are described by first-degree polynomial approximations to their contours. Matching is performed by a heuristic tree search procedure, where a rigid model contour is iteratively matched to the image pattern segments by successively adding compatible segments to the current partial contour match. At each iteration, a dissimilarity measure is calculated between the active model segment and each image pattern segment. This measure is a weighted sum of three terms: the difference between the orientation of the model segment and the image segment, the Euclidean distance between their midpoints, and the difference between their lengths. The model segment is then matched with the best image segment, that is, the image segment with the minimal dissimilarity. The estimates of the transformation parameters are then updated. For each hypothesis, a quality measure at each iteration of the search measures the length of the identified model relative to the total model length. At the end of the heuristic search procedure, a final verification is done on the hypothesis with the highest quality measure, and the hypothesis is accepted if the quality measure is above a pre-specified threshold.

Perceptual Grouping of Structures. The SCERPO system [Lowe 1987] extended and refined the ACRONYM system, and was able to recognize polyhedral 3D objects from intensity images under perspective transformation, using perceptual grouping of image features and a hierarchical search strategy with backtrack capability. Simpler features such as edges and straight lines are combined into perceptual structures, that is, instances of collinearity, proximity, and parallelism. Next, these primitive relations are combined into larger, more complex structures such as polygonal shapes. These generic structural patterns are then used to limit the search by hypothesizing the location of manufactured parts, which is then backprojected onto the edge data to verify the hypothesis.

Constraint Propagation and Interpretation Trees. The paradigm advocated in [Grimson 1990a] consists on the application of geometric constraints to prune the search for correspondences and arrive at coherent explanations of labeled scenes. The approach performs a depth-first backtracking search of a tree of possible interpretations. At each node, unary and binary constraints on the relative shapes of data and model features are applied to cut off fruitless paths in the tree. Any leaf of the tree reached by the process defines an hypothesis for a feasible interpretation; solving for the pose of the object and verifying that such a pose is consistent with the other components of the interpretation. Other heuristic criteria for search termination can be employed to handle missing and spurious data [Grimson 1990b]. An alternative to the use of binary geometric constraints is the use of attributed features [Hummel 1995], [Liu and Hummel 1995].

Generic Parts Structures and Model-Driven Optimization. The system of Pentland [Pentland 1990] uses a general-purpose "parts" representation to recognize natural 3D objects in range images. Objects are described in terms of shapes of the component parts, which are modeled as deformable superquadrics. A binary image is first obtained by automatic thresholding of texture, intensity, or range data. A set of 2D binary patterns, whose shapes are 2D projections of 3D superquadrics, is then fit to the binary image using template matching. The detected parts are subsequently considered as hypotheses, and a minimum description length criterion is used to select the subset of part hypotheses that best describes the binary image data. Given a segmentation into 2D patterns, the corresponding 3D parts of similar width, length, and orientation are deformed in order to minimize the error between the visible surface of the 3D object and the available range measurements.

Linear Subspaces and Dimensionality Reduction. Numerous statistical pattern recognition approaches have their origin in linear subspace methods. The central idea underlying these methods is to represent images in terms of their projection into a relatively low-dimensional space that captures most of the important characteristics of the objects to be recognized. Examples of this category include linear combination of models [Ullman and Basri 1991], Eigenfaces [Turk and Pentland 1991], and other regular shape descriptors and point distribution models. Many of these schemes use intermediate representations based directly on two-dimensional views rather than explicit 3D models. Other related techniques such as sparse representations, robust metrics, and matching pursuits need to be devised in order to deal effectively with occlusions, cluttered background, and extraneous objects [Liu et al 1996].

Energy Minimization and Active Contour Models. Image contour models called Snakes, are useful for the dynamic specification of shapes as deformable templates using curvilinear features. Curves are implemented as splines that can be deformed under the influence of image constraints to attract them towards features of interest in the data, as well as internal continuity constraints that force them to remain smooth. Both constraints are realized as additive energy fields, and the best compromise between them is achieved by deforming the curve in order to minimize of its total energy [Kass et al. 1987].

Rule-Based Interpretations. In the system described by [Ohta 1985] for outdoor scene analysis, models are semantic networks that contain properties of scene entities and their relational constraints. A rough interpretation –called a plan– is obtained by coarse segmentation and probabilistic relaxation labeling operating in large patches, tentatively merged with surrounding small patches into homogeneous compact regions. A set of heuristic rules operates subsequently on the preliminary patches and the plan, in order to arrive at a detailed interpretation.

Context Driven Recognition and Learning. The Condor system (for context-driven object recognition) from [Strat 1992] incorporates judicious use of contextual information to control all different levels of reasoning, in a hierarchical knowledge-based strategy for recognition of natural scenes. Hypotheses are generated using low-level special-purpose operators whose invocation is controlled by context sets, which explicitly define the conditions and assumptions necessary for successful invocation. Condor produces a 3D interpretation of the environment, labeled with terms from its recognition vocabulary; that is used to update the terrain database for use during the analysis of subsequent imagery. Candidates for each label are ranked using various measures, so that the best ones can be tested first for consistency, detecting and rejecting physically impossible combinations of hypotheses.

Model-Driven Correlation-Based Hypothesis Verification. The 3DPO system [Bolles and Horaud 1986] locates overlapping industrial parts, generating and verifying its hypothesis and refining its pose estimate by backprojecting the prediction onto the range data. The comparison is performed with template matching using correlation, and thus the approach requires rigid objects and detailed models of the physics of data acquisition. The generation of hypotheses is done using maximum clique finding in a relational graph. A high-level strategy uses clusters of features to generate hypotheses, which are then compared with complementary features in a low-level verification and parameter refinement step.

Bipartite Matching and Relaxation Labeling. [Kim and Kak 1991] used bipartite matching together with discrete relaxation to perform recognition of 3D objects from bin parts using the output of an structured light scanner. This represents to our knowledge one of the first attempts to use the technique of bipartite matching in computer vision. Each object is represented by an attributed graph whose nodes are surface features, and whose arcs are edges between the surfaces. Every node in the graph of a scene object is assigned a set of labels for the corresponding model features on the basis of binary similarity criteria. These label sets are then pruned by enforcing relational constraints. If the iterative application of the constraint enforcement leads to a unique labeling, then it provides a consistent interpretation of the scene.

We illustrate a different application of bipartite matching to find an optimal maximum likelihood interpretation of a scene in Chapter 4. Our approach differs from this in that they use bipartite graphs to encode surface features and binary relational constraints between them and they are concerned with complete one-to-one mappings from scene to model features, whereas we use an explicit quantitative score to measure the similarity of matching between point features to arrive at a maximum likelihood or maximum a posteriori interpretation in terms of a particular assignment (Section 3.7).

Geometric Hashing and Affine Invariant Matching. Lamdan, Schwartz and Wolfson [Lamdan et al. 1988a], [Wolfson and Hummel 1988], [Wolfson 1990] present a general and efficient recognition scheme using a transformation invariant hashing scheme. Invariant geometric relations among object features are used to encode model-to-scene transformations using minimal feature subsets as reference coordinate frames in which other features can be represented by their transformation invariant coordinates. The recognition procedure has two major steps. In the first step the representation of the database objects is precompiled in a hash table; this step is executed off-line on the database objects and is independent of the next phase of the algorithm. The second step is executed on the image scene using the hash table for fast indexing and recovering of candidate models and transformations. The hash table serves as an associative memory over the set of all model objects, allowing for retrieval of "similar" feature subsets and hence effectively prunes the space of candidate model features.

Massively Parallel Bayesian Model Matching. The geometric hashing approach discussed above is attractive because it is able to deal with arbitrary groups of transformations, multiple feature types, and is inherently parallel. The main parallelization effort was undertaken by [Rigoutsos 1992], [Rigoutsos and Hummel 1995] who describe scalable algorithms for hypercube SIMD architectures and an implementation in the Connection Machine with similarity and affine invariance. They also introduced Bayesian weighted voting with a Gaussian error model. In this work, we demonstrate how more general scoring functions involving feature correspondences can be implemented in geometric hashing.

Bayesian Hashing and Matching with Attributed Features. A detailed tradeoff error analysis such as those undertaken in [Grimson and Huttenlocher 1990], [Sarachik 1992], shows that minimal feature vectors do not provide adequate discrimination capability in practical systems. The use of features augmented with attribute information can enhance system performance and reduce false alarm rates [Hummel 1995]. Liu and Hummel [1995] used attributed features and Bayesian score functions as a means for false alarm reduction in geometric hashing. They describe a similarity-invariant geometric hashing system that uses line features together with line orientation information, in order to filter candidate matches and significantly reduce the false alarm rate.

Model-Based Bayesian Indexing. A different approach to Bayesian indexing is presented in [Ho and Chelberg 1998] who use local surface groups as index features and statistical estimates of the discriminatory power of each feature. Domain-specific knowledge is compiled offline from CAD models and used to estimate posterior probabilities that define the discriminatory power of features for model objects. In order to speed up the selection of correct objects, object hypotheses are generated and verified in the order of their estimated discriminatory power.

1.5 Applications of Object Recognition

Among the numerous applications of object recognition, we can mention the following:

- **Automatic Target Detection and Recognition.** Automatic Target Recognition (ATR) generally refers to the autonomous or aided target detection and recognition by computer processing of data from a variety of sensors . It is an extremely important capability for targeting and surveillance missions of defense weapon systems operating from a variety of platforms. The major technical challenge for ATR is contending with the combinatorial explosion of target signature variations due to target configuration and articulation, target/sensor acquisition geometry, target phenomenology, and target/environment interactions. ATR systems must maintain low false alarm rates in the face of varying and complex backgrounds, and must operate in real time. The main objective is to locate and identify time-critical targets and vehicles of military interest to aid in surveillance operations, battlefield reconnaissance, intelligence gathering, remote sensing, weapons guidance, and exploitation of imagery from unmanned aerial vehicles and other reconnaissance platforms [Dudgeon et al 1993]. A second application is to look for militarily significant change detection, site monitoring, battle damage assessment and activity tracking. An operational goal is to significantly reduce the volume of imagery presented to a human image analyst. The ATR field has evolved from using statistical pattern recognition approaches to model-based vision, recognition theory, and knowledge-based information exploitation systems. For a recent survey, see [Bhanu et al 1997].

- **Autonomous Robots.** Any mobile robot needs to sense its environment to maintain a dynamic model of the external world and develop an intelligent computational capability for visual processing. The visual analysis of shape and spatial relations is used in many other tasks, such as object manipulation, planning, grasping, guiding and executing movements in the environment, selecting and following a path, or interpreting and understanding world properties [Horn 1986]. Modern home and service robots work in complex environments with complex objects and are able to perform a variety of tasks both indoors and outdoors [Moravec 1998].

- **Vehicle navigation and obstacle avoidance.** This category includes mobile robots as well as autonomous vehicles, smart weapons, and unmanned platforms that navigate through an unknown or partially-known environment. Research in this field has received considerable attention in the past two decades due to the wide range of potential applications, from surveillance to planetary exploration. Autonomous vehicle control, symbolic planning and environment exploration involve the actions of moving around, tracking objects of interest, planning a safe route to avoid collision, servoing to guide motion with respect to road constraints, and integrating sensor information for efficient navigation [Kanade et al 1994].

- **Industrial Visual Inspection.** The traditional examples of object recognition come from the domain of visual inspection of industrial parts. Among the numerous applications we can mention the following: assembly control and verification, metrology, precision measurements of machine parts and electronic patterns, unconstrained material handling, geometric flaw inspection, surface scan and assembly, food processing, quality control, manufacturing, modeling and simulation [International Journal of Machine Vision, Special Issue 1999].

- **Face Recognition.** People in computer vision and pattern recognition have been working in automatic recognition of human faces for more than 25 years [Kanade 1977], [Turk and Pentland 1991]. Recently there has been renewed interest in the problem due in part to numerous security applications ranging from identification of people in police databases to video-based biometric person authentication, and identity verification at automatic teller machines. Numerous commercial systems are currently available. The potential applications include, but are not limited to: video surveillance and monitoring, building security, site monitoring, videoconferencing, law enforcement operations, photo interpretation, medical, commercial and industrial vision. The literature is vast, especially on the web.

- **Medical Image Analysis.** Medical image analysis has developed into an independent flourishing branch of computer vision and image processing as is evidenced by the tremendous interest and growth in the field. Medical imaging concerns both the analysis and interpretation of biomedical images through quantitative mensuration and manipulation of objects, and the visualization of qualitative pictorial information structures [Kalvin 1991]. The main purpose of current research in medical imaging is to improve diagnosis, evaluation, detection, treatment and understanding of abnormalities in internal physiological structures and in their function. The last decades have witnessed a revolutionary development in the use of computers and image processing algorithms in the practice of diagnostic medicine. Images of both anatomical structure and physiological functioning are now produced by a host of imaging modalities: computerized tomography, magnetic resonance imaging, optical sectioning, positron-emission tomography, cryosectioning, ultrasound, thermography and others. This has enabled the acquisition of detailed images carrying vast amounts of multidimensional information. Furthermore, we have seen the appearance and dissemination of online digital libraries of volumetric image data, such as the “Visible Human” project, undertaken by the U.S. National Library of Medicine, which comprises the construction of highly detailed templates for human anatomies from various digital sources. Medical Imaging is one of the most dynamic research fields in action today, and there are regular international conferences and academic journals [IEEE Transactions on Medical Imaging].

- **Optical character recognition.** The importance of document image analysis and optical character recognition has increased markedly in recent years, since paper documents are still the most dominant medium for information exchange, while the computer is the most appropriate device for processing this information. Document image understanding and retrieval research seeks to discover efficient methods for automatically extracting and organizing information from handwritten and machine-printed paper documents containing text, line drawings, graphics, maps, music scores, etc. Its characteristic problems include some of the earliest attempted by computer vision researchers. Document analysis research supports a viable industry stimulated by the growing demand for digital archives, document image databases and paperless sources, the proliferation of inexpensive personal document scanners, and the ubiquity of fax machines. Related areas of research include document image databases, information filtering, text categorization, hand-written document interpretation, document image understanding and retrieval, etc [Kanai and Baird 1998], [International Journal of Computer Vision, Special Issue 1999].

1.6 Scope of this Dissertation

This thesis is about a theory of feature matching in the domain of object recognition. The work arose in the context of a DARPA project on Automatic Target Recognition using synthetic aperture radar (SAR) images. We discuss this application in detail in Chapter 6. In SAR imagery, peaks of the magnitude image form the most salient features, and patterns of peaks (both their spatial arrangements and the relative amplitude at the peak locations) form signatures that are characteristic of the objects. Accordingly, many of the concepts and the terminology used in this thesis have been motivated by this application. The main challenges when dealing with matching of features have to do with the following problems:

- Noise that causes the features not to line up;

Association ambiguity resulting from the noise;

- Missing features, in either the reference or the test pattern;
- Statistics of the underlying processes that determine the features;
- Algorithmic issues in implementation of matching methods.

We confront these issues throughout this thesis, using a Bayesian framework to guide the scoring. The objective is to provide an efficient and effective method for recognizing objects based on extracted features, by matching against large databases of characteristic signature patterns. The database of realizations of the models can be built off-line and stored or alternatively, can be built dynamically as required during a hierarchical search for the matching pattern and the projection parameters. The latter approach uses the notion of “model-based technology” and is the approach that forms the baseline application that motivated this work. However, the issue of prestored models versus dynamic image formation is largely orthogonal to the matching problems considered here.

Our principal contributions are in the form of theory and experimental results comparing multiple feature match similarity measures and matching algorithms. We emphasize the extensibility, robustness, and scalability of our results and document performance statistically by use of rigorous analysis.

This thesis is organized as follows:

- Chapter 1 is a brief outline of the problems presented in this thesis, and in particular reviews the field of object recognition and image understanding in artificial vision.

- Chapters 2 and 3 present an overview of the theory of feature-based matching in object recognition, and discuss the notions of observation models, match similarity measures and match scoring functions. We present a Bayesian formulation for object recognition that relies on maximizing a posterior expected utility to select the best model hypothesis for a given collection of image features.In Chapter 4 we present the problem of matching per se, that is, to determine potential correspondences between model and image features.
- Algorithmic implementation issues are discussed in Chapter 5, especially regarding the efficient implementation of geometric hashing to handle multiple association algorithms and match score functions. The subject of invariance theory is also reviewed briefly.
- In Chapters 6 and 7 we discuss experiments and present experimental results in two application domains respectively.
- In Chapter 8, we conclude with a general summary of results and suggestions for future research work.

Chapter 2. Observed Features

2.0 Overview

The matching process is one of the essential components of the model-based paradigm in image understanding. Here we describe the architecture of the match module in an abstract setting.

The main inputs to the match process consist of two feature sets, namely a model feature set and a data feature set. Feature sets are understood as random vectors that characterize a particular modeled object or as a collection of properties extracted from measured data that represent an observation. A model feature set will be denoted as $\{\mathbf{Y}_i\}_{i=1}^{m}$ and a data feature set as $\{\mathbf{X}_j\}_{j=1}^{s}$. The number of features in these sets can also be considered as realizations of random variables M and S respectively. Model features arise from a *Target Model Hypothesis* defined below. Random vectors $\mathbf{Y}_i$ and $\mathbf{X}_j$ take values in an abstract feature space that contains an appropriate representation of measured characteristics of the objects of interest; these representations should be in some sense isomorphic in order to be able to compare individual model and data features by means of some measure of similarity $d\left(\mathbf{Y}_i, \mathbf{X}_j\right)$. The specification of features includes probability density functions that capture uncertainty and statistical variability in the feature modeling process as well as sensor noise, modeling errors, and other (known or unknown) sources of error. Therefore, in an ideal situation, we would have a precise specification of the feature space with joint probability density functions on all the measurements of interest. As an example, if each feature consists of a pixel *location* in the image plane together with an *amplitude* or grayscale value and if we assume these two are independent of one another, then the inputs could specify two-dimensional Gaussian distributions for the image locations (for both model and data) as well as a Rayleigh distribution for the measured amplitude and a Beta distribution for the model amplitude; in this case we would have two collections of features for model and data respectively, characterized by parameters of the corresponding probability densities. Observe that these features sets are independent of one another and there is in principle no relationship between individual features $\mathbf{Y}_i$ and $\mathbf{X}_j$; in other words, there are no a priori correspondences between feature sets; if necessary, those correspondences need to be found by other means.

The primary output of the match process is a quantitative measure of similarity between the two feature sets. Ideally, this measure would involve some parameterized form of the input in order to permit comparisons between different models. As an example, a generalized likelihood ratio test would require us to compute $\Pr\left(\{\mathbf{Y}_i\} \middle| \{\mathbf{X}_j\}, S, M, \ldots\right)$ but in practice we can only get a rough approximation to this quantity; the main difficulty behind this being the burden of statistical computations required to come up with such a measure. In principle, the measure of similarity doesn't need to be related to a probability — there is a lot of latitude in what is considered to be an acceptable measure of match similarity.

In our work, however, the key to evaluating the quality of a match is a conditional density function, $f\left(\{\mathbf{X}_j\} \middle| \{\mathbf{Y}_i\}\right)$, which is the (differential) probability of observing the feature information $\{\mathbf{X}_j\}$ under a specific hypothesis that yields the prediction $\{\mathbf{Y}_i\}$. This chapter is about means of modeling this conditional density function. In Chapter 3, we discuss the matching theory itself and review different match similarity criteria using this function as a basis for statistical models.

2.1 Introduction

As discussed in Chapter 1, our constrained version of the object recognition problem consists of finding subsets of model instantiations embedded in a given image, subject to invariance under some group of transformations. An instance of a model is a projection of an object's three-dimensional model into an image domain according to parameters of the position of the object and parameters of the sensor imaging system.

For applications in image processing, it is customary to view a model instance as an image "chip," i.e., a small subset of an image, representing a segmentation of the target object from the background. However, in this work, we view a model as being represented by a parameterized mapping from a three-dimensional physical object to a collection of image features, together with information about the statistics of those features conditioned on the occurrence of the model. A collection of features can then be thought as a "pattern," since they form a "dot pattern" when the features are associated with nominal image locations. Pattern points can contain attribute information as well. We refer to such a model, being projected onto the feature space, and to the resulting projected pattern as a ***signature***, which uniquely characterizes an instance or realization of the model.

In the hypothesize-and-test paradigm [Winston 1992], [Binford 1982], instances of a predefined model are supposed to be present in an observed scene and evidence is sought in the sensed image to support or reject each candidate hypothesis. Search strategies are used to guide the hypothesis evidence accrual process, which continues until a decision is made about the identity of particular objects appearing in the image. Alternatively, an empty decision could be advocated, implying that the evidence is not strong enough (in some statistical sense) to support any of the existing model hypotheses. The search process not only identifies the actual objects, but also determines the geometric transformation that best superimposes the model features into the scene. The criterion on which the decision is based is typically a score function derived from local evidence, as defined below and illustrated in Chapter 3.

2.2 An Abstract Formulation of Matching Theory

2.2.0 Background

The essential components of the matching engine that form the core of the matching process, are

1. Discriminative features to measure characteristic attributes in the signatures and statistical models capturing uncertainty and variability of features across the data.

2. An observation generation model, describing how observed features will be generated given the validity of a prediction. This model will give a class conditional density function for the observed features.

3. A feature match quality measure used to compare two individual features, namely a feature predicted from a 3D model against a feature extracted from measured data.

4. A match score function to quantify the global similarity between two feature sets, namely predicted and measured.

5. A search/match decision logic to assess the feasibility of statistical hypotheses over the hypothesis space consisting of (models, parameters, interpretations, others.)

In the rest of this Chapter and in the next Chapter we cover each of these topics as follows:

After a brief discussion of model and image features in 2.2.1, we discuss in subsection 2.2.2 feature generation models and their relation to observed features, feature density functions and feature uncertainty. In the next Chapter in subsection 3.1.1 we describe match score functions and their definition in terms of feature match quality measures which are the subject of subsection 3.1.2. Finally, in subsection 3.1.3 we address search decision logic and search match functionality.

2.2.1 Features

Features are discussed in Section 1.1.2 in Chapter 1. Here we simply recall that features are particular to each application and that feature spaces should be isomorphic in order to be able to compare model features against image features.

Useful features for model-based reasoning should include models of the statistical variability representative of features across the data of interest. This variability is effectively accounted for by a conditional probability density function over the feature space.

Consider two examples. First, in SAR imagery, a feature might be simply the 2D image location of a peak. For an observed feature extracted from an image, we would then have $\mathbf{X}_j = (r_j, c_j)$. For a predicted peak feature, we might have a Gaussian distribution function characterized by a mean location (r_i, c_i) and a covariance C_i. As a second example, every pixel might be a feature, with the attribute of a gray level magnitude. For each observed pixel, the feature information consists of the location and magnitude for the pixel $\mathbf{X}_j = (r_j, c_j, a_j)$. For a predicted pixel, the information consists of the location for the pixel, and a probability density function over the set of possible magnitudes, as in $\mathbf{Y}_i = (r_i, c_i, f_i(\cdot))$.

2.2.2 Model Hypotheses and Interpretation Hypotheses

Features arise on the predicted side from a ***Model Hypothesis*** and on the extracted side from measured signatures and a feature extraction process.

A feature observation generation model provides the link between extracted features and a parametric model of the feature extraction process that allows us to represent joint conditional density functions of specific feature sets. This model provides a unique explanation for how the observed features are generated in terms of the model features.

In the next paragraph we define the concept of model hypothesis and how it relates to the feature generation models.

The Model Hypothesis. In order to specify uniquely an object of interest that can potentially be present in the image, we introduce the concept of a model hypothesis.

Definition. A **model hypothesis** $H \equiv (T, \Theta)$ consists of:

A model T from a fixed but potentially large collection of models, represented as a set of features (see the discussion of features in Section 1.1).

A parametric description Θ of the configuration of the model in three-dimensional space, describing a transformation of the model into its 3D position and orientation.

The purpose of Θ is to specify a transformation of the model into the scene, depending on some number of unknown parameters; these parameters are intended to describe in a unique way the location and orientation of the object, as well as potential articulation, configuration and variations of the original model. We assume that we have detailed models of objects that allow arbitrary configuration states parameterized by Θ.

We will assume that the model hypothesis gives rise to a unique collection of features (a signature) that one expects to see in the observed scene under the assumption that the model is present at the given pose. These features will be denoted by $\mathbf{Y}(T,\Theta) \equiv \mathbf{Y} = \{\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_m\}$. The $\mathbf{Y}_i$ are in fact random vectors with values in an abstract feature space. The random vectors $\mathbf{Y}_i$ are characterized by probability density functions $f_{\mathbf{Y}_i}(\mathbf{y}_i) \equiv f_{\mathbf{Y}_i|T,\Theta}(\mathbf{y}_i|\, t,q) = f_{\mathbf{Y}_i|H}(\mathbf{y}_i|H=h)$ capturing the uncertainties involved in the modeling process and the intrinsic model variability [Hummel 1996a], [Hummel 1996b].

The collection $\mathbf{Y}(T,\Theta) = \{\mathbf{Y}_i\}_{i=1}^{m}$ forms an instantiation of a model pattern in feature space that characterizes the model/parameter pair (T,Θ) in terms of the features of interest. Hereafter, we refer to each $\mathbf{Y}_i$ as a model feature and to the collection of $\{\mathbf{Y}_i\}_{i=1}^{m}$ as a model pattern or model signature.

On the other hand, as a result of the image extraction process there will be a set of extracted features $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_s\}$ representing the evidence obtained from image data. These are also random processes in some appropriate feature space with probability densities $f_{\mathbf{X}_j}\left(\mathbf{x}_j\right)$, and result in observed realizations $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_s$. Note that the number of features s is also an observation, and that the indexing of the image features is performed after they are observed. Typical features in digital images include edges (linear and curvilinear segments), corners or multi-corners (curvature extrema), as well as curvature discontinuities or other prominent landmark points. Such features need to be characteristic of the object and useful for discrimination among the objects of interest.

A predicted model is conceptually represented as a ***set*** of features, as opposed to an ordered tuple of features. Likewise, the image extraction process also produces a ***set*** of features. There is no natural ordering of the features in the sets, even though we have numbered them. As a consequence, some of the $\mathbf{X}_j$ may not correspond to any model feature $\mathbf{Y}_i$ and some $\mathbf{Y}_i$ may not result in an observation $\mathbf{X}_j$ (subsection 2.2.2.1.)

A feature generation model specifies the way in which particular feature realizations can arise from predicted model patterns accounting for uncertainty. As such, it stipulates an algorithmic procedure that models how image features are generated from a hypothesis H and the corresponding model features $\{\mathbf{Y}_i\}$. It is often the case that there might appear implicit correspondences or interpretations in the evaluation of the model. We do not treat the associations as random events nor as nuisance parameters nor hidden latent variables that must be integrated out, but instead we postulate different models that may contain implicit correspondences that should be determined or approximated in order to compute the relevant class conditional densities.

Here we define the notions of interpretation hypothesis and correspondence hypothesis, which will be used in the definition of certain feature generation models.

The Interpretation Hypothesis. Given the assumption that a particular object is present in the scene (i.e., a model hypothesis), a number of different realizations of the model are possible. Each of them is completely characterized by a correspondence hypothesis or by an interpretation hypothesis. We define these concepts below, as we consider the issue of feature correspondences.

Definition. A **correspondence hypothesis** consists of:

(a) A partition $\mathbf{Y} = \mathbf{Y}^{(M)} \bigcup \mathbf{Y}^{(U)}$ of the set of predicted features into those which are actually detected $\mathbf{Y}^{(M)}$, and those which are presumed to be present in the image but are missing because of sensor or measurement errors, inaccurate parameters or unexpected obscuration, namely $\mathbf{Y}^{(U)}$;

(b) A partition of the extracted features into disjoint sets $\mathbf{X} = \mathbf{X}^{(M)} \bigcup \mathbf{X}^{(U)}$, namely those that correspond to explicitly modeled and detected features, and those that are spurious to the model;

(c) A bijection $\Omega : \mathbf{Y}^{(M)} \rightarrow \mathbf{X}^{(M)}$ mapping each detected predicted feature to an associated extracted location in the discretized feature space.

Without loss of generality, and only for notational convenience, let us assume that the features have been labeled in such a way that

$$\mathbf{Y}^{(M)} = \{\mathbf{Y}_1, \mathbf{Y}_2, ..., \mathbf{Y}_n\}, \quad \mathbf{Y}^{(U)} = \{\mathbf{Y}_{n+1}, \mathbf{Y}_{n+2}, ..., \mathbf{Y}_m\},$$

$$\mathbf{X}^{(M)} = \{\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_n\}, \quad \mathbf{X}^{(U)} = \{\mathbf{X}_{n+1}, \mathbf{X}_{n+2}, ..., \mathbf{X}_s\},$$

and

$$\Omega = \{(\mathbf{Y}_1, \mathbf{X}_1), (\mathbf{Y}_2, \mathbf{X}_2), \ldots, (\mathbf{Y}_n, \mathbf{X}_n)\}.$$

More generally, we can define an **Interpretation Hypothesis** by conditions (a), (b) and (c) as above, but where the mapping in (c) is not required to be a bijection and is replaced by an arbitrary binary relation on $\mathbf{Y}^{(M)} \times \mathbf{X}^{(M)}$ (hereby called w.) In this case, the set of model features that are matched, $\mathbf{Y}^{(M)}$, is the domain of w and the set of matched features in the image $\mathbf{X}^{(M)}$, is the range of w. The graph relation induced by w can thus contain arbitrary cycles and cliques of mutually matched features.

For notational simplicity and uniqueness, we will in practice represent w and Ω as mappings and relations between index sets, with the underlying sets $\mathbf{X}$ and $\mathbf{Y}$ being implicitly understood. That is, we will write $(i,j)\in w$, $w(i)=j$ or $\Omega(i)=j$ instead of the more accurate but cumbersome $(\mathbf{Y}_i,\mathbf{X}_j)\in w$, $w(\mathbf{Y}_i)=\mathbf{X}_j$, $\Omega(\mathbf{Y}_i)=\mathbf{X}_j$.

2.2.3 Feature generation models

A feature generation model provides a joint conditional density for the observed feature set given a model feature set. Such a model depends on the specific individual density functions involved and how are they combined to form joint class conditional densities.

We consider only the following examples of feature generation models:

2.2.3.1 Independent associative model with correspondences

In what follows we assume that $g_i(\cdot)$ is a parametric probability density function with location parameter $\mathbf{y}_i$. For example, a Gaussian density function with mean $\mathbf{y}_i$ and covariance matrix Θ_i will be denoted by $g_i(y)=g_i(y-y_i;\Theta_i)$.

In the simplest case of the independent associative model, each model feature $\mathbf{Y}_i$ generates with probability one an observed feature $\mathbf{X}_i$ according to a density $g_i(\cdot)$. We then get the fully independent one-to-one model with no “dropouts.” In this case, the joint density function is given by:

$$f_{\mathbf{X}|H}\left(\mathbf{X}, S \mid H\right) = \Pr_S\left(m\right) \cdot \prod_{x_j \in \mathbf{X}} g_j\left(\mathbf{x}_j - \mathbf{y}_j\, ; \Theta_j\right)$$

Note that the correct interpretation hypothesis in this case pairs each $\mathbf{Y}_i$ with $\mathbf{X}_i$ for $i = 1, \cdots m$, and $m = s$, and that $\mathbf{X}^{(u)}$ and $\mathbf{Y}^{(u)}$ are empty.

More generally, there would be s observations that would be classified in two disjoint subsets of matched features and spurious features and the resulting joint conditional density function could be written as

$$f_{\mathbf{X}|H}\left(\mathbf{X}, S \mid H\right) = \Pr_S\left(s\right) \cdot \prod_{\substack{x_j \in \mathbf{X}^{(M)} \\ (i,j) \in w}} q_i\, g\left(\mathbf{x}_j - \mathbf{y}_i\, ; \Theta_i\right) \cdot \prod_{x_j \in \mathbf{X}^{(U)}} r\left(\mathbf{x}_j\right) \cdot \prod_{y_i \in \mathbf{Y}^{(U)}} \left(1 - q_i\right) \quad (2.2.3.1)$$

Here q_i is the probability that model feature $\mathbf{Y}_i$ is observed, and $r(x_j)$ is the density distribution for spurious observed features. $\Pr_S\left(s\right)$ is the probability that s features will be observed in the image. A simple model for $\Pr_S\left(s\right)$ would be a Poisson distribution on the integer s, namely $\Pr_S\left(s\right) = e^{-\Lambda} \cdot \frac{\Lambda^s}{s!}$ where Λ is the expected number of features observed, which depends on the feature extraction algorithms.

Note that the associations w here are part of the model specification and are inherent to the model. There is no randomness involved in assigning predicted to extracted features, but instead, the computation above gives a density for a particular association.

There is no need to integrate out or sum over correspondences since there is a single association involved and other concepts such as “probability of associations” are meaningless in this model.

Also note that strictly speaking the w in this example need not be a correspondence, but instead can be an interpretation according to the definitions in the previous section.

2.2.3.2 Independent mixture model

In the simplest case, the independent mixture model consists of the following: given m predicted features $\{\mathbf{Y}_i\}_{i=1,\ldots,m}$, each one of s observations $\mathbf{X}_j$ is generated independently according to a weighted mixture density of the form $g(\mathbf{x}) = \sum_{i=1,\ldots,m} w_i \cdot g_i(\mathbf{x} - \mathbf{y}_i)$. In this way, each observation is effectively associated with all of the predictions. Note that with this model, all observations might, with one particular realization, lie near a single y_i. The joint density function is thus:

$$f_{\mathbf{X}|H}(\mathbf{X}, S \mid H) = \prod_{x_j \in \mathbf{X}} \left(\sum_{i=1}^{m} q_i \; g(\mathbf{x}_j - \mathbf{y}_i \, ; \Theta_i) \Big/ \sum_{i=1}^{m} q_i \right)$$

More generally, the observations are divided in two disjoint subsets of matched features and spurious features and the resulting joint conditional density function can be written as

$$f_{\mathbf{X}|H}\left(\mathbf{X}, S \mid H\right) = \Pr_S(s) \cdot \prod_{x_j \in \mathbf{X}^{(M)}} \left(\sum_{i=1}^{m} q_i \; g\left(\mathbf{x}_j - \mathbf{y}_i \, ; \Theta_i\right) \Big/ \sum_{i=1}^{m} q_i \right) \cdot \prod_{x_j \in \mathbf{X}^{(U)}} r\left(\mathbf{x}_j\right) \quad (2.2.3.2)$$

2.2.3.3 Diffusive scattering model

This model has first been proposed by [Irving et al. 1997], [Irving 1997]. Our version of this model has the form

$$f_{\mathbf{X}|H}\left(\mathbf{X}, S|H\right) = \Pr_S(s) \cdot \exp(-R) \cdot \prod_{j=1}^{s} \left(l_0\left(\mathbf{x}_j\right) + \; k \cdot \sum_{i=1}^{m} q_i \cdot g\left(\mathbf{x}_j - \mathbf{y}_i ; \Theta_i\right) \right) \quad (2.2.3.3)$$

In this model each predicted model feature $\mathbf{y}_i$ gives rise to a Poisson-distributed random number of extracted features, and the locations of the extracted features “associated” with a given predicted feature are randomly perturbed away from the nominal predicted feature location; in this case, this perturbation is modeled as an independent random vector with density specified by $g_i\left(\mathbf{x} - \mathbf{y}_i\right)$. Furthermore, additional extracted features are created via a Poisson point process with rate l_0 (which can be spatially varying, as $l_0(\mathbf{x})$). Therefore, each predicted feature has an associated spatial concentration region in which extracted features are assumed to occur according to a non-homogeneous Poisson point process with local rate $m_i = q_i \cdot g_i\left(\mathbf{x} - \mathbf{y}_i\right)$; this, together with the clutter process, results in a compound Poisson process with mean rate

$l = l_0(\mathbf{x}) + k \cdot \sum q_i \cdot g_i(\mathbf{x} - \mathbf{y}_i)$. The density distribution (2.2.3.3) follows [Irving et al. 1997]. The parameter k provides an overall control gain for the target Poisson process and T is a scaling constant related to the area of the region of interest in which features occur.

Analogous to the independent mixture model, this model effectively involves a "mean field" approximation for both the matched and unmatched observations, which accounts for a smoothing of the resulting components. Therefore, the "associations" are implicit in this model and are taken into account by modeling each observation as a realization from a mixture distribution.

2.2.3.4 Strong scattering model

The structure of this model is similar to the diffusive scattering model but the mixture density is replaced by a single density involving only one predicted feature for each generated feature. Each feature commits to a single association. The conditional joint density function is then

$$f_{\mathbf{X}|H}(\mathbf{X}, S|H) = \Pr_S(s) \cdot \exp(-T) \cdot \prod_{\substack{j=1 \\ (i,j)\in\Omega}}^{s} \left(l_0(\mathbf{x}_j) + \quad z_i \cdot h(\mathbf{x}_j - \mathbf{y}_i; \Theta_i) \right) \qquad (2.2.3.4)$$

Each z_i is a Bernoulli random variable with $\Pr(Z_i = 1) = q$ and $h(\cdot)$ is a uniform density function on a disk of fixed radius e [Irving et al. 1997].

2.2.3.5 Bifurcation models

This model is related to the one introduced in (2.2.3.1) but involves a generic association relation and a sum over an arbitrary subset of associations instead of a single association for each observed feature. Then

$$f_{\mathbf{X}|H}(\mathbf{X} \mid H) = \Pr_S(s) \cdot \prod_{\mathbf{X}^{(M)}} \left(\sum_{(i,j) \in w} c_w \cdot g(\mathbf{x}_j - \mathbf{y}_i ; \Theta_i) \right) \cdot \prod_{\mathbf{X}^{(U)}} r(x_j) \cdot \prod_{\mathbf{Y}^{(U)}} (1 - q_i) \qquad (2.2.3.5)$$

This density contains both components from the models (2.2.3.1) and (2.2.3.2).

2.2.3.6 Permanent model

The formula for this model is

$$f_{\mathbf{X}|H}(\mathbf{X}, S | H) = \Pr_S(s) \cdot \sum_{\Omega} \prod_{j=1}^{s} g\left(\mathbf{X}_j \middle| \mathbf{Y}_{\Omega^{-1}\{j\}}\right) \Pr(\mathbf{Y}_i | H_k)$$

$$= \Pr(s) \cdot \mathrm{Perm}(P_k \circ Q_k) \qquad (2.2.3.6)$$

This model involves a full summation over all one-to-one mappings of model features into observed features. Compare with (2.2.3.3) and (2.2.3.5).

The permanent model is related to the so-called “exact point features model” from [Morgan 1992], [Morgan et al. 1996]. The motivation behind the later model involves the integration of correspondences considered as nuisance variables. This model is relevant to the general theory here only insofar as the association is often unknown, and must be approximated by some means. Thus, for observation generation models that incorporate independent associations, one possible way to “guess” the proper association is to sum over all possible associations. Of course, the approximation is only good if all incorrect association contribute little or nothing to the overall score, when summed together.

2.2.3.7 Feature vector models

The concept of feature generation model subsumes the classical statistical theory of pattern recognition as a special case.

By using a random vector distribution $g(\cdot)$ for a single observation, a feature generation model encapsulates arbitrary statistical properties of random ensembles of patterns. The feature space consists of vectors (ordered tuples) of measurements with a given joint distribution of the vector components. A predicted model can be constructed from first principles and prior knowledge of the problem domain or can be learned empirically from existing data.

For a particular example of a feature vector model see the Hausdorff quality metric in section 2.4.

2.2.3.8 Models with stochastic attributes

Features in any of the above models can be enriched with stochastic attributed features to improve discriminability and reduce false alarms [Liu and Hummel 1995]. The density functions are modified by the inclusion of terms based on the density distributions of attributes. For example, the

2.2.3.9 Other hybrid models

Arbitrary combinations of the above models can be built in order to produce generalized feature observation models in an abstract setting. We have restricted our attention to models that involve some sort of independence between observations. This is clearly an oversimplification of realistic conditions but is nevertheless useful to provide an understanding and it has been successful in applications. Observe that this independence assumption doesn't imply that individual components of vector features are independent among themselves and doesn't preclude arbitrary joint distributions of feature components.

2.2.3.10 Clutter generation models

As a special case of a feature generation model, a model for clutter generation provides a formula to account for the appearance of clutter features by means of the clutter densities $f_{\mathbf{X}|H_0}\left(\mathbf{X}, S \middle| H_0\right)$. Here we only discuss the (non-homogeneous) Poisson mixture model.

For applications in traditional image processing, where clutter represents a random ensemble of features originating from non target-like objects, sensor noise, or other sources of error; these can be modeled as a spatial Poisson process with expected density rate of l_0. If the spatial distribution is $g_{H_0}\left(\mathbf{x}; \Theta_0\right)$, then the resulting clutter density distribution will have the form:

$$f_{\mathbf{X}|H_0}\left(\mathbf{X}, S \middle| H_0\right) = \Pr(s) \cdot \prod_{j=1}^{s} g_{H_0}\left(\mathbf{X}_j \middle| H_0\right) \qquad (2.2.3.10)$$

2.3 Summary

Following Section 2.2.0, we have reviewed in this Chapter the concepts of observed features and feature generation models. These models provide formulas for conditional density distributions as functions of the observed features given a hypothesis. In the next

Chapter we continue by considering in turn the following topics: feature match quality measures, feature match scores, and search/match decision logic.

We might note here that some of the formulas for the conditional density functions rely on associations or correspondences between model and observed features. These associations are known by the observation model, but are unknown if we are only given a set of observations. Chapter 4 deals with the problem of reconstructing associations or correspondences under these circumstances.

Chapter 3. Feature Matching

3.0 Overview

In this Chapter, we consider the notions of feature similarity and feature matching, that is, how to assign a statistical measure of likeness between collections of features in the presence of uncertainty. Our point of departure is the discussion in Chapter 2 about the components of the Matching process in object recognition. In particular, we consider the last three components outlined in Section 2.2.0, namely: feature match-quality measures, feature match scores and decision logic. The first two notions deal with the comparison of individual features and of feature sets respectively, and the last notion deals with the algorithmic processes for optimizing feature scores across large sets of hypotheses.

3.1 Deriving Feature Match Scores from conditional density distributions

3.1.1 Feature match score

A feature match score provides a quantitative measure of the global similarity between two feature sets.

The matching score can also be viewed as a global (cost) energy function to be optimized over all possible hypotheses and interpretations. It can alternatively be regarded as a loss function in the context of utility theory. The first formulation is prevalent in physics and thermodynamics applications and has also been popular in the computer vision literature, while the second approach is favored in the Bayesian paradigm in statistical decision theory.

We discuss here two Bayesian approaches for determining a feature match score:

- The posterior odds ratio, based on a maximum a posteriori estimate.
- The maximum likelihood, based on a generalized likelihood ratio test.

3.1.2 Posterior estimates

The essential idea is to compute the posterior odds ratio

$$\Pi_k \equiv \frac{\Pr\left(H_k \middle| \mathbf{X}\right)}{\Pr\left(H_0 \middle| \mathbf{X}\right)}$$

in terms of the individual pairwise “distances” $d(\mathbf{Y}_i, \mathbf{X}_j)$ (Section 3.2) where H_k is the hypothesis under consideration, and H_0 is the reference hypothesis, hereby referred to as the “Clutter Hypothesis.” The matrix of pairwise distances is denoted by $\mathbf{D}$. In practice, we approximate a monotonic function of the resulting Π_k value as the score $S_k(\mathbf{D})$ for hypothesis H_k. The decision logic for determining the correct hypothesis usually attempts to maximize this function of Π_k over all k. A critical component in the computation of Π_k will be the joint class conditional density $f(\mathbf{X} \mid H_k)$, as developed in Chapter 2.

Note that the denominator in the definition of Π_k depends upon the data, so that an appropriate normalization has taken place based on the observed evidence. In many cases, extracted features can be assumed to occur with some frequency even when none of the “target” hypotheses is correct. The clutter hypothesis accounts for these occurrences of clutter features.

The posterior estimate method is similar to the odds formulation of uncertainty reasoning in artificial intelligence [Tanimoto 1995], but rather than computing a ratio of an hypothesis and the negation of the hypothesis, we use a “match hypothesis” and a “reference hypothesis.” The reference hypothesis is based on the notion that the observed features are part of a cluttered background. That is, the observed features do not match any of the candidate target models, but rather match a generic notion of noise or clutter. Of course, this generic notion requires that we have a model of clutter, which is how we compute, in practice, the log probability of the reference hypothesis.

Given a feature match similarity measure $d(\cdot,\cdot)$ and a matrix of match qualities $\mathbf{D} = (d_{ij}) = \{d(\mathbf{Y}_i, \mathbf{X}_j) \mid i = 1 \cdots m, j = 1 \cdots s\}$, our objective is to compute a numerical score for each potential model hypothesis, which will represent the support for that particular hypothesis in view of the evidence extracted from the data. The match score is a functional $S_k(\mathbf{D}) = S_k(\mathbf{D}(\mathbf{Y}, \mathbf{X}))$ of the similarity matrix $\mathbf{D}$ (Section 3.2.)

The score function should be based on pairwise feature affinities as given by $\mathbf{D}$, and it should incorporate as well appropriate information in the form of local and/or global constraints, such as uniqueness or correspondence consistency requirements. What we are really after is a function that resembles a joint density function whose arguments are the feature sets $\mathbf{X}$ and $\mathbf{Y}$. The quantities $\mathbf{D}$ and $S_k(\mathbf{D})$ are then statistics that summarize this joint distribution. As before, in practice $\mathbf{D}$ is often approximated as a family of conditional densities of $\mathbf{X}_j$ given $\mathbf{Y}_i$ for all pairs (i, j).

In the approach that we consider here, the function $S_k(\mathbf{D})$ will be evaluated as a log posterior odds ratio

$$S_k(\mathbf{D}(\mathbf{X},\mathbf{Y})) = \log \Pi_k = \log\left(\frac{\Pr(\mathbf{X}|H_k)}{\Pr(\mathbf{X}|H_0)}\right) + \log\left(\frac{\Pr(H_k)}{\Pr(H_0)}\right) \qquad (3.1.0)$$

which can be computed in terms of the class conditional models for $\mathbf{X}$ and $\mathbf{Y}$ and the clutter models defined in subsection 2.2.2.10.

Observe next that the quantities t can be written as Radon-Nikodym derivatives, i.e., probability density functions. Under fairly general regularity assumptions, they can be expressed as evaluations of density functions at the measured realization of the data $\mathbf{X}_j$, that is $\Pr(\mathbf{X}_j|H_k) = f_{\mathbf{X}_j|H_k}(\mathbf{X}_j|H_k)$ where this quantity represents the class-conditional probability density function of the extracted feature $\mathbf{X}_j$ obtained from a feature generation model. This fact is a consequence of the Radon-Nikodym theorem [Halmos 1950]. The essential property required is the absolute continuity of the density functions. The subtlety consists in that probabilities are usually computed on continuous events, such as $\lfloor \mathbf{x}_j \le \mathbf{X}_j \le \mathbf{x}_j + e \rfloor$, rather than discrete evidence, such as $\lfloor \mathbf{X}_j = \mathbf{x}_j \rfloor$.

Specifically, if the probability measure induced by $\mathbf{X}_j$ on the Borel sets in $\Re$ (assuming for simplicity that $\mathbf{X}_j$ is a real-valued random variable)

$$m_{\mathrm{X}}(\mathrm{B}) = \Pr\lfloor \mathbf{X}_j \in \mathrm{B} \rfloor \qquad \mathrm{B} \in \Im\mathbf{B}$$

is absolutely continuous with respect to the Lebesgue measure l on the real line (which is written as $m_{\mathrm{X}} << l$), then the Radon-Nikodym derivative of m_{X} with respect to l exists and is denoted as $f_{\mathrm{X}} = dm_{\mathrm{X}}/dl$, and satisfies almost everywhere (l) the following identity

$$m_{\mathrm{X}}(\mathrm{B}) = \int_{\mathrm{B}} f_{\mathrm{X}} \cdot dl \quad \forall\, \mathrm{B} \in \Im\mathbf{B}$$

Here $f_{\mathrm{X}}(\cdot)$ is a l -measurable function defined on all Borel sets in the real line, and in particular when $\mathrm{B} = (-\infty, x]$ we have,

$$F_{\mathbf{X}_j}(x) = m_{\mathbf{X}_j}((-\infty, x]) = \int_{-\infty}^{x} f_{\mathrm{X}}(u) \cdot du \ .$$

Therefore, f_{X} can be formally identified with the density function for $\mathbf{X}_j$.

This fact will allow us to rewrite (3.1.1) in terms of the class conditional densities given by a particular observation generation model

$$S_k(\mathbf{D}(\mathbf{X},\mathbf{Y})) = \log \Pi_k = \log\left(\frac{f_{\mathbf{X}|H_k}(\mathbf{X}|H_k)}{f_{\mathbf{X}|H_0}(\mathbf{X}|H_0)} \right) + B_k \qquad (3.1.1)$$

where B_k is the prior odds bias $B_k = \dfrac{\Pr(H_k)}{\Pr(H_0)}$, which is independent of $\mathbf{X}$.

3.1.3 Maximum likelihood estimates

A generalized likelihood ratio test is based on maximizing the following log-likelihood ratio

$$\Lambda_k \equiv \frac{\log f_{\mathbf{X}|H_k}\left(\mathbf{X}\middle|H_k\right)}{\log f_{\mathbf{X}|H_0}\left(\mathbf{X}\middle|H_0\right)} \tag{3.1.2}$$

The objective is to maximize Λ_k over the space of all model hypothesis $H_k = \left(T_k, \Theta\right)$ and over all interpretations w. In order to understand the dependency of (3.1.2) on the models, parameters, and interpretations we rewrite it for the independent associative model with Gaussian density functions as

$$\begin{aligned}\Lambda_k &\equiv \frac{\log g_{\mathbf{X}|H_k}\left(\mathbf{x}\middle|H_k\right)}{\log g_{\mathbf{X}|H_0}\left(\mathbf{x}\middle|H_0\right)} \\ &= \frac{K_1 - \frac{1}{2}\sum_{(i,j)\in w}\left(\mathbf{x}_j - m_i\right)^T \Sigma_i^{-1}\left(\mathbf{x}_j - m_i\right)}{K_0 - \frac{1}{2}\sum_{j}\left(\mathbf{x}_j - m_0\right)^T \Sigma_0^{-1}\left(\mathbf{x}_j - m_0\right)}\end{aligned}$$

Therefore, maximum likelihood estimation involves not a simple linear discriminant test but an optimization over a large space of correspondences (ω) and parameters.

Given our simplified models, the likelihood scoring function (3.1.2) is functionally similar to the posterior score in (3.1.1) except for the prior bias term and the relative argument of the logarithmic function. Indeed, since a prior probability for the likelihood of a complex hypothesis is often impossible to obtain, the maximum likelihood estimate is in some sense preferable, since it doesn't require such priors. Of course, their use is equivalent to assuming that all priors are equal. In the following we consider instead a variation of the former estimate, which is actually just the log of the ratio of likelihoods, not the log-likelihood ratio (compare with 3.1.1 and 3.1.2):

$$S'_k(\mathbf{D}(\mathbf{X},\mathbf{Y})) = \log \Pi'_k = \log\left(\frac{f_{\mathbf{X}|H_k}(\mathbf{X}|H_k)}{f_{\mathbf{X}|H_0}(\mathbf{X}|H_0)}\right) \tag{3.1.3}$$

We use primes to distinguish these scores from the corresponding posterior estimates (3.1.1). Our next task is to rewrite the class conditional densities $f_{X|H_k}(X|H_k)$ in terms of individual "distances" $d(\mathbf{Y}_i, \mathbf{X}_j)$, ie., feature match similarities. We define possible feature match-quality measures in the next section.

3.2 Feature match quality

In the next section, we apply the feature generation models from section 2.2.2 as we identify functions of probability densities $g_i(\mathbf{x}-\mathbf{y})$ with similarity measures between individual features $\mathbf{Y}_i$ and $\mathbf{X}_j$.

Given a model feature $\mathbf{Y}_i$ and an extracted feature $\mathbf{X}_j$, a feature match quality measure is a function $d(\mathbf{Y}_i, \mathbf{X}_j)$ with the following property:

The function $d(\,\cdot\,, \mathbf{X}_j)$ takes a unique global maximum over $\mathbf{Y}_i$ that occurs whenever the event $\lfloor \mathbf{Y}_i = \mathbf{X}_j \rfloor$ takes place.

In general the function $d(\cdot,\cdot)$ will depend on the statistics of the random aggregates $\mathbf{X}$ and $\mathbf{Y}$, that is, on the probability densities $g_{\mathbf{X},\mathbf{Y}}(\cdot,\cdot)$**,** $g_{\mathbf{X}}(\cdot)$ and $g_{\mathbf{Y}}(\cdot)$, as well as in the actual realizations $\mathbf{x}$, $\mathbf{y}$. In practice it will often be the case that the features are decomposed in such a way that

$$d(\mathbf{Y}_i, \mathbf{X}_j) = \log \int g_{\mathbf{X}_j|\mathbf{Y}_i}(\mathbf{x}|\mathbf{y}) \cdot g_{\mathbf{Y}_i}(\mathbf{y}) \cdot d\mathbf{y} \qquad (3.2.1)$$

which is independent of the realization of $\mathbf{Y}_i$, whenever the quantities involved are well defined. Another possible choice would simply be

$$d(\mathbf{Y}_i, \mathbf{X}_j) = \log g_{\mathbf{X}_j|\mathbf{Y}_i}(\mathbf{x}|\mathbf{y}) g_{\mathbf{Y}_i}(\mathbf{y}). \qquad (3.2.2)$$

In the first case, d is called the “log predictive density” of $\mathbf{X}_j$ given $\mathbf{Y}_i$, and $\mathbf{Y}_i$ provides parameters for the model. In the second case, the density is conditioned on a prototype realization $\mathbf{Y}_i = \mathbf{y}_i$. Alternatively, we could use a maximum likelihood estimate $\tilde{\mathbf{Y}}_i(\tilde{\mathbf{y}}_i)$. The conditional densities should be understood in the Radon-Nikodym sense.

Furthermore, often the distance measure $d(\mathbf{Y}_i, \mathbf{X}_j)$ can be interpreted as some variant of a localized energy function that is minimized when $\mathbf{X}_j$ and $\mathbf{Y}_i$ are coincident.

For example, when the priors on $\mathbf{Y}_i$ are Gaussian and the pairwise conditionals of $\mathbf{X}_j$ given $\mathbf{Y}_i$ are also Gaussian we have,

$$g_{\mathbf{Y}_i|H_k}\left(\mathbf{Y}_i \mid H_k\right) = G\left(\mathbf{y}_\mathrm{i}\ \text{-ì}\ {}_\mathrm{i};\text{Ó}_i\right)\ \text{and}\ g_{\mathbf{X}_j|\mathbf{Y}_i}\left(\mathbf{X}_j \mid \mathbf{Y}_i\right) = G\left(\mathbf{x}_j - \mathbf{y}_i;\text{Î}\ {}_j\right)$$

where $G(\ )$ is the Gaussian density function given by $G(\mathbf{u};\textbf{Ó}) = (2p)^{-1}|\textbf{Ó}|^{-1/2}$ $\exp\left(-\frac{1}{2}\mathbf{u}^T\textbf{Ó}^{-1}\mathbf{u}\right)$. Therefore,

$$\begin{aligned} g_{\mathbf{X}_j|H_k,\Omega}\left(\mathbf{X}_j \mid H_k,\Omega\right) &= \int g_{\mathbf{X}_j|\mathbf{Y}_i,\Omega}\left(\mathbf{X}_j - \mathbf{Y}_i \mid \mathbf{Y}_i,\Omega\right)\cdot g_{\mathbf{Y}_j|H_k}\left(\mathbf{Y}_i - \text{ì}\ {}_i \mid H_k,\Omega\right) d\mathbf{Y} \\ &= G\left(\mathbf{X}_j - \text{ì}\ {}_i;\text{Ó}_i + \text{Î}\ {}_j\right) \end{aligned}$$

In this case, the resulting quality measure yields a negative metric

$$d\left(\mathbf{Y}_i, \mathbf{X}_j\right) = A - \tfrac{1}{2}\log\left|\ \textbf{Ó} + \textbf{Î}\ \right| - \tfrac{1}{2}\left\|\ \mathbf{X}_j - \textbf{ì}\ {}_i\right\|^2_{\textbf{Ó}+\textbf{Î}}$$

and is related to a multidimensional Mahalanobis distance, namely $\left\|\mathbf{u}\right\|_{ó}^{2} = \mathbf{u}^{T}\,\mathbf{Ó}^{-1}\mathbf{u}$. This is the case also for any distribution in the exponential family. Note that $d(\cdot,\cdot)$ depends on $\mathbf{Y}_i$ only through the parameters $\mathbf{ì}_{\ i}$ and $\mathbf{Ó}_i$.

The above example illustrates the convolution argument involving the convolution of two Gaussian uncertainties. In other words, the model and observed uncertainty measures (represented here by covariance matrices) just add up in this case to give a global joint uncertainty matrix $\mathbf{Ó}_i + \mathbf{Î}_{\ j}$. This represents the Bayesian conjugate Gaussian pair, that is, the posterior function corresponding to a Gaussian prior (with covariance $Ó_i$) and a Gaussian likelihood (with covariance $Î_{\ j}$) is itself a Gaussian whose covariance is given by $\mathbf{Ó}_i + \mathbf{Î}_{\ j}$.

3.3 Feature match scores in terms of feature match quality measures

In this section, we use the observation generation models of Chapter 2 to write joint conditional density functions, which can then be used in the posterior estimates (or the maximum likelihood estimates) to provide match scores, that can therefore be formulated in terms of the pairwise feature "distances" $d\left(\mathbf{Y}_i, \mathbf{X}_j\right)$. We only provide the modified posterior estimates (3.1.3) here, and we use a spatial Poisson background clutter model. Authentic posterior estimates would require a modification based on prior probabilities, and other normalization methods are possible.

3.3.1 Independent associative model with correspondences

$$\begin{aligned}
\mathbf{S}'(\mathbf{D}(\mathbf{X},\mathbf{Y})) &= \log\left\{\prod_{\substack{x_j\in\mathbf{X}^{(M)}\\(i,j)\in w}} q_i\,\frac{g\left(\mathbf{x}_j-\mathbf{y}_i\,;\Theta_i\right)}{g_0\left(\mathbf{x}_j\right)}\cdot\prod_{x_j\in\mathbf{X}^{(U)}}\frac{r\left(\mathbf{x}_j\right)}{g_0\left(\mathbf{x}_j\right)}\cdot\prod_{y_i\in\mathbf{Y}^{(U)}}\left(1-q_i\right)\right\}\\
&= \sum_{\substack{\mathbf{X}^{(M)}\\(i,j)\in w}}\left(\log q_i+\log\frac{g\left(\mathbf{x}_j-\mathbf{y}_i\,;\Theta_i\right)}{g_0\left(\mathbf{x}_j\right)}\right)+\sum_{\mathbf{X}^{(U)}}\log\frac{r\left(\mathbf{x}_j\right)}{g_0\left(\mathbf{x}_j\right)}+\sum_{\mathbf{Y}^{(U)}}\log\left(1-q_i\right) \qquad (3.3.1)\\
&= \sum_{w} d_{ij}+\sum_{\mathbf{X}^{(U)}} b_j+\sum_{\mathbf{Y}^{(U)}} a_i
\end{aligned}$$

where the *a*'s, *b*'s, and *d*'s stand for the corresponding terms in the previous equation. The *d*'s correspond to normalized match-quality measures:

$$d_{ij} \equiv d\left(\mathbf{Y}_i, \mathbf{X}_j\right) = \log\left(q_i \cdot \frac{g\left(\mathbf{x}_j - \mathbf{y}_i ; \Theta_i\right)}{g_0\left(\mathbf{x}_j\right)} \right) \tag{3.3.1'}$$

In Chapter 4 we review again this model, and we present an algorithm to solve the resulting search optimization problem (Section 3.4) as a bipartite assignment problem.

3.3.2 Independent mixture model

$$\begin{aligned}
\mathbf{S}'\left(\mathbf{D}\left(\mathbf{X},\mathbf{Y}\right)\right) &= \log\left\{ \prod_{x_j \in \mathbf{X}^{(M)}} \left(\sum_{i=1}^{m} q_i \, \frac{g\left(\mathbf{x}_j - \mathbf{y}_i \,; \Theta_i\right)}{g_0\left(\mathbf{x}_j\right)} \Big/ \sum_{i=1}^{m} q_i \right) \cdot \prod_{x_j \in \mathbf{X}^{(U)}} \frac{r\left(\mathbf{x}_j\right)}{g_0\left(\mathbf{x}_j\right)} \right\} \\
&= \sum_{\mathbf{X}^{(M)}} \log\left(\sum_{i=1}^{m} q_i \, \frac{g\left(\mathbf{x}_j - \mathbf{y}_i \,; \Theta_i\right)}{g_0\left(\mathbf{x}_j\right)} \Big/ \sum_{i=1}^{m} q_i \right) + \sum_{\mathbf{X}^{(U)}} \log \frac{r\left(\mathbf{x}_j\right)}{g_0\left(\mathbf{x}_j\right)} \qquad (3.3.2) \\
&= \sum_{\mathbf{X}^{(M)}} \log\left(\sum_{i=1}^{m} \exp\left(d_{ij}\right) \right) + \sum_{\mathbf{X}^{(U)}} b_j
\end{aligned}$$

This model is considered in [Ettinger et al 1996]. The “pseudo-distances” *d* use here are

$$d_{ij} = \log\left(\frac{q_i}{\sum q_l} \cdot \frac{g\left(\mathbf{x}_j - \mathbf{y}_i ; \Theta_i\right)}{g_0\left(\mathbf{x}_j\right)} \right) \tag{3.3.2'}$$

3.3.3 Diffusive scattering model

$$
\begin{aligned}
\mathbf{S}'(\mathbf{D}(\mathbf{X},\mathbf{Y})) &= \log\left\{\exp(-T)\cdot\prod_{j=1}^{s}\left(\frac{l_0(\mathbf{x}_j)}{g_0(\mathbf{x}_j)} + k\cdot\sum_{i=1}^{m} q_i\cdot\frac{g(\mathbf{x}_j-\mathbf{y}_i;\Theta_i)}{g_0(\mathbf{x}_j)}\right)\right\} \\
&= -T + \sum_{j=1}^{s}\log\left(\frac{l_0(\mathbf{x}_j)}{g_0(\mathbf{x}_j)} + k\cdot\sum_{i=1}^{m} q_i\cdot\frac{g(\mathbf{x}_j-\mathbf{y}_i;\Theta_i)}{g_0(\mathbf{x}_j)}\right) \qquad (3.3.3) \\
&= -T + \sum_{j=1}^{s}\log\left(b_j + \sum_{i=1}^{m}\exp(d_{ij})\right)
\end{aligned}
$$

where *d* is defined as

$$
d_{ij} = \log\left(k\, q_i\cdot\frac{g(\mathbf{x}_j-\mathbf{y}_i;\Theta_i)}{g_0(\mathbf{x}_j)}\right) \qquad (3.3.3')
$$

This model constitutes an approximation to the exact points model and is discussed by [Irving 1996a].

3.3.4 Strong scattering model

$$
\begin{aligned}
\mathbf{S}'(\mathbf{D}(\mathbf{X},\mathbf{Y})) &= \log\left\{\exp(-T)\cdot\prod_{\substack{j=1\\(i,j)\in\Omega}}^{s}\left(\frac{l_0(\mathbf{x}_j)}{g_0(\mathbf{x}_j)} + z_i\cdot\frac{h(\mathbf{x}_j-\mathbf{y}_i;\Theta_i)}{g_0(\mathbf{x}_j)}\right)\right\} \\
&= -T + \sum_{\substack{j=1\\(i,j)\in\Omega}}^{s}\log\left(\frac{l_0(\mathbf{x}_j)}{g_0(\mathbf{x}_j)} + z_i\cdot\frac{h(\mathbf{x}_j-\mathbf{y}_i;\Theta_i)}{g_0(\mathbf{x}_j)}\right) \qquad (3.3.4) \\
&= -T + \sum_{\substack{j=1\\(i,j)\in\Omega}}^{s}\log(b_j + z_i\cdot\exp(d_{ij}))
\end{aligned}
$$

As opposed to the previous model (3.3.3), this model incorporates a random Bernoulli term; in other words, the sum contains an indefinite number of terms, depending on omega.

3.3.5 Bifurcation models

$$
\begin{aligned}
\mathbf{S}'\left(\mathbf{D}\left(\mathbf{X},\mathbf{Y}\right)\right) &= \log\left\{\prod_{\mathbf{X}^{(M)}}\left(\sum_{(i,j)\in w} c_w \cdot \frac{g\left(\mathbf{x}_j - \mathbf{y}_i\, ;\Theta_i\right)}{g_0\left(\mathbf{x}_j\right)}\right)\cdot\prod_{\mathbf{X}^{(U)}}\frac{r\left(x_j\right)}{g_0\left(\mathbf{x}_j\right)}\cdot\prod_{\mathbf{Y}^{(U)}}\left(1-q_i\right)\right\} \\
&= \sum_{\mathbf{X}^{(M)}}\log\left(\sum_{(i,j)\in w} c_w \cdot \frac{g\left(\mathbf{x}_j - \mathbf{y}_i\, ;\Theta_i\right)}{g_0\left(\mathbf{x}_j\right)}\right) + \sum_{\mathbf{X}^{(U)}}\log\frac{r\left(x_j\right)}{g_0\left(\mathbf{x}_j\right)} + \sum_{\mathbf{Y}^{(U)}}\log\left(1-q_i\right) \qquad (3.3.5) \\
&= \sum_{\mathbf{X}^{(M)}}\log\left(\sum_{(i,j)\in w} d_{ij}\right) + \sum_{\mathbf{X}^{(U)}} b_j + \sum_{\mathbf{Y}^{(U)}} a_i
\end{aligned}
$$

As we have seen before, this model is an hybrid from (3.3.1) and (3.3.2).

3.3.6 Permanent model

$$
\begin{aligned}
\mathbf{S}'\left(\mathbf{D}\left(\mathbf{X},\mathbf{Y}\right)\right) &= \log\left\{\sum_{\Omega}\prod_{j=1}^{s}\frac{g\left(\mathbf{X}_j \middle| \mathbf{Y}_{\Omega^{-1}\{j\}}\right)}{g_0\left(\mathbf{X}_j\right)}\Pr\left(\mathbf{Y}_i \middle| H_k\right)\right\} \qquad (3.3.6) \\
&= \log\,\mathrm{Perm}\left(P_i \circ R_i\right)
\end{aligned}
$$

This model requires a different treatment and comments are provided in Chapter 4.

3.4 Search/Match Decision Logic

We are now in a position to formulate the object recognition problem as an optimization search over a ***space*** of models, parameters, hypotheses and interpretations. Multiple algorithms and search strategies are conceivably possible.

The output of an object recognition system is based on a decision logic which optimizes some criteria depending on the match scores for different hypotheses, in order to select the optimal hypothesis value and to declare which objects are present in the image, if any. The aim of the object recognition system is to compute and execute the system's decision logic.

In a simple system, the decision logic merely requires the maximum match score over all possible hypotheses, and declares that hypothesis to be the winner. More sophisticated decision logic systems take into account:

(a) The possibility that the winning hypothesis is not good enough, and the best declaration is that no match has been found;

(b) The fact that the various hypotheses must be determined dynamically and tested in a hierarchical fashion, so that it is impossible to test against all hypotheses sequentially. In this case, the decision logic incorporates a search strategy over the hypothesis space.

More generally, the search problem can become arbirtrarily complex incorporating stochastic payoffs and utilities resulting in sequential decisions and optimal control design parameters to find the optimal navigation and observation strategy which leads to a final decision [Pressman and Sonin 1990].

The general search problem is not considered in this work; interested readers should consult [Wissinger et al. 1996] and [Wissinger et al. 1999].

3.5 Summary

In this Chapter we have constructed match score functions from the density functions given by particular observation-generation models of Chapter 2.

In this way, we have seen how the basic observation-generation models lead to various score functions that will be used in subsequent chapters to present results that allow us to compare between different

- Observation-generation models (such as (2.2.3.1).)
- Match score functions (such as (3.3.1).)
- Feature match-quality measures (such as (3.3.1').)

In this Chapter we have seen how all these components of the match process together with the search decision logic provide an algorithmic framework for the match subproblem in object recognition. In the following two Chapters we offer the particular algorithmic details pertaining to the implementation, and finally we present the results developed in this work in the final Chapters.

Chapter 4. Feature Correspondences

4.1 Introduction

We have seen in the previous Chapter that the computation of a match score often depends on an interpretation. That is, in order to compute a match score (which depends on the class conditional density function evaluated on the observed data), an interpretation that pairs off observed features with predicted features is often required. Alas, the interpretation is not known in advance.

The models that required an interpretation were:

- Independent associative model with correspondences
- Strong scattering model
- Bifurcation models

Other models lead to density functions that do not require an interpretation, since there is no explicit association between observed features and predicted features. This is a great advantage of those models. However, observation generation models involving associations are more realistic in many applications. We examine below some of the implications of requiring an interpretation, versus methods that don’t require an interpretation.

For those methods that require an interpretation, a method is needed to compute the density functions without prior knowledge of the interpretation. This can be accomplished by one of three methods:

- Guessing at a set of correspondences, thereby determining a likely interpretation. We then use the estimated interpretation.
- Attempt to approximate the score, despite the fact that the interpretation is unknown, but summing over all possible correspondences (hoping that incorrect correspondences contribute negligible amounts to the score), or taking a weighted sum over correspondences.
- Maximize over all possible interpretations, and assume that the interpretation yielding the maximum score is correct.

In this Chapter we are concerned with algorithmic methods for putting features in correspondence, and we consider examples of these three previous methods.

4.2 Pattern Recognition and Signature Recognition

In previous chapters, we have seen how an observation-generation model provides an statistical realization for the observed features that can be used to determine the class joint conditional density function. However, when confronted with real data from which observed features, and eventually hypothesized objects are extracted, we must manufacture an interpretation, or pairing between observed and predicted features. In this section, let us consider the situation in the absence of a precise observation-generation model.

In nearly all recognition problems, features are extracted from an observed object, in order to perform discrimination. We now distinguish between two flavors of feature extraction.

If the aggregate collection of features forms a multi-component ordered tuple, then the implication is that each feature component can be compared with a corresponding model component, and the degree of similarity between the components is a measure of the similarity of the observed component values. The essential point in this case is that the components are ordered. Thus, the first component extracted from an observed signature is supposed to match up with the first component of the hypothesized model.

The ordering of feature components means that this particular matching of signatures belongs to the field of "pattern recognition."

As an example, a region might be segmented from a scene, and the features of the segment might be defined as the (1) area, (2) perimeter, (3) eccentricity (suitably defined), (4) centroid and (5) circular moment of inertia about the centroid. These five components are uniquely defined and ordered, and so can be used to compare against five identically defined components of a region predicted from the model.

The classical theory of statistical pattern recognition [Duda and Hart 1973], [Devivjer and Kittler 1982], [Therrien 1992] relies on such feature vector models of an object's characteristics. Objects are represented as the ordered lists of their global characteristic features, each of which corresponds to a point in feature space. Objects are classified by comparison of the feature vectors of the object instance to those of a prototype or "model." Most of the techniques amount to generating a partition of the feature space into regions corresponding to different object models; this allows the assignment of unknown objects to known object classes. The decision boundaries are usually constructed during a learning or training phase using some variation of one of the following approaches: discriminant functions, nearest-neighbor classification rules, decision trees, clustering algorithms, neural networks, etc.

On the other hand, in many cases, the features extracted from a signature form a set of features, as opposed to an ordered tuple. Indeed, the number of features that will be extracted is unknown in many of these cases. This is the situation, for example, when the features are the locations of peaks extracted from a region of interest in a radar image. The collection of such peaks forms a set of features, where each feature is an image location. When performing recognition based on sets of features, we might say that we are in the realm of "signature recognition." In this domain, we have a set of features, where each feature can be a vector, but there is no obvious ordering of the set.

An important issue arises when performing signature recognition. Consider the problem when the observed signature gives rise to a set of s features, and the hypothesized signature contains m features. Not only might there be a different number of features in each set, but there is no obvious way to find a correspondence between the observed and the predicted features. Indeed, it can happen that certain extracted features do not have corresponding predictions in the model, and certain model features do not have corresponding image features. In this case, we might say that part of the problem is the determination of a candidate interpretation for the observed features, in light of the predicted features. This interpretation might involve a set of correspondences, or might arise in some other fashion. In the remainder of this section, we consider methods for finding such an interpretation. These methods will in turn provide means to produce a "best-guess" interpretation.

4.2.1 One-to-one versus many-to-many

Our nominal notion of a feature is that a predicted feature should match up with exactly one observed feature when the model is present.

However, extracted features might be absent, due to noise in the feature extraction process, obscuration of the object in the scene, or other variability. In this case, a model feature should be labeled "unmatched." In other cases, there might be extra features extracted from the scene, due to noise, extraneous objects, or extraction errors. In this case, the extracted feature is "unmatched."

There are also situations where features can be matched to multiple features. That is, features might be non-exclusive. As a simple example, we can imagine a single feature called the "unmatched label," where we would have one such feature for the model features, and one for the extracted features; then every unmatched predicted feature matches to the single unmatched label of the extractions. A similar one-to-many pairing can occur with the unmatched label of the predictions. A less trivial case can occur with bifurcation of peak locations in SAR imagery (Chapter 6.) For example, a single predicted peak might actually be resolved into a pair of peaks in the observed image. In that case, both extracted peaks should be matched to the single predicted peak. It might also occur that a pair of peaks in the predicted image is not actually resolved in the scene. One might view one peak as being matched, and the other unmatched, but instead one can view the pair of predicted peaks as both matching the single observed peak.

One of the crucial assumptions in many approaches is the fact that feature correspondences arise in a one-to-one fashion, or can be approximated by one-to-one mappings. In general, the one-to-one situation seems to be simpler, because we need only to deal with correspondence sets from a permutation group. However, as we will see, the many-to-many situation permits certain computational simplifications, and so is potentially advantageous. One of the prime obstacles with the many-to-many correspondences is that it is difficult to take into account the prior probabilities of correspondences that are not permutations.

The number of potential correspondences grows exponentially with the number of model features (m) and scene features (s).

Specifically, the number of one-to-one assignments can be seen to be $\sum_{k=0}^{\min(m,s)} \binom{m}{k}\binom{s}{k} \cdot k!$, which is of the asymptotic order of $\max(m,s)^{\min(m,s)}$.

Considering all sorts of correspondences and allowing for clutter and obscured features we arrive at the numbers shown in Table 4.1.

Model	Scene	Interpretations	Injective Interpretations	Surjective Interpretations
One	One	$\sum_{k=0}^{m\wedge s}\binom{m}{k}\binom{s}{k}\cdot k!$ $\approx (m\vee s)^{m\wedge s}$	$m!\binom{s}{m}\cdot \mathbf{1}(m\le s)$	$s!\binom{m}{s}\cdot \mathbf{1}(s\le m)$
One	Many	$(m+1)^{s}$	$m!\,(m+1)^{s-m}\cdot \mathbf{1}(m\le s)$	m^{s}
Many	One	$(s+1)^{m}$	s^{m}	$s!\,(s+1)^{m-s}\cdot \mathbf{1}(s\le m)$
Many	Many	$2^{m\cdot s}$	$(2^{s}-1)^{m}$	$(2^{m}-1)^{s}$

Table 4.1. Number of interpretations

In subsequent sections we consider three flavors of the problem of searching for correspondences and interpretations.

4.2.2 Greedy Interpretations and its Variations

Finally, we consider algorithmic methods for determining interpretations that provide "guesses" or likely associations between observed features and predicted features. This section considers a collection of methods collectively called "greedy interpretations."

The greedy one-to-many model-to-image interpretation is given by

$$w^*(\mathbf{y}_i) = \arg\max\{d(\mathbf{y}_i, \mathbf{x}) : \mathbf{x} \in \mathbf{X}\} \text{ for } i = 1,2,\ldots,m. \tag{4.2.2.1}$$

On the other hand, the image-to-model greedy interpretation is a many-to-one partial association defined as

$$(w^{-1})^*(\mathbf{x}_j) = \arg\max\{d(\mathbf{y}, \mathbf{x}_j) : \mathbf{y} \in \mathbf{Y}\} \text{ for } j = 1,2,\ldots,s. \tag{4.2.2.1'}$$

We can define the one-to-one versions of these greedy schemes by simply eliminating those pairs $(\mathbf{y}, \mathbf{x})$ in a greedy fashion as necessary until we obtain a bijection between subsets of $\mathbf{X}$ and $\mathbf{Y}$.

Yet another variation arises if we wish to designate elements of $\mathbf{X}$ and/or elements of $\mathbf{Y}$ as being unmatched. A simple way of doing this is to eliminate any pairings $(\mathbf{y}, \mathbf{x})$ whose distance (namely its match-quality measure) falls outside a permissible range. This could be done either before or after eliminating non-one-to-one pairings.

4.2.3 Diffuse Interpretations

Although the Diffuse Scattering model (2.2.3.3), (3.1.3.3) does not involve explicit correspondences or interpretations, it leads to a match score evaluation formula equivalent to a weighted sum over all possible many-to-one products of individual correspondences [Ettinger et al. 1996]. The weights are proportional to prior probabilities of detection of the predicted features q_i. In order to see this, observe that the product appearing in (3.1.3.3) is

$$\exp\{R+\mathbf{S}'(\mathbf{D}(\mathbf{X},\mathbf{Y}))\}=\prod_{j=1}^{s}\left(\frac{l_0(\mathbf{x}_j)}{g_0(\mathbf{x}_j)}+k\cdot\sum_{i=1}^{m}q_i\cdot\frac{g(\mathbf{x}_j-\mathbf{y}_i;\Theta_i)}{g_0(\mathbf{x}_j)}\right) \qquad (4.2.3.0)$$

and this product can be rewritten as follows:

$$\prod_{j=1}^{s}\left(\frac{l_0(\mathbf{x}_j)}{g_0(\mathbf{x}_j)}+k\cdot\sum_{i=1}^{m}q_i\cdot\frac{g(\mathbf{x}_j-\mathbf{y}_i;\Theta_i)}{g_0(\mathbf{x}_j)}\right)=\sum_{w}\prod_{(i,j)\in w}\left(r_i\cdot\frac{f_i(\mathbf{x}_j)}{g_0(\mathbf{x}_j)}\right) \qquad (4.2.3.1)$$

To see this, observe that the right-hand side of the preceding equation is obtained by expanding its left-hand side using the following general formula:

$$(a_{11}+\cdots+a_{1s})\cdots\cdots(a_{m1}+\cdots+a_{ms})=\sum_{w:M\mapsto S}a_{1w(1)}a_{2w(2)}\cdots a_{mw(m)}\ .$$

There are exactly $(m+1)^s$ product terms in the sum in the right-hand side of (4.2.3.1). Each such term corresponds to a different interpretation including the possibility of null or vacuous correspondences. This approach includes all the many-to-one image-to-model interpretations in addition to the one-to-one interpretations. Therefore, there are no explicit correspondences in the evaluation of the left-hand side of (4.2.3.1) but the right-hand side formula can be considered as an exhaustive enumeration over many-to-one image-to-model interpretations. See the next subsection 4.2.4 for an alternative model.

However, the left-hand side formula can be computed in time proportional to $O(m \cdot s)$ and there is no explicit need for correspondences although they are implicit in the product.

4.2.4 Permanent Correspondences

As an alternative to the previous formula, here we discuss the situation when we consider only the one-to-one interpretations only, instead of arbitrary products of correspondences.

This is the case for example for landmark point features when the image resolution is high enough that features are well separated from one another, or when correspondences can otherwise be considered to be one-to-one on a phenomenological basis.

In this case, the sum can be expressed as a permanent function on a related matrix as in (3.1.3.6).

$$\exp \mathbf{S}'(\mathbf{D}(\mathbf{X},\mathbf{Y})) = \sum_{\Omega} \prod_{(i,j)\in\Omega} \left(r_i \cdot \frac{f_i(\mathbf{x}_j)}{g_0(\mathbf{x}_j)} \right) \qquad (4.2.4.0)$$

Note the similarity of this equation with the corresponding one arising from (4.2.3.0) and (4.2.3.1), although there is no evaluation simplification here analogous to (4.2.3.1).

Though none of the other observation-generation models leads to a requirement to sum over all one-to-one correspondences, we might nonetheless be interested in the permanent model computation because it will give an over-estimate of the match score for the maximum-likelihood one-to-one interpretation Ω^* (Subsection 4.2.5). The error incurred by including correspondences that are "incorrect" might be small if the "correct" correspondence dominates the sum.

Unfortunately, the computational complexity of the permanent of an n by n matrix turns out to be $O(n!)$, although there are some useful approximations and bounds that can be applied. Nonetheless, the number of operations required to compute the sum is exponential in the number of features.

In the next section, we present an efficient method of computing the maximum likelihood one-to-one interpretation Ω^*.

4.2.5 Bipartite Correspondences

We have discussed above methods for normalizing over possible correspondences by summing over permutations or mappings. However, another method for normalizing would be to determine a maximum over possible correspondence sets. Consider the problem of finding the maximum likelihood or the maximum a posteriori interpretation Ω^* or $\tilde{\Omega}^*$.

The independent associative model of equation (3.2.4.1) results in a match score function given by

$$\mathbf{S}'\left(\mathbf{D}(\mathbf{X},\mathbf{Y})\right)=$$
$$\sum_{\substack{x_j\in\mathbf{X}^{(M)}\\(i,j)\in w}}\log\left\{q_i\,\frac{g\left(\mathbf{x}_j-\mathbf{y}_i\,;\Theta_i\right)}{g_0\left(\mathbf{x}_j\right)}\right\}+\sum_{x_j\in\mathbf{X}^{(U)}}\log\left\{\frac{r\left(\mathbf{x}_j\right)}{g_0\left(\mathbf{x}_j\right)}\right\}+\sum_{y_i\in\mathbf{Y}^{(U)}}\log\left(1-q_i\right) \quad (4.2.5.1)$$

It is easy to see that equation (4.2.5.1) can be written as a sum of weights for a given one-to-one correspondence hypothesis of predicted to extracted features. Namely, we can write a weighted bipartite assignment problem as follows:

$$\text{maximize}\quad \sum_i\sum_j c_{ij}z_{ij}+\sum_i a_iy_i+\sum_j b_jx_j \qquad (4.2.5.2.0)$$

subject to

$$\sum_j z_{ij}+y_i=1 \text{ for } i=1,2,\ldots,m. \qquad (4.2.5.2.1)$$

$$\sum_i z_{ij}+x_j=1 \text{ for. } j=1,2,\ldots,s. \qquad (4.2.5.2.2)$$

The coefficients a_i, b_j, c_{ij} are not necessarily nonnegative, but it is easily seen that the problem is equivalent to

$$\text{minimize} \quad \sum_i^m \sum_j^s \left(M - c_{ij}\right) z_{ij} + \sum_i^m \left(\tfrac{1}{2} M - a_i\right) y_i + \sum_i^s \left(\tfrac{1}{2} M - b_j\right) x_j \qquad (4.2.5.3)$$

subject to the same constraints (4.2.5.2.1), (4.2.5.2.2). In fact, the solutions to both problems are the same. As a consequence, the original problem can be converted to a minimization problem where all the coefficients are nonnegative, by simply choosing a sufficiently large value for M. We can therefore assume that the original weights are nonnegative.

By introducing appropriate slack variables, the problem can be further rewritten as the following problem in canonical form, which can then easily be solved.

The modified problem is

$$\text{minimize} \quad \sum_i^{m+s} \sum_j^{m+s} \tilde{c}_{ij} \cdot \tilde{z}_{ij} \qquad (4.2.5.4.0)$$

subject to

$$\sum_j^{m+s} \tilde{z}_{ij} = 1 \text{ for } i = 1,2,\ldots,m+s\,. \qquad (4.2.5.4.1)$$

$$\sum_{i}^{m+s} \tilde{z}_{ij} = 1 \text{ for } j = 1,2,\ldots,m+s. \qquad (4.2.5.4.2)$$

where the matrix (c_{ij}) is defined in terms of the values a_i, b_j, and c_{ij} as follows:

$$\tilde{C} = \left(\begin{array}{ccc:ccc} c_{11} & \cdots & c_{1s} & \ddots & & +\infty \\ \vdots & \ddots & \vdots & & a_i & \\ c_{m1} & \cdots & c_{ms} & +\infty & & \ddots \\ \hdashline \ddots & & +\infty & \ddots & & \\ & b_j & & & 0 & \\ +\infty & & \ddots & & & \ddots \end{array}\right)\begin{array}{c} \uparrow \\ m \\ \downarrow \\ \uparrow \\ s \\ \downarrow \end{array}$$

To prove this, observe that any feasible solution to (4.2.5.2.1), (4.2.5.2.2) will give rise to a solution of (4.2.5.4.1), (4.2.5.4.2). Conversely, a solution $\{z_{ij}, y_i, x_j\}$ to the former problem can be obtained from the solution to the latter one by reading off the values: $z_{ij} = \tilde{z}_{ij}$ for $1 \le i \le m$, $1 \le j \le s$, and $\tilde{x}_j = \tilde{z}_{m+j,j}$ for $1 \le j \le s$, $y = \tilde{z}_{i,s+i}$ for $1 \le i \le m$. The values $\tilde{z}_{m+j,s+i}$ are just slack variables for the original problem and are uninteresting.

Therefore the original problem can be seen as a weighted minimization in a bipartite graph where m model feature nodes are supplemented with s extra nodes, one for each image feature to serve as potential no-match nodes, and the s image features are supplemented with m no-match nodes, one for each model feature.

In this way, a combinatorial number of possible assignments can be examined and an optimal assignment can be obtained in polynomial time. The essential ingredients is the structure of the match score as a sum of nonnegative weights, and this formulation in turn depends on using log-likelihood functions and conditional independence assumptions.

For a practical implementation, the critical issue is the estimation of parameter values that provide the penalization terms r_j and q_i.

For a detailed presentation of the algorithms, see [Garcia and Hummel 1997].

Previous Use of Bipartite Matching in Computer Vision

Many authors have made use in the past of various algorithms to solve assignment problems in order to pair features or to accomplish other tasks requiring correspondences.

Baird [1985] points out that given a registration and assuming no spurious or missing feature points, the problem of finding an optimal matching can be transformed to an instance of the assignment problem in quadratic time. He then uses a suboptimal strategy based on constraining the set of feasible matchings by imposing a bounded error model, and presents an algorithm to find a matching and a similarity transformation in average quadratic time.

[Kim and Kak 1991] made use of bipartite matching together with discrete relaxation to perform recognition of objects from bin parts using the output of an structured light scanner producing range depth data. Each object is represented by an attributed graph whose nodes are surface features, and whose arcs are edges between the surfaces. Bipartite matching is used for two different tasks: in an early stage to prune large segments of the search space by quick wholesale rejection of inapplicable models, and in the final stage to determine the compatibility of a scene surface with potential model surfaces, taking into account local compatibility constraints.

[Breuel 1990] used pruned correspondence search and bipartite matching to solve for the best interpretation using alignment.

[Cox et al. 1995] use bipartite matching for hierarchical grouping of edge and line features. Edge measurements are assigned to possible tracks (partial hypotheses) that provide an evolutionary description of image contours. Assignments of measurements to tracks are organized into a hypothesis tree where each node is characterized by a probability computed from track estimators and priors. At each iteration in the process, the top few segmentation hypotheses are selected, and current parameter estimates are updated according to new measurements. Grouping decisions are postponed until a sufficient amount of information is available. Variations of Murty’s Algorithm [Murty

1968] and Multiple Hypotheses Tracking [Raid 1979] are used to keep the computation within feasible time bounds during the search process.

[Wolfson et al. 1991] used bipartite matching to find edge pieces when constructing jigsaw puzzles. In this case, pieces from the jigsaw puzzle with flat sides must belong to the boundary, and there are two kinds of such pieces –even or odd—which form a natural bipartition of the set of boundary pieces. The formulation follows by maximizing the number of pieces being matched.

[Poore 1995] has used weighted multidimensional assignment in the context of multiple hypothesis tracking.

4.2.6 Generalized models

In the same way we can formulate and solve an arbitrarily constrained problem over a subset of the possible interpretations or correspondences as a problem of finding a maximum cut or a maximum flow over a graph or matroid structure. For example, when joint distributions of the sets $\mathbf{X}$ and $\mathbf{Y}$ are considered in order to take account of possible dependencies among multiple subsets of extracted and predicted features, we can arrive at complex combinatorial optimization problems. As a particular case, pairwise dependencies lead to a quadratic or higher-order multidimensional assignment problems discussed in [Pardalos and Wolkowicz 1995]. Solving such problems may require exponential time and space but the particular form of the underlying combinatorial structures can be exploited. This remains a topic of future research; see also [Geiger and Ishikawa 1998].

4.3 Summary

In this chapter we have seen examples of how to establish interpretations and correspondences between model and scene features; such interpretations are necessary to evaluate particular instances of the class density functions appearing in many observation-generation models.

In this context, observe that correspondences and interpretations are inherent to the particular observation-generation model. The fact that they happen to be unknown to the observer is not considered to be an attribute of the model.

Chapter 5. Geometric Hashing

5.0 Overview

In this Chapter, we discuss algorithmic methods for implementing match scoring functions that were presented in Chapters 2 and 3, and that have been used for experiments described in subsequent chapters. The main challenges are to achieve transformation invariance of the features (such as invariance to translation when searching for target objects), and efficient implementation. We discuss the use of geometric hashing to handle both of these challenges.

5.1 Transformation Invariance

The matching problem is complicated by the fact that the position of the model in the scene is not necessarily known in advance. That is, the model can undergo a transformation prior to being corrupted by noise and embedded in the scene.

The important point of invariance theory is the idea that a collection of features can be replaced by a new collection of features such that the new collection is invariant under some class of transformations. For example, if we have a collection of point locations, then a larger collection of the differences of all pairs of point features forms a new pattern that will be translation-independent. Similar transformations, involving all triples of point features, can render a new collection of features translation and rotation, or translation, rotation, and scale invariant.

5.2 Overview of Geometric Hashing

We have first discussed geometric hashing in Section 1.9. The attractive aspect of this technique is that offline computation is used to replace online computation by pre-compiling an index table, which is then used as an associative memory at runtime for fast retrieval of promising candidate hypotheses. Exhaustive references describing the method can be found in [Rigoutsos 1992]. We limit the presentation here to a brief overview and we mention only those aspects that are relevant in subsequent discussion.

In geometric hashing, the collection of models is used in a preprocessing phase (executed offline and only once) to build a hash table data structure, which provides, for each normalized feature in the observed scene, a list of candidate model/parameter pairs. In this way, the hash table encodes information about the models in a redundant way, including all potential interpretations indexed by minimal subsets of transformation-invariant features. During the online recognition phase, when the algorithm is presented with an observed scene and extracted features, the hash table is used to recover candidate matching models by accumulating votes for potential models and transformations on the basis of partial correspondences between subsets of features. A search is still required over scene features, but no search is needed over model features and/or correspondences.

The novel approach in our implementation of geometric hashing is that it is able to handle multiple match score functions used as criteria for selecting the best hypotheses during the voting procedure. The concept and initial implementation for this approach owe much to James Russell and to Eric Freudenthal; subsequent work by the author and others refined the software and its ability to handle multiple match metrics.

5.3 Algorithmic Implementation

We present below a pseudo-code implementation of the algorithm's online phase, using an object-oriented SETL-like language [Schwartz et al. 1986].

The crucial step in the algorithm is the accumulation of votes in the innermost loop, which can be fully implemented in parallel SIMD processors [Rigoutsos and Hummel 1990]. This implementation shows how arbitrary score function computations can be achieved in geometric hashing by maintaining a separate accumulator for each one of the scene features.

At the end of the first inner loop over each basis set, the accumulator arrays contain the value of contributions for each predicted model and transformation, i.e., the first loop efficiently computes the quantities

$$\mathrm{votes}[\mathbf{X}_j][T,\Theta] = \sum_{\mathbf{Y}_i(T,\Theta)} \mathrm{score}\left(\mathbf{X}_j, \mathbf{Y}_i \middle| T,\Theta\right) \qquad (5.3.1)$$

for **all** possible models and transformations and for all observed features $\mathbf{X}_j$

Afterwards, in the second loop over model/transformation pairs with nonzero votes, the contributions for individual features can be combined using an arbitrary function Φ to evaluate a match score for the model hypothesis (T,Θ)

$$\mathrm{total\_votes}[T,\Theta] = \Phi\left(\left\{\mathbf{X}_j\right\}_{j=1}^{n}, \left\{\mathrm{votes}[\mathbf{X}_j][T,\Theta]\right\}\right) \qquad (5.3.2)$$

The right-hand side usually contains an additive `bias` term, which depends on the model *T* only, and is independent of the features X_j, and can therefore be preloaded at the initialization stage.

The critical issue is the distribution of entries in the hash table, so that the list of nonzero vote-getting entries should be short enough. Upper bounds on the list length and the number of entries depend on the distribution of entries in the hash space and on the noise model [Rigoutsos and Hummel 1990].

Details about the code and the implementation are described in the following paragraphs.

```
Given pattern EP, Hash Table HT
for ebasis in EP.enumerate_bases() loop
  initialize votes, nonzero_votes, total_votes
  for ef in EP.features loop
    transformed_feature = ef.normalize(ebasis)
    for entry in HT.entries(transformed_feature) loop
       ppb = entry.pattern_basis
    votes[ef][ppb] = entry.increment_score(transformed_feature)
       nonzero_votes += { ppb }
    end loop
  end loop
  for ppb in nonzero_votes loop
    total_votes[ppb] = EP.bias();
    total_votes[ppb] = + / [ votes[ef][ppb] :  ef in EP.features ]
  end loop
  assert exists pbasis in nonzero_votes |
    total_votes[pbasis] = max / total_votes
  if stop_criteria( total_votes[pbasis] ) then output(pbasis,ebasis)
end loop
```

5.3.1 Code description

It is assumed that model and observed features are transformed to a nominal coordinate system in order to account for geometric invariance; the required transformation is known as normalization and is implemented by the function `normalize`. A basis is a minimal transformation-invariant subset of features and therefore describes a unique coordinate transformation, which is required by the function `normalize`. Accordingly, in line 1, a basis is chosen from the set of bases for the extracted pattern EP. The function `enumerate_bases` provides a heuristic enumeration of bases that effectively accomplishes a search over observed scene features and scene transformations. Invariance is accomplished by applying the function `normalize` to each of the features ef in EP, in line 4.

The hash table structure provides, for each normalized feature in the scene, a list of candidate model/parameter pairs (known as `entries`). The parameter pair is a designation of the basis that was used at the time that the entry was created. Thus in line 5, we access the hash table at the location indicated by the normalized feature, and walk down the list of entries at that location. For each entry, there is a candidate model and a basis within that model, which is encoded as an index ppb as extracted in line 6. The function `increment_score` is explained for each of the observation generation models in Section 5.4. Keeping track of `entries` with nonzero votes allows sub-linear time performance of the algorithm [Hummel and Wolfson 1988]. Lines 11 to 14 perform the accumulation of the scores for each of those entries; this is the function Φ of (5.3.2). The search in line 15 is over the same set of entries with nonzero votes. The notation +/list adds together the elements in the list, and the notation max/list returns the maximum in the list.

At the end of the outermost loop, or when `stop_criteria` is satisfied in line 16, the current best model and transformation comprise the output of the recognition algorithm.

5.4 Geometric Hashing Implementation of Match Score Functions

In this section, we show how some of the match score functions from Chapter 3 can be implemented using Geometric Hashing within the general framework of Section 5.3.

We show the necessary score increment functions that need to be plugged in (5.3.1) and the Φ functions in (5.3.2) in order to compute the match score metrics from Chapter 3.

Observe that the traditional interpretation that has historically been associated with geometric hashing is a greedy image-to-model interpretation (Section 4.2.2) but we could use other variations as well (see the examples below).

For example, the independent mixture model 5.4.2 and the bifurcation model in 5.4.5 permit the inclusion of arbitrary interpretations w .

5.4.1 Independent associative model with correspondences

In this case we simply see that (5.3.1) and (5.3.2) reduce to

$$\text{votes}[\mathbf{X}_j][T,\Theta] = b_j + \sum_{w^{-1}(i)} d_{ij} + \sum_{w^{-1}(i)} a_i \qquad (5.4.1)$$

$$\text{total\_votes}[T,\Theta] = \sum_{\mathbf{X}^{(U)}} b_j + \sum \sum_{(i,j)\in w} d_{ij} + \sum_{\mathbf{Y}^{(U)}} a_i \qquad (5.4.1')$$

where (i, j) characterize the interpretation ω, and the bias term is $\Psi(\{\mathbf{X}_j\}) \equiv \sum_{\mathbf{X}^{(U)}} b_j$ and the definitions of *a*, *b*, *d* are as in (3.3.1) and (3.3.1') namely

$$d_{ij} \equiv d(\mathbf{Y}_i, \mathbf{X}_j) = \log\left(q_i \cdot \frac{g(\mathbf{x}_j - \mathbf{y}_i; \Theta_i)}{g_0(\mathbf{x}_j)} \right) \qquad (3.3.1')$$

These terms can be directly computed from the values of `entry` and `ef` in the code above.

5.4.2 Independent mixture model

Following (3.3.2) and (3.3.2')

$$\text{votes}[\mathbf{X}_j][T,\Theta] = \sum_i \exp(d_{ij}) \qquad (5.4.2)$$

$$\text{total\_votes}[T,\Theta] = \sum_{\mathbf{x}^{(U)}} b_j + \sum_{\mathbf{x}^{(M)}} d_{ij} \qquad (5.4.2')$$

where

$$d_{ij} = \log\left(\frac{q_i}{\sum q_l} \cdot \frac{g(\mathbf{x}_j - \mathbf{y}_i; \Theta_i)}{g_0(\mathbf{x}_j)} \right) \qquad (3.3.2')$$

This model requires an interpretation hypothesis.

5.4.3 Diffusive scattering model

From (3.3.3) and (3.3.3')

$$\text{votes}[\mathbf{X}_j][T,\Theta] = b_j + \sum_i \exp(d_{ij}) \qquad (5.4.3)$$

$$\text{total\_votes}[T,\Theta] = -R + \sum_j \log\left(b_j + \sum_i \exp(d_{ij}) \right) \qquad (5.4.3')$$

where the definitions are as in (3.3.3) and (3.3.3')

$$d_{ij} = \log\left(k\, q_i \cdot \frac{g(\mathbf{x}_j - \mathbf{y}_i; \Theta_i)}{g_0(\mathbf{x}_j)} \right) \qquad (3.3.3')$$

and the bias term here is $\Psi(\{\mathbf{X}_j\}) \equiv -R$. Observe that an interpretation is not needed in this model.

5.4.4 Strong scattering model

This model follows from (3.3.4)

$$\text{votes}\left[\mathbf{X}_j\right]\left[T,\Theta\right] = b_j + \exp\left(d_{w^{-1}(j)\,j}\right) \qquad (5.4.4)$$

$$\text{total\_votes}\left[T,\Theta\right] = -T + \sum_{(i,j)\in w} \log\left(b_j + \exp\left(d_{ij}\right)\right) \qquad (5.4.4')$$

5.4.5 Bifurcation models

For this model we have according to (3.3.5)

$$\text{votes}\left[\mathbf{X}_j\right]\left[T,\Theta\right] = b_j + \log\left(\sum_i \exp\left(d_{ij}\right)\right) \qquad (5.4.5)$$

$$\text{total\_votes}\left[T,\Theta\right] = \sum_{\mathbf{X}^{(U)}} b_j + \sum_j \log\left(\sum_{(i,j)\in w} \exp\left(d_{ij}\right)\right) + \sum_{\mathbf{Y}^{(U)}} a_i \qquad (5.4.5')$$

Any variations of the above schemes involving arbitrary interpretations are possible.

5.4.6 Permanent model

The score function for the permanent model cannot be represented as in (5.3.1), (5.3.2) and therefore an implementation using geometric hashing is not efficiently realizable.

5.5 Summary

In this Chapter we have seen the use of a voting technique in geometric hashing to yield an efficient implementation of the match algorithm with the observation- generation models and score functions from Chapter 3.

The use of geometric hashing provides a reliable, fast way of computing the best model and interpretation of an observed scene according to a match score function. Measures of performance identification and false alarm rates are not a property of the algorithmic implementation but rather of the feature attributes and of the match score functions used for recognition; therefore, statements like “Geometric hashing yields high false alarm rates” are inaccurate since such statements almost surely refer to a particular implementation instance using a particular observation-generation model and a particular score function (e.g. a count measure.) We have shown that the technique of geometric hashing is able to handle the implementation of score functions successfully used in other recognition schemes.

Chapter 6. Synthetic Imagery

6.1 Experiments with Synthetic Data

In this chapter, we report on experiments comparing different match score functions, using synthetically generated data. Although synthetic data are not necessarily realistic in the sense of being representative of the real world, it allows us to carefully control the model under which data are instantiated, and therefore to compare match measures without regard to the validity of underlying noise models. If we rely only on experimental data from a real application, such as the SAR application studied in the next Chapter, then it can happen that one model works better than another due to the particular application domain or to the errors in the features for that particular domain, and not due to superior performance by the match scores.

6.2 Experiment Design

We discuss here the generation of data, the experiment methodology, the performance measures, and the different experiments as a function of the number of clutter and occluded features and as a function of the number of models and patterns.

6.2.1 Experiment Data

The baseline test data consists of $N = 20$ model point sets generated under two different observation models: uniform and diffusive. The test set used for evaluation consists of $W = 25$ corrupted patterns for each of the $N = 20$ target models for a total of $M = 500$ model point sets, in addition to $M = 500$ clutter point sets. The experiment was repeated $T = 1000$ times using independent MonteCarlo simulations.

A. Uniform Target Test Set.

For each $k = 1,2,...,N$ a model M_k consists of *n*=15 uniformly distributed random point locations inside a bounding box of 6 by 3 meters.

For each target model M_k, we have generated *W* perturbed test patterns based on model M_k. The perturbed patterns are constructed as follows:

An integer *d* is chosen from a Poisson distribution with mean $\bar{d} = 5$. Then *d* feature points from model M_k are chosen at random, and are deleted from the pattern. An integer e is chosen from a Poisson distribution with mean $\bar{e} = 5$. Then e extra feature points are generated and added to the test pattern. The generation of these points is done in the same manner as the generation of the model points. Finally, zero-mean Gaussian noise is added to the location of each feature, using a standard deviation of $s = 0.20$

meters in each coordinate. One of the original model points is designated as a basis point.

B. Diffusive Target Test Set.

This set is similar to the uniform test set (A), with the difference that for each model point in M_k, a Poisson number of point locations are generated around that point by adding independent zero-mean Gaussian noise. The expected number of points generated for each model point is related to the probability of occurrence of that model point [Wissinger et al. 1996].

C. Clutter Test Set.

The clutter set is generated as follows:

An integer s is chosen at random based on a Poisson distribution with mean $\bar{s} = n - \bar{d} + \bar{e} = 15$. We generate s independent random point locations inside a 6 by 3 meter box, according to a uniform distribution. One of the s points is designated as a basis point. We generated M=500 such clutter models.

6.2.2 Experiment Definition

Our experiments fall into the following categories:

A. Comparison of Match Algorithms.

For the first set of experiments, we perform recognition assuming that the model can be translated in the test scene. In fact, all test models are translated by a zero amount, and a completely correct identification will identify the correct model number, and deduce that the translation is zero.

We measured probability of correct identification (P_{ID}) and probability of false alarm (P_{FA}) over T=1000 MonteCarlo realizations of the experiment data. The false alarm rate is simply measured as relative correct target identification rate as a percentage over the clutter set, and not as a False Alarm Rate per square kilometer.

We wish to compare the following different match score and algorithm models:

1 Baseline greedy with model-to-image interpretation

2 Baseline greedy with image-to-model interpretation

3 Baseline greedy with one-to-one model-to-image interpretation

4 Baseline greedy with one-to-one image-to-model interpretation

5 Baseline with bipartite one-to-one correspondences

6 MAT MSTAR evaluation score (Many-to-All)

7 Diffuse scattering score

8 Two-sided Hausdorff score

9 Permanent score

As discussed in Chapter 5, geometric hashing can be used to implement each of the above scores, except for the Baseline with bipartite correspondences and the Permanent score. We have implemented a fast network algorithm from [Cherkassky and Goldberg 1998] to evaluate the Bipartite score and we left out the Permanent score from the list due to expediency and time considerations.

Furthermore, we have used the values of $p_d = \left(n - \bar{d}\right)/n$ for all of the predicted points, and $g = n/\left(n + \bar{e}\right)$. Of course, here we have assumed that we have perfect knowledge of the probabilities with which predicted features fail to match, and the extracted features fail to match.

We performed multiple experiments using different values for the control parameters s, $\bar{d}$, $\bar{e}$, N and we chose to report results for those parameter values that were well suited to show tradeoffs for the selected performance measures.

It would be desirable to conduct a sensitivity analysis study as a function of s, $\bar{d}$, $\bar{e}$, N (see subsections B, C, D, and E). A thorough analysis is deferred for future work. Instead, we only offer some comments based on limited experience.

B. Breakdown with *N*.

In both pattern recognition and signature recognition, there is a perception among researchers that a sudden breakdown effect will be observed as the number of models in the database grows indefinitely. That is, one expects reasonable performance as long as there are not too many different models amongst which we wish to discriminate. However, when the number of models becomes too large, we expect the performance to suddenly fall to a point that is essentially worthless. This perception perhaps exists because many pattern recognition studies claim good performance, but generally deal with relatively few models, whereas many researchers have experience with achieving relatively poor performance, and have ascribed the difficulties to real world variability and an excessively difficult discrimination task.

In our experience, we see degradation in performance as *N* increases, but we have not yet seen a breakdown effect. Perhaps we have not taken *N* out far enough, and perhaps the remaining parameters fortuitously show a graceful performance.

C. Breakdown with Noise.

As the amount of Gaussian perturbation is increased beyond the cell resolution, performance suffers due to the inherent noise generation and the lesser reliability of the matching algorithms.

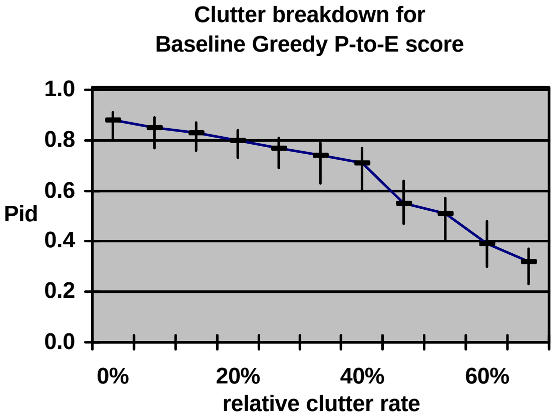
The effect on performance degradation is limited by the inherent greediness of the score in the case of the greedy measures and bipartite score or due to the smoothing effects for the diffuse scattering model.

The most significant effect for our purposes, however, is on the running time. Overall running time is amplified in geometric hashing since the representation of model covariances in the hash table requires many more models to be searched for each particular bucket, as the list length in every hash bucket grows. In the limit, geometric hashing becomes tantamount to an exhaustive search over correspondences. In our experiments, the noise breakdown effect occurs for s somewhere between 0.45 and 0.50, such that the search is as inefficient as exhaustive search.

D. Breakdown with Clutter.

When spurious clutter features are near or inside the “target region,” the effect of increased clutter is to reduce the score and to make discrimination harder.

The effect is relatively minor as long as the number of clutter features is small and the penalty is limited, but if the relative number of clutter features goes above 40 percent, there is considerable confusion that is observed. Oftentimes, a spurious correspondence occurs with a feature closer than that where a true correspondence would have occurred if there were no clutter. The result is that all scores are increased, although incorrect models can be raised more than the true match, causing miss-ID.



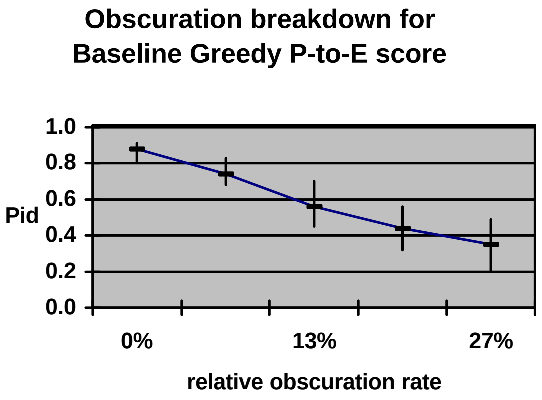
On the other hand, if clutter features appear outside of the bounding region of the target, they are for the most part ignored due to saturation effects and the relative penalties on the scores. This is simply what was to be expected.

The issue of clutter breakdown is much related to the inherent separability and confusability of the underlying models, and as such is an extremely complex subject on its own. Our experiments used the percentage of non-model points as a measure of clutter, and our conclusions are that we can handle up to 40% relative non-model features.

E. Breakdown with Obscuration.

The issue of obscuration is a serious one. Performance in the presence of obscuration is dependent on the correlation between features in the occluded region (as in blocking obscuration.) Our approach to obscuration implicitly penalizes the score when features are missing, but the penalty is limited by the model for non-target features (i.e., background features). Our approach fails to take into account the fact that obscuration usually results in a spatially contiguous region of missing features. Other approaches are possible. For example, the Markov Random Field model as proposed by [Castañon et. al. 1999] could provide better results, although others have reported similar results with both random obscuration and blocking obscuration [Ettinger and Klanderman 1996] with the diffuse scattering model and its equivalent Many-to-All model.

Our experiments indicate the existence of a breakdown, with approximately 10 to 15 percent of random obscuration, for the greedy models. In these experiments, 30 models were used. Better performance should be possible by using an iterative approach, where obscuration regions are hypothesized based on preliminary recognitions. However, we have not implemented such a scheme here. Accordingly, performance with obscuration is disappointingly poor, in large part due to the relatively large number of models.



6.3 Experiment Results

A. Uniform Target Test Set.

The table shown below summarizes the correct-identification and false alarm probabilities that have been measured in experiments using the uniform target test set as described in Section 6.2.1.A above. We have included error bars as determined by 1000 Monte Carlo replications of the experiment.

Measure	P_{ID}	P_{FA}
Greedy E-to-P	0.853±0.066	0.331±0.041
Greedy E-to-P 1-1	0.871±0.069	0.355±0.056
Greedy P-to-E	0.882±0.065	0.330±0.048
Greedy P-to-E 1-1	0.884±0.069	0.339±0.054
Many-to-All	0.865±0.081	0.320±0.054
Diffusive Scatter	0.865±0.081	0.320±0.054
Hausdorff 2-sided	0.859±0.067	0.331±0.061
Bipartite	0.908±0.039	0.393±0.066

We observe that the bipartite measure yields the best PID, but also leads to higher PFA. If we measure rates in a relative fashion, then the increase in false alarm rate of 10.7 percent for the bipartite score as compared to the [greedy E-to-P one-to-one score] is compensated for by a decrease of $\frac{1.0-0.871}{1.0-0.908}-1.0$ = 40.2 percent in miss-identification rate. In addition, the average translation error associated with those targets correctly identified by the bipartite algorithm is 0.19 meters and the standard deviation is 0.02 meters. The corresponding numbers for the greedy score are 0.11 and 0.02 meters. These are relatively small translation errors, in both cases.

B. Diffusive Target Test Set.

It turns out that in the experiment described in (A), the diffusive scattering score measure is at a disadvantage, since features were generated according to a uniform model that is different than the observation generation model assumed by the diffusive scattering measure. Accordingly, we generated a new test set according to the diffuse scattering model (Subsection 6.2.0 B.)

In this case, the measures for the three top measures are as shown in the following figure.

Measure	P_{ID}	P_{FA}
Greedy E-to-P	0.890+/-0.061	0.381+/-0.067
Diffusive Scatter	0.936+/-0.055	0.339+/-0.051
Bipartite	0.869+/-0.057	0.347+/-0.022

As shown, we re-ran the experiment using the baseline greedy measure, the bipartite measure and the diffusive scattering score measure. We see that indeed the diffusive scattering score now outperforms the others, as expected.

6.4 Experiment Conclusions

- The various methods work fairly similarly. I.e., performance levels don't vary that much. We manipulated parameters to attempt to highlight the` distinctions.
- Greedy methods are all pretty similar; when PID increases, PFA also increases some.
- Bipartite produces considerably better PID, at the expense of a somewhat higher PFA. This makes sense because…
- If features are generated according to the diffusive scattering model, then the diffusive scattering metric yields better performance, both in PID and PFA.
- Geometric hashing provided implementation efficiencies in terms of run-time.

Chapter 7. Experiments with SAR Signature Data

7.1 Introduction

In this Chapter, we present the results of our experiments using data from a real target recognition application. We first present a brief description of the MSTAR model-based target recognition system.

We have already discussed the MSTAR (Moving and Stationary Target Acquisition and Recognition) project in previous chapters. The MSTAR project has provided a public database of sample SAR imagery with a large collection of example vehicles. Over 200,000 target chips, and over 100 square kilometers of clutter data were collected during three supervised collections under the auspices of the DARPA MSTAR program. In this chapter, we describe experimental results of the matching algorithms applied to some selected MSTAR imagery.

7.2 MSTAR Target Recognition

SAR sensors have many advantages over electro-optical sensors for target recognition applications, such as range-independent resolution and superior performance under all-weather conditions.

The detection scenario begins with SAR imagery representing terrain in which an unknown number of targets of interest are deployed, each having an unknown location and pose. The objective is to maximize the target detection probability, subject to a constraint on the false alarm density. Targets are detected using a two-stage algorithm. In the first stage, a simple pre-screening algorithm (e.g. a two-parameter CFAR procedure) is applied to the imagery, thereby yielding a collection of regions of interest (ROIs) centered at possible target locations. In the second stage, point features are extracted from each ROI, and a decision is made for each whether the constellation of point extractions is consistent with the target hypothesis. Measuring this consistency entails searching through a collection of hypothesized constellations of peaks indexed by target type and pose; if a hypothesized constellation is found having point features that can be put in suitable good correspondence with the extracted features, then a target is declared to be present, and otherwise clutter is declared to be present.

The ideas and contributions described in this thesis were partially developed in conjunction with MSTAR, which is a U.S. government model-based vision Program, jointly sponsored by DARPA and AFRL AACA.

The purpose of the MSTAR system is to develop a demonstration of the capabilities of model-based vision to recognize and identify targets in Synthetic Aperture Radar

(SAR) data. The SAR images are nominally one-foot resolution taken from a high altitude aircraft. The targets are military vehicles and the MSTAR program is aggressively pursuing a process of increasing the number of possible target types and the conditions under which targets may be found. Early in the program ten target types were considered, with all targets in nominal configurations and "in the clear."

SAR imagery is a challenging object recognition environment because minor changes such as small rotations and slight configuration changes can have strong influences in the resulting SAR signature.

The author of this dissertation participated in a research effort at NYU as part of the MSTAR program. NYU was under contract from AFRL for the implementation of the Match Module as well as assisting in algorithm development of the Search module. In this subsection we will provide an overview of the entire MSTAR system in order to place the Search and Match modules in the proper context. The matching algorithms described in this thesis were in part inspired by the application needs of the MSTAR program.

Accordingly, after describing the MSTAR System, we present results using the matching approaches described earlier to specific MSTAR problems.

Finally, in light of all the work required for MSTAR system development and of the NYU contribution to this development, we provide a discussion of the system engineering process that stimulated the environment for matching research.

The system consists of two broad subsystems:

- A front end formerly known as FIX (Focus-of-Attention/Index) which attempts to reduce the amount of processing required by focusing on subsets of the Image Space and of the Target Hypothesis Space. This subsystem generates potential "Regions of Interest" in the input SAR image and a coarse set of initial candidate hypotheses for each of the ROI's.
- A back end known as PEMS (Prediction/Extraction/Match/Search) which explores the hypothesis space and refines the initial hypotheses estimates by iteratively generating online predictions about target appearance and matching them to features extracted from the measured image.

Input image data enters the system through the "Focus of Attention" module. The FOA module attempts to discard image locations that have no chance of containing a target, while focusing attention on all regions of interest that might conceivably contain a target. This module uses multiresolution processing in order to analyze local spatial frequencies (much like a wavelet transform) to ascertain whether the neighborhood has properties that allow it to be discarded from further consideration. Ideally, the FOA module discards large portions of the input image, and outputs relatively few "Regions of Interest" (ROI's) that contain all targets. An ROI will often contain "clutter" or some other non-target object; however, all instances of actual targets should be processed and become an enclosing ROI.

The Focus of Attention module sends its output to the "Indexing" (IX) module. This module produces a set of hypotheses for each Region of Interest. The possible hypotheses fall into two classes: a target model hypothesis, and "OTHER." The latter label indicates the possibility that the ROI does not contain a target of interest. All other hypotheses are target model hypotheses, and contain information specifying likely parameters for the identity, location, orientation, and other properties of the target. The IX module compares the ROI against a template that has been formed from previously collected data and which provides representative examples of true target signatures. The IX module essentially uses a cross-correlation (which is equivalent to mean square error), not of the grayscale data, but of "feature planes" representing nonlinear functionals applied to the data. One such functional consists of zero-crossings from the Laplacian of a Gaussian (LoG) applied to the image data. However, the mean square difference is modified by using a distance transform that compares, for example, zero-crossings in the observed data against the location of the nearest zero-crossing in the exemplar data. A contribution to a score is registered inversely according to this distance, with the largest contribution if the distance is zero. In order to build in some robustness, the penalty increases with distance to a certain point, at which point the penalty decreases. That is, spurious mismatches do not penalize as much as near mismatches.

The IX module produces an ordered list of hypotheses. Hypotheses can repeat the same target type indifferent locations and different orientations. One of the hypotheses can be “OTHER.” Typically, dozens or even hundreds of hypotheses can be generated, but only a few of them (10 or 15) will be considered by the subsequent subsystem.

The back-end of the System implements a hypothesize-and-test closed-loop mechanism for evaluation and refinement of target hypotheses.

A generic automatic object recognition system is capable of managing numerous hypotheses about potential targets. The system can operate on a single hypothesis by refining it (changing pose parameters, for example), rejecting it, or replacing it by multiple hypotheses. Further, newly generated hypotheses can be merged with other existing hypotheses that subsume them. Accordingly, the (search) system deals with a graph of hypotheses where certain nodes are considered live, and edges represent the dynamic heritage of hypotheses. Information in a hypothesis includes the identity of the object itself, as well as parameters specifying its three-dimensional location, pose, configuration, articulation, obscuration, surrounding and other environmental factors, and uncertainties associated with all of the above elements. The operations that the system performs at each step on a hypothesis can depend upon the current state of the system. Given all this flexibility, the challenge is both to design and to express the search logic that emulates cognitive reasoning that leads to a decision in the space of hypotheses. The reasoning process needs to be sufficiently flexible and transparent so as to permit easy adaptation to varying conditions [Mossing, Ross et. al. 1998]. Since the models and hypotheses involve 3D multi-part objects in a 3D world, the reasoning process needs to operate with knowledge of three-dimensional geometry and knowledge of semantic parts of models. The procedure involves both symbolic processing, with model parts and objects, and quantitative processing, with likelihood and match scores.

The PEMS subsystem consists of a searching engine driving the hypotheses evidence accrual process in a hypothesize-and-test environment, using an on-line prediction module and a matching module. PEMS operates as a transformation function in hypotheses space which inspects and probes elements of this space and generates new elements (likely hypotheses) until some stopping criterion is reached, at which point a decision is made about the identity of the object(s) present in each ROI.

The purpose of the Match module is to provide a measure of similarity between extracted and predicted signatures and to suggest matching improvements by strictly local parameter refinement.

7.3 Experiment Design

Experiment Goals

The purpose of these experiments is to assess the feasibility of many alternative matching algorithms and to compare different matching schemes, as well as to provide an objective analysis of the sensitivity of various parametric assumptions under a subset of operating conditions.

Experiment Data

The test data set consists of:

- 475 measured target chips from nineteen ground-order-of-battle tactical vehicles, and
- 323 clutter chips containing various sparsely built-up and built-up environments that have been detected as target-like objects by the prescreening modules (i.e., FIX) of MSTAR.

This data set is representative of small deviations from standard operating conditions and training data. Training is involved because the hash tables were constructed from databases of previously computed predictions. There is no online prediction involved in this process.

7.4 Experiment Results

MEASURE	P_{ID}	FA
Greedy E-to-P	0.610	0.289
Greedy E-to-P 1-1	0.596	0.232
Greedy P-to-E	0.553	0.219
Greedy P-to-E 1-1	0.576	0.237
Many-to-All	0.622	0.239
Diffusive Scatter	0.622	0.239
Hausdorff 2-sided	0.591	0.271
Bipartite	0.571	0.251

Results are presented in the table shown above. The P_{ID} value is measured using a percentage of correctly identified target chips among the 475 chips of true target. We see immediately that correct identification rates are much lower than those observed with simulated data, as reported in section 6.3. The false alarm rates give the percentage of the trials that a clutter chip (among the 323) is identified as a true target, of one sort or another. The false alarm rates are lower than those seen with the simulated data.

Among the different schemes, the performance levels vary only slightly, although the bipartite method no longer yields the highest identification rates. The Greedy E-to-P and the Many-to-All methods both yield among the best identification rates, with a lower false alarm rate for the Many-to-All scheme. The diffusive scattering and Many-to-All schemes yield identical results because they are scaled versions of one another.

$$\mathbf{S}_{diffusive}\big(\mathbf{D}(\mathbf{X},\mathbf{Y})\big) = \log\big(\exp(-R)\big) + \sum_j \log\left(\frac{l_0(\mathbf{x}_j)}{g_0(\mathbf{x}_j)} + k \cdot \sum_i q_i \cdot \frac{g_i(\mathbf{x}_j - \mathbf{y}_i)}{g_0(\mathbf{x}_j)} \right)$$

and

$$\mathbf{S}_{M-to-A}\big(\mathbf{D}(\mathbf{X},\mathbf{Y})\big) = \sum_j \log\left(\frac{l_0}{l_0 + \sum_r q_r} \cdot \frac{1}{g_0(\mathbf{x}_j)} + \sum_i \left(\frac{q_i}{l_0 + \sum_r q_r} \right) \cdot \frac{g_i(\mathbf{x}_j - \mathbf{y}_i)}{g_0(\mathbf{x}_j)} \right)$$

Note that with these results, there have been only 475 plus 323 executions of the system for each method, and that furthermore, with certain target chips, recognition is hopeless because the extended operating conditions makes the observed signature too distant from the correct pre-stored prediction. This is a different situation from the simulation studies of Chapter 6, where each scoring method was executed 1000 times against carefully controlled test signatures.

The extended operating conditions explain the lower identification rates. The lower false alarm rates are explained by the fact that many of the 323 clutter chips are considerably different from targets, even though the MSTAR front end (FIX) deemed them worthy of further study by the back end.

7.5 Experiment Conclusions

Limited conclusions are possible from the studies performed on MSTAR data.

- Peak Features modeling in MSTAR is limited by sensor resolution: Empirical studies have shown that several scatterers can collude in a single resolution cell, causing scintillation and making it difficult to differentiate and track the origin of a peak feature in the image. The search for more discriminative image features is still ongoing. Extended attributed peak features constitute a promising alternative that has not been explored in this work.

- Limited set of modeling conditions in GH database: Our Geometric Hashing implementation is only an approximate solution due to the fact that only a partial subset of operating conditions can be stored in the model database, specifically

    - Approximate geometry: elevation and squint corrections;
    - Approximate sampling of scattering centers;
    - Nominal model configurations (SOC).

- Use of cues and contextual information to mitigate False Alarms by PEMS in the real system: The MSTAR system employs a complex search logic that incorporates contextual cues, terrain database modeling, Image Analysis heuristics, knowledge-based information and domain-specific heuristics. Any target recognition system should involve such enhancements.

- Complex interaction of elements, diverse technology and expertise from multidisciplinary backgrounds: These are required in order to achieve even modest performance levels.

Chapter 8. Conclusions

We have presented a complete theory for model-based feature matching in the presence of uncertainty, and we have demonstrated the robustness of the approach in realistic applications in automatic target recognition under highly unconstrained image analysis scenarios.

The system performance is improved by the use of novel match quality measures, used in conjunction with a Bayesian posterior expected utility to quantify the support for model hypotheses. The matching scores are also used successfully to prioritize search strategies and find the most promising directions for hypothesis generation in complex systems involving hundreds of models.

In summary, we have observed improved discrimination performance, false alarm reduction and a quantitative measure of the reliability of the system.

We have also obtained asymptotic results for performance bounds using synthetically generated models, and we have verified the results are consistent with the theory.

References

[Atjai et al. 1984] Atjai, M., Komlós, J. and Tusnády, G. On Optimal Matchings. *Combinatorica*, **4**: 259—264, 1984.

[Aráoz and Edmonds 1985] Aráoz, J. and Edmonds, J. A Case of Non-Convergent Dual Changes in Assignment Problems. *Discrete Applied Mathematics*, **11**: 95—102, 1985.

[Arora et al. 1996] Arora, S., Frieze, A. and Kaplan, H. A New Rounding Procedure for the Assignment Problem with Applications to Dense Graph Arrangement Problems. Annual IEEE Symposium on Foundations of Computer Science, 1—30, 1996.

[Avis 1983] Avis, D. A Survey of Heuristics for the Weighted Matching Problem. *Networks*, **13**: 475—493, 1983.

[Ayache and Faugeras 1988] Ayache, N. and Faugeras, O. HYPER: A New Approach for the Recognition and Positioning of Two-Dimensional Objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **8**(1): 44—54, 1988.

[Baird 1985] Baird, H.S. *Model-Based Image Matching Using Location*. MIT Press, Cambridge, 1985.

[Ball and Derigs 1983] Ball, M.O. and Derigs, U. An Analysis of Alternative Strategies for Implementing Matching Algorithms. *Networks*, **13**: 517—549, 1983.

[Ballard and Brown 1982] Ballard, D.H. and Brown, C. *Computer Vision*. Prentice Hall, Englewood Cliffs, 1982.

[Basri and Weinshall 1993] Basri, R. and Weinshall, D. Distance Metric between 3D Models and 2D Images for Recognition and Classification. Technical Memorandum, MIT Artificial Intelligence Laboratory, 1993.

[Berger 1985] Berger, J.O. *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag, New York, 1985.

[Besag 1989] Besag, J.E. Towards Bayesian Image Analysis. *Journal of Applied Statistics*, **16**, 395—407, 1989.

[Besag and Green 1993] Besag, J.E. and Green, P.J. Spatial Statistics and Bayesian Computation. *Journal of the Royal Statistical Society*, **B 55**: 25—37, 1993.

[Besl and Jain 1985] Besl, P.J. and Jain, R.C. Three-Dimensional Object Recognition. *ACM Computing Surveys*, **17**(1): 75—145, 1985.

[Bhanu et al. 1997] Bhanu, B., Zelnio, E.G., Dudgeon, D.E., Rosenfeld, A., Casasent, D. and Reed, I.S. Special Issue on Automatic Target Detection and Recognition. IEEE Transactions on Image Processing, **6**(1), 1997.

[Biedermann 1985] Biederman, I. Human Image Understanding: Recent Research and a Theory. *Computer Vision, Graphics and Image Processing*, **32**: 29—73, 1985.

[Billingsley 1968] Billingsley, P. *Convergence of Probability Measures*. John Wiley and Sons, New York, 1968.

[Binford 1982] Binford, T.O. Survey of Model-Based Image Analysis Systems. *International Journal of Robotics Research*, **1**(1): 18—64, 1982.

[Binford and Levitt 1983] Binford, T.O. and Levitt, T.S. Quasi-Invariants: Theory and Exploitation. *Proceedings of DARPA Image Understanding Workshop*, 819—830, 1993.

[Binford et al. 1989] Binford, T.O., Levitt, T.S. and Mann, W.B. Bayesian Inference in Model-Based Vision. *Uncertainty in Artificial Intelligence* **3** (L.N. Kanal, T.S. Levitt, J.F. Lemmer, editors). Elsevier Science Publishers, New York, 1989.

[Bolles and Horaud 1986] Bolles, R.C. and Horaud, P. A Three-Dimensional Part Orientation System. *International Journal of Robotics Research*, **5**(3): 3—26, 1986.

[Box and Tiao 1973] Box, G.E.P. and Tiao, G.C. *Bayesian Inference in Statistical Analysis*. Addison-Wesley Publishing Co., Reading, 1973.

[Brooks 1983] Brooks, R.A. Model-Based Three-Dimensional Interpretations of Two-Dimensional Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **5**(2): 140—150, 1983.

[Brown 1967] Brown, W.M. Synthetic Aperture Radar. *IEEE Transactions in Aerospace and Electronic Systems*, AES-**3** (2): 217—229, 1967.

[Casey and Lecolinet 1996] Casey, R.G. and Lecolinet, E. A Survey of Methods and Strategies in Character Segmentation. *IEEE Transactions in Pattern Analysis and Machine Intelligence*, **18**: 690—706, 1996.

[Cass 1997] Cass, T.A. Polynomial-Time Geometric Matching for Object Recognition. *International Journal of Computer Vision*, **21**(1): 37—61, 1997.

[Cherkassky et. al. 1998] Cherkassky, B.V., Goldberg A.V., Martin, P., Setubal, J.C. and Stolfi, J. Augment or Push? A Computational Study of Bipartite Matching and Unit Capacity Flow Algorithms. Technical Report 98-036R, NEC Research Institute, 1998.

[Chin and Dyer 1986] Chin, R.T. and Dyer, C.R. Model-Based Recognition in Robot Vision. *ACM Computing Surveys*, **18**(1): 67—108, 1986.

[Cipolla and Pentland 1998] Cipolla, R. and Pentland, A. *Computer Vision for Human-Machine Interaction*. Cambridge University Press, New York, 1998.

[Cox et al. 1995] Cox, I.J., Rehg, J.M., Hingorani, S.L. and Miller, M.L. Grouping Edges: An Efficient Bayesian Multiple Hypothesis Approach. *Partitioning Data Sets*. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, **19**: 199—235, 1995.

[Cox et al. 1996] Cox, I.J., Ghosn, J. and Yianilos, P.N. Feature-Based Face Recognition Using Mixture Distance. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 209—216, 1996.

[Crevier and Lepage 1997] Crevier, D. and Lepage, R. Knowledge-Based Image Understanding Systems: A Survey. *Computer Vision and Image Understanding*, **67**(2): 161—185, 1997.

[Cross and Hancock 1998] Cross, A.D.J. and Hancock, E.R. Graph Matching with a Dual-Step Expectation Maximization Algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**(11), 1998.

[Dantzig 1993] Dantzig, G.B. *Linear Programming and Extensions*. Princeton University Press. Princeton, 1993.

[Davis and Farid 1996] Davis, M.H.A., and Farid, M. A Target Recognition Problem: Sequential Analysis and Optimal Control. *SIAM Journal on Control and Optimization*, **34**(6): 2116—2132, 1996.

[Dempster 1998] Dempster, A.P. Logicist Statistics I. Models and Modeling. *Statistical Science*, **13**(3), 248—276, 1998.

[Derigs and Metz 1992] Derigs, U. and Metz, A. On the Construction of the Set of Best Matchings and its Use in Solving Constrained Matching Problems. *Combinatorial Optimization: New Frontiers in Theory and Practice*, Springer-Verlag, New York, 1992.

[Devijver and Kittler 1982] Devijver, P. and Kittler, J. *Pattern Recognition: A Statistical Approach.* Prentice Hall, Englewood Cliffs, 1982.

[Devroye 1986] Devroye, L. *Non-Uniform Random Variable Generation*. Springer-Verlag, Berlin, 1986.

[Devroye et al. 1996] Devroye, L., Györfi, L. and Lugosi, G. *A Probabilistic Theory of Pattern Recognition.* Springer-Verlag, New York, 1996.

[Doob 1953] Doob, J.L. *Stochastic Processes*. John Wiley and Sons, New York, 1953.

[Draper 1995] Draper, D. Assessment and Propagation of Model Uncertainty. *Journal of the Royal Statistical Society*, **B 57**: 45—97, 1995.

[Duda and Hart 1973] Duda, R.O. and Hart, P.E. *Pattern Classification and Scene Analysis.* John Wiley and Sons, New York, 1973.

[Duda et al. 1999] Duda, R.O., Hart, P.E. and Stork, D.G. *Pattern Classification.* John Wiley and Sons, New York, Second Edition, 1999. In Press.

[Dudgeon and Lacoss 1993] Dudgeon, D.E. and Lacoss, R.T. An Overview of Automatic Target Recognition. *Lincoln Laboratory Journal*, **6**(1): 3—10, 1993.

[Ettinger et al. 1996] Ettinger, G.J., Klanderman, G.A., Wells, W.M. and Grimson, W.E.L. A Probabilistic Optimization Approach to SAR Feature Matching. *Algorithms for Synthetic Aperture Radar Imagery III*, *Proceedings of the SPIE* **2757**, 318—329, 1996.

[Faugeras 1993] Faugeras, O. Three-dimensional Computer Vision: a Geometric Viewpoint. MIT Press, Cambridge, 1993.

[Fischler and Firschein 1987] Fischler, M. and Firschein, O. Readings in Computer Vision: Issues, Problems, Principles and Paradigms. Morgan Kauffman, San Mateo, 1987.

[Garcia and Hummel 1997] Garcia, M. and Hummel, R.A. Use of Bipartite Matching and Combinatorial Optimization Algorithms for Object Recognition. Technical Memorandum, New York University, 1997.

[Geiger et al. 1997] Geiger, D., Parida, L. and Hummel, R. A. A Multijunction Detector Using the Minimum Description Length Principle. *Proceedings of the First International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, 1997.

[Gleser 1998] Gleser, L.J.A. Assessing Uncertainty in Measurement. *Statistical Science*, **13**(3): 277—290, 1998.

[Goldberg and Kennedy 1993] Goldberg, A.V. and Kennedy, R. An Efficient Cost-Scaling Algorithm for the Assignment Problem. Technical Report, Stanford University, 1993.

[Goldberg et al. 1993] Goldberg, A.V., Plotkin, S.A. and Vaidya, P.M. Sublinear-Time Parallel Algorithms for Matching and Related Problems. *Journal of Algorithms*, **14**: 180—213, 1993.

[Grenander 1981] Grenander, U. *Abstract Inference*. John Wiley and Sons, New York, 1981.

[Grenander 1995] Grenander, U. *General Pattern Theory.* Oxford University Press, Oxford, 1995.

[Grimson 1990a] Grimson, W.E.L. *Object Recognition by Computer: The Role of Geometric Constraints.* MIT Press, Cambridge, 1990.

[Grimson 1990b] Grimson, W.E.L. The Combinatorics of Heuristic Search Termination for Object Recognition in Cluttered Environments. *Proceedings of the First European Conference on Computer Vision*, 1990.

[Grimson and Huttenlocher 1990] Grimson, W.E.L. and Huttenlocher, D.P. On the Sensitivity of the Hough Transform for Object Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**(3): 255—274, 1990.

[Grimson et al. 1991] Grimson, W.E.L., Huttenlocher, D.P. and Jacobs, D.W. Affine Matching with Bounded Sensor Error: A Study of Geometric Hashing and Alignment. Technical Report 1250, MIT Artificial Intelligence Laboratory, 1991.

[Grötschel and Lovász 1993] Grötschel, M. and Lovász, L. Combinatorial Optimization: a Survey. DIMACS Technical Report 93-29, 1993.

[Halmos 1950] Halmos, P.R. *Measure Theory.* Van Nostrand, New York, 1950.

[Ho and Chelberg 1998] Ho Yi, J. and Chelberg, D.M. Model-Based 3D Object Recognition Using Bayesian Indexing. *Computer Vision and Image Understanding,* **69**(1): 87—105, 1998.

[Horn 1986] Horn, B.K.P. *Robot Vision.* MIT Press, Cambridge, 1986.

[Huber 1981] Huber, P.J. *Robust Statistics*. John Wiley and Sons, New York, 1981.

[Hummel 1995] Hummel, R.A. Performance Predictions for Generic ATR Systems: Requirements on the Number of Features. Technical Memorandum, New York University , 1995.

[Hummel 1996a] Hummel, R.A. Object Recognition Research: Matched Filtering Becomes Bayesian Pattern Matching. *Advances in Image Understanding: A Festschrift for Azriel Rosenfeld* (K. Bowyer, N. Ahuja, editors), IEEE Computer Society Press, 1996.

[Hummel 1996b] Hummel, R.A. Uncertainty Reasoning in Object Recognition by Image Processing. *Reasoning with Uncertainty in Robotics* (L. Dorst, M. van Lambalgan, F. Voorbraak, editors), Royal Netherlands Academy of Science, Amsterdam, 1996.

[Hummel and Gorman 1996] Hummel, R.A. and Gorman, J.D. Interim Report for the MSTAR MCI Match Module. Technical Report WL-TR-97-1101, New York University, 1996.

[Hummel and Landy 1988] Hummel, R.A. and Landy, M.S. A Statistical Viewpoint on the Theory of Evidence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **10**(2): 235—247, 1988.

[Huttenlocher et al. 1993] Huttenlocher, D.P., Klanderman, G.A. and Rucklidge, W.J. Comparing Images Using the Hausdorff Distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **15**(9): 850—863, 1993.

[Illingwoth and Kittler 1988] Illingwoth, J. and Kittler, J. A Survey of the Hough Transform. *Computer Vision, Graphics and Image Processing*, **44**: 87—116, 1988.

[Irving 1997] Irving, W.W. A Clarification of the Relationship Assumed by Match between Predicted and Extracted Features. Technical Memorandum, Alphatech Inc., 1997.

[Irving et al. 1997] Irving, W.W., Wissinger, J.W., Ettinger, G.J., Chaney, R., Klanderman, G.A. Analysis of ATR Performance Degradation in Presence of Bayesian Modeling Errors. Technical Memorandum, Alphatech, Inc., 1997.

[Ishikawa and Geiger 1998] Ishikawa, H. and Geiger, D. Segmentation by Grouping Junctions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1998.

[Jain and Jain 1990] Jain, R.C. and Jain, A.K. Analysis and Interpretation of Range Images. Springer-Verlag, New York, 1990.

[Johnson and Hebert 1977] Johnson, A.E. and Hebert, M. Recognizing Objects by Matching Oriented Points. Technical Report RI-TR-96-04, Carnegie-Mellon University, Pittsburgh, 1996.

[Johnson and Kotz 1977] Johnson, N.L. and Kotz, S. Urn Models and Their Application. John Wiley and Sons, New York, 1977.

[Kalvin 1991] Kalvin, A.D. Segmentation and Surface-Based Modeling of Objects in Three-Dimensional Biomedical Images. Ph.D. Thesis, New York University, UMI Research Press, Ann-Arbor, 1991.

[Kanade 1977] Kanade, T. Computer Recognition of Human Faces. Birkhäuser Verlag, Stuttgart, 1977.

[Kanade et al. 1994] Kanade, T. Hebert, M. and Kweon, I. 3-D Vision Techniques for Autonomous Vehicles. Proceedings of DARPA Image Understanding Workshop, 1994.

[Kanai and Baird 1998] Kanai, J. and Baird, H.S. Special Edition on Document Image Understanding and Retrieval. Computer Vision and Image Understanding, **70**(3), 1998.

[Kanatani 1993] Kanatani, K. Geometric Computation for Machine Vision. Oxford University Press, Oxford, 1993.

[Kass et al. 1987] Kass, M., Witkin, A. and Terzopolous, D. Snakes: Active Contour Models. International Journal of Computer Vision **1**: 321—331, 1987.

[Keydel and Lee 1996] Keydel, E.R. and Lee, S.W. Signature Prediction for Model-Based Automatic Target Recognition. *Algorithms for Synthetic Aperture Radar Imagery III, Proceedings of the SPIE* **2757**: 306—317, 1996.

[Kim and Kak 1991] Kim, W.Y. and Kak, A.C. 3D Object Recognition Using Bipartite Matching Embedded in Discrete Relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13**(3): 224—251, 1991.

[Knerr et al. 1998] Knerr, S., Augustin, E., Baret, O. and Price, D. Hidden Markov Model Based Word Recognition and Its Application to Legal Amount Reading on French Checks. *Computer Vision and Image Understanding*, **70**(3): 404—419, 1998.

[Knuth 1973] Knuth, D.E. *The Art of Computer Programming*, **1-3**. Addison-Wesley Publishing Co., Reading, 1973.

[Koopmans and Beckman 1957] Koopmans, T. and Beckman, M. Assignment Problems and the Location of Economic Activities. *Econometrica*, **25**: 53—76, 1957.

[Kotz and Johnson 1988] Kotz, S. and Johnson, N.L. *Encyclopedia of Statistical Sciences*. John Wiley and Sons, New York, 1988.

[Kuhn 1955] Kuhn, H.W. The Hungarian Method for the Assignment Problem. *Naval Research Logistic Quarterly*, **2**: 83—97, 1955,

[Lamdan 1989] Lamdan, Y. *Object Recognition by Geometric Hashing*. Ph.D. Thesis, New York University, UMI Research Press, Ann Arbor, 1989.

[Lamdan et al. 1988a] Lamdan, Y., Schwartz, J.T. and Wolfson, H.J. Affine Invariant Model-Based Object Recognition. *IEEE Transactions on Robotics and Automation*, **6**: 238—249, 1988.

[Lamdan et al. 1988b] Lamdan, Y., Schwartz, J.T. and Wolfson, H.J. On Recognition of 3D Objects from 2D Images. *Proceedings of the IEEE International Conference on Robotics and Automation*, **3**: 1407—1413, 1988.

[Le Cam 1986] Le Cam, L. Asymptotic Methods in Statistical Decision Theory. Springer-Verlag, New York, 1986.

[Lehmann 1983] Lehmann, E.L. Theory of Point Estimation. John Wiley and Sons, New York, 1983.

[Lehmann 1985] Lehmann, E.L. Testing Statistical Hypotheses. John Wiley and Sons, New York, 1985.

[Levitt 1986] Levitt, T.S. Model-Based Probabilistic Inference in Hierarchical Hypothesis Spaces. Uncertainty in Artificial Intelligence **1** (L.N. Kanal, J.F. Lemmer, editors). Elsevier Science Publishers, New York, 1986.

[Lindenstrauss and Tzafriri 1977] Lindenstrauss, J. and Tzafriri, L. Classical Banach Spaces, **I**. Springer-Verlag, Berlin, 1977.

[Lindenstrauss and Tzafriri 1979] Lindenstrauss, J. and Tzafriri, L. Classical Banach Spaces, **II**. Springer-Verlag, Berlin, 1979.

[Liu 1995] Liu, J.J. *Model-Based Three-Dimensional Object Recognition Using Geometric Hashing with Attributed Features*. Ph.D. Thesis, New York University, UMI Research Press, Ann Arbor, 1995.

[Liu and Hummel 1995] Liu, J.J. and Hummel, R.A. Geometric Hashing with Attributed Features. Technical Report, New York University, 1995.

[Liu et al. 1996] Liu, T.L., Geiger, D., Donahue, D., and Hummel, R.A. Sparse Representations for Image Decomposition with Occlusions. *Proceedings of the Fourth European Conference on Computer Vision*, 556—565, 1996.

[Lovász and Plumer 1986] Lovász, L. and Plumer, M.D. *Matching Theory*. Elsevier Science Publishers. New York, 1986.

[Lowe 1985] Lowe, D.G. *Perceptual Organization and Visual Recognition.* Kluwer Academic, Hingham, 1985.

[Lowe 1987] Lowe, D.G. Three-dimensional Object Recognition from Single Two-dimensional Images. *Artificial Intelligence*, 31: 355—395, 1987.

[Marr 1982] Marr, D. *Vision: a Computational Investigation into the Human Representation and Processing of Visual Information.* W.H. Freeman and Company, San Francisco, 1982.

[Minc 1978] Minc, H. *Permanents*. Addison-Wesley, Reading, 1978.

[Moravec 1980] Moravec, H.P. *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover.* Ph.D. Thesis, Artificial Intelligence Laboratory, Stanford University, UMI Research Press, Ann Arbor, 1980.

[Moravec 1981] Moravec, H.P. *Robot Visual Navigation.* Robotics Institute, Carnegie-Mellon University, Pittsburgh, 1981.

[Moravec 1998] Moravec, H.P. *Robot: Mere Machine to Transcendent Mind.* Oxford University Press, New York, 1998.

[Morgan 1992] Morgan, D.R. Point Feature Detection Algorithm: Explicit State Integral Version. Technical Memorandum 06327-025, Advanced Design Systems, 1992.

[Morgan et al. 1995] Morgan, D.R. Ettinger, G.J., Hummel, R.A. and Wissinger, J.W. Probability and Uncertainty for MSTAR. Technical Memorandum, AFRL, 1995.

[Morgan et al. 1996] Morgan, D.R. Chong, C.Y. and Fung, R. Investigation of Algorithmic Approaches to MSTAR Evidence Accrual for Peak Features. Technical Report, Booz-Allen and Hamilton, 1996.

[Mossing and Ross 1997] Mossing, J.C. and Ross, T.D. An Evaluation of SAR ATR Algorithm Performance Sensitivity to MSTAR Extended Operating Conditions. Algorithms for Synthetic Aperture Radar Imagery IV, Proceedings of the SPIE **3070**, 1997.

[Mossing and Ross 1998] Mossing, J.C. and Ross, T.D. MSTAR Evaluation breaks new ground: Methodology, Results, Infrastructure and Data Analysis. Algorithms for Synthetic Aperture Radar Imagery V, Proceedings of the SPIE **3370**, 318—329, 1998.

[Mumford 1994] Mumford, D. Pattern Theory: A Unifying Perspective. *Proceedings of the First European Congress in Mathematics*. Birkhaüser, Berlin, 1994.

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

[Murty 1968] Murty, K.G. An Algorithm for Ranking All Assignments in Order of Increasing Cost. *Operations Research,* **16**: 682—686, 1968.

[Ohta 1985] Ohta, Y. *Knowledge-Based Interpretation of Outdoor Natural Color Scenes*. Pitman Advanced Publishing, Boston 1985.

[Olson 1998] Olson, C.F. A Probabilistic Formulation for Hausdorff Matching. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 150—156, 1998.

[Pardalos and Wolkowicz 1995] Pardalos, P.M. and Wolkowicz, H. *Quadratic Assignment and Related Problems*. American Mathematical Society, Providence, 1995.

[Pearl 1988] Pearl, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kauffman Publishers, San Mateo, 1988.

[Pentland 1986] Pentland, A.P. *From Pixels to Predicates: Recent Advances in Computational and Robotic Vision*. Ablex Publishing Corp., Norwood, 1986.

[Pentland 1990] Pentland, A.P. Automatic Extraction of Deformable Part Models. *International Journal of Computer Vision,* **4**:107—126, 1990.

[Petkôvsek et al. 1996] Petkôvsek, M., Wilf, H.S. and Zeilberger, D. *A equals B*. A.K. Peters, Wellesley, 1996.

[Poggio 1985] Poggio, T. Early Vision: From Computational Structure to Algorithms and Parallel Hardware. *Computer Vision, Graphics and Image Processing,* **31**: 139—155, 1985.

[Poore 1995] Poore, A.B. Multidimensional Assignment and Multitarget Tracking. Partitioning Data Sets. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, **19**: 169—196, 1995.

[Pressman and Sonin 1990] Pressman, E.L. and Sonin, I.N. Sequantial Control with Incomplete Information. Academic Press, New York, 1990.

[Reid 1979] Reid, D.B. An Algorithm for Tracking Multiple Targets. IEEE Transactions on Automatic Control, AC-**24**(6): 843—854, 1979.

[Richardson and Green 1997] Richardson, S. and Green, P.J. On Bayesian Analysis of Mixtures with an Unknown Number of Components. Journal of the Royal Statistical Society, **B 59**, 1997.

[Rigoutsos 1992] Rigoutsos, I. Massively Parallel Bayesian Object Recognition. Ph.D. Thesis, New York University, UMI Research Press, Ann Arbor, 1992.

[Rigoutsos and Hummel 1990] Rigoutsos, I. and Hummel, R.A. Scalable Parallel Geometric Hashing for Hypercube Architectures. Technical Report, New York University, 1990.

[Rigoutsos and Hummel 1995] Rigoutsos, I. and Hummel, R.A. A Bayesian Approach to Model Matching with Geometric Hashing. Computer Vision and Image Understanding, **62**(1): 11—26, 1995.

[Riordan 1968] Riordan, J. *Combinatorial Identities*. John Wiley and Sons, New York, 1968.

[Rosenfeld 1998] Rosenfeld, A. *Vision Bibliography. 27 Feb 1998.*
http://iris.usc.edu/Vision-Notes/rosenfeld/

[Ross et al. 1997] Ross, T.D., Westerkamp, L.A., Zelnio, E.G. and Burns, T.J. Extensibility and Other Model-Based ATR Evaluation Concepts. Algorithms for Synthetic Aperture Radar Imagery IV, Proceedings of the SPIE **3070**, 1—10, 1997.

[Ross et al. 1998] Ross, T.D., Worrell, S.W., Velten, V.J., Mossing, J.C. and Bryant, M.L. Standard SAR ATR Evaluation Experiments Using the MSTAR Public Release Data Set. Algorithms for Synthetic Aperture Radar Imagery V, Proceedings of the SPIE **3370**, 1998.

[Ryan and Egaas 1996] Ryan, T.W. and Egaas, B. SAR Target Indexing with Hierarchical Distance Transforms. Algorithms for Synthetic Aperture Radar Imagery III, Proceedings of the SPIE **2757**, 294—305, 1996.

[Sarachik 1992] Sarachik, K.B. Limitations of Geometric Hashing in the Presence of Gaussian Noise. Technical Report 1395, MIT Artificial Intelligence Laboratory, 1992.

[Sinai 1992] Sinai, Y.G. *Probability Theory*. Springer-Verlag, Berlin, 1992.

[Stein and Medioni 1992] Stein, F. and Medioni, G. Structural Indexing: Efficient 3-D Object Recognition. *IEEE Transactions on Pattern aAnaluysis and Machine Intelligence*, **14**(2): 125—145, 1992.

[Strat 1992] Strat, T. *Natural Object Recognition*. Springer-Verlag, New York, 1992.

[Suetens et al. 1992] Suetens, P., Fua, P. and Hanson, A.J. Computational Strategies for Object Recognition. *ACM Computing Surveys*, **24**(1): 5—61, 1992.

[Tamburino 1996] Tamburino, L.A. 1996. Metrics and algorithms for matching in MSTAR. Technical Report, Air Force Research Laboratory, 1996.

[Tanimoto 1995] Tanimoto, S. The Elements of Artificial Intelligence. W.H. Freeman and Company, New York, 1995.

[Tanner 1996] Tanner, M.A. Tools for Statistical Inference: Methods for the Exploration of Posterior Distributions and Likelihood Functions. Springer-Verlag, New York, 1996.

[Therrien 1992] Therrien, C.W. Decision, Estimation and Classification. John Wiley and Sons, New York, 1992.

[Tierney 1994] Tierney, L. Markov Chains for Exploring Posterior Distributions. Annals of Statistics, **22**: 1701—1762, 1994.

[Tsai 1993] Tsai, F.C. A Probabilistic Approach to Geometric Hashing Using Line Features. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 393—399, 1993.

[Turk and Pentland 1991] Turk, M.A., and Pentland, A.P. Face Recognition Using Eigenfaces. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 586—591, 1991.

[Udupa 1989] Udupa, J.K. Computer Aspects of 3-D Imaging in Medicine. *3D Imaging in Medicine* (Udupa, J.K. and Herman, G.T.) Lewis Publishers, Chelsea, 1989.

[Ullman 1996] Ullman, S. *High-Level Vision: Object Recognition and Visual Cognition.* MIT Press, Cambridge, 1996.

[Ullman and Basri 1991] Ullman, S. and Basri, R. Recognition by Linear Combination of Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **13**(10): 992—1006, 1991.

[Velten 1998] Velten, V. Geometric Invariance for Synthetic Aperture Radar Sensors. Technical Report, Air Force Research Laboratory, 1998.

[Viola and Wells 1995] Viola, P. and Wells, W.M. Alignment by Maximization of Mutual Information. Proceedings of the International Conference on Computer Vision, 1995.

[Wasserman and Kadane 1990] Wasserman, L.A. and J.B. Kadane. Bayes Theorem for Choquet Capacities. The Annals of Statistics, **18**(3): 1328—1339, 1990.

[Weiss 1988] Weiss, I.P. Projective Invariants of Shapes. Proceedings of DARPA Image Understanding Workshop, 1125—1134, 1988.

[Winston 1992] Winston, P.H. Artificial Intelligence. Addison-Wesley Publishing Co., Reading, 1992.

[Wissinger et al 1996] Wissinger, J.W., Washburn, R.B., Friedland, N.S., Nowicki, A., Morgan, D.R., Chong, C.Y., and Fung, R. Search Algorithms for Model-based SAR ATR. Algorithms for Synthetic Aperture Radar Imagery III, Proceedings of the SPIE **2757**, 279—293, 1996.

[Wissinger et al 1999] Wissinger, J.W., Diemunsch, J., Ristroph, R.G., Severson, W.E., and Freudenthal, E.A. MSTAR Extensible Search Engine and Model-Based Inference Toolkit. Algorithms for Synthetic Aperture Radar Imagery VI, Proceedings of the SPIE, 1999. In Press.

[Wolfson 1990] Wolfson, H.J. Model-Based Object Recognition by Geometric Hashing. Proceedings of the First European Conference in Computer Vision, 526—536, 1990.

[Wolfson and Hummel 1988] Wolfson, H.J. and Hummel, R.A. Affine Invariant Matching. Proceedings of DARPA Image Understanding Workshop, 1988.

[Wolfson et al. 1991] Wolfson, H.J., Schonberg, E.J., Kalvin, A.D. and Lamdan, Y. Solving Jigsaw Puzzles by Computer Vision. *Annals of Operations Research*, **12**(1): 51—64, 1991.

[Yuille 1989] Yuille, A. Generalized Deformable Templates, Statistical Physics, and Matching Problems. *Neural Computation,* **2**: 1—24, 1989.

[Zelnio 1992] Zelnio, E.G. ATR Paradigm Comparison with Emphasis on Model-Based Vision. Model-Based Vision Development Tools, Proceedings of the SPIE **1609**: 2—15, 1992.