A Quantitative Approach to Perceptual Grouping in Computer Vision

Arnon Amir
A Quantitative Approach to Perceptual Grouping
in Computer Vision

Research thesis

Submitted in partial fulfillment of the requirements
for the degree of doctor of science

Arnon Amir

Submitted to the Technion — Israel Institute of Technology
Iyar, 5757 Haifa May, 1997
The work described herein was supervised by Dr. Michael Lindenbaum and by Prof. Alfred Bruckstein under the auspices of the Computer Science Department.

The generous financial help of the graduate school is gratefully acknowledged.

I want to thank Micha and Freddy for giving me the credit and the freedom to choose my way, and for their great support along this way. I own a special thank to Angelica, my wife, for being there with me.
# Contents

Abstract iii

1 Introduction 1
   1.1 Gestalt Psychology .............................................. 2
   1.2 Perceptual Organization in Machine Vision .................... 6
       1.2.1 Motivation .................................................. 7
       1.2.2 A Perspective Overview .................................... 10
       1.2.3 Grouping Cues .............................................. 14
       1.2.4 The Non-Accidentalness Criterion ......................... 16
       1.2.5 Grouping Mechanisms ...................................... 17
   1.3 The Contribution of This Work .................................. 17

2 The Grouping Task and its Graph Representation 21
   2.1 The Grouping Task ............................................... 21
   2.2 Grouping Cues ................................................... 22
   2.3 Representing Groups and Cues Using Graphs .................... 25

3 The Generic Grouping Method 27
   3.1 The User Decisions: Cue and Connectivity Selection .......... 27
   3.2 First Stage: Evaluating the Grouping Cues .................... 28
   3.3 Second Stage: Maximum Likelihood Graph Clustering (MLGC) ... 29
   3.4 The Grouping Likelihood Graph ................................ 31

4 Analysis of the Grouping Method 33
   4.1 Analysis of the Grouping Quality ............................... 33
       4.1.1 A Complete Underlying Graph ............................. 36
       4.1.2 Locally Dense Underlying Graphs ......................... 38
   4.2 Complexity Analysis of The MLGC Criterion .................. 40
       4.2.1 Related Work on Graph Clustering Methods ............... 41
       4.2.2 The MLGC criterion is NP-hard ........................... 45
5 The Cue Enhancement Procedure

5.1 Overview

5.2 Wald’s SPRT and its Application to Cue Enhancement

5.3 Calculating \( P_0 \{ c_m(A) \}, P_1 \{ c_m(A) \} \)

5.4 Optimizing the Raw Cue.

5.5 Conclusions

6 Grouping Experimentation

6.1 A Heuristic Algorithm for the MLGC Criterion

6.2 Example 1: Grouping of Points Into Co-Linear Sets

6.3 Example 2: Grouping of Edgels Lying on Smooth Curves

6.3.1 Creating a Saliency Map

6.4 Example 3: Motion-Based Segmentation

7 Grouping-Based Hypothesis Verification

7.1 Introduction

7.2 Grouping Cues and Their Representation

7.3 The Grouping Based non-Additive Score

7.3.1 The Hypothesis-Induced Partition of The Data

7.4 The Likelihood of an Hypothesis

7.5 A More Compact Form of The Log-likelihood Score

7.6 Some Properties of The Log-Likelihood Score

7.7 The Possible Failure Modes and Their Relation to The \( \epsilon_{bg} \) Parameter

7.8 Experiments

7.9 Conclusions

8 Ground From Figure Discrimination

8.1 Introduction

8.1.1 Related Work

8.2 Grouping Cues and Their Graph Representation

8.3 The Algorithm

8.3.1 A Fast Graph Construction

8.3.2 The Smoothness Grouping Cue

8.3.3 The Iterative Pruning Process and the Bootstrap Mechanism

8.3.4 Complexity and Run Time

8.4 Experimentation

8.5 Conclusions

9 Discussion
Abstract

Visual processes deal with analyzing images and extracting information from them. They play an important role in many of today's industries, e.g., in the medical instrument industry, the electronic and micro-electronic industry, the movie industry, the robotics industry, and in many other fields which require automatic extraction of visual information from images. One of the major difficulties in image analysis is that only a few subsets of the data features contain the useful information, while all others are not relevant for the task. The irrelevant features interfere with the visual process and make it more difficult, slower, and less accurate. Grouping processes, which rearrange the given data by eliminating the irrelevant data features and sorting the rest into groups each corresponding to a certain object, are indispensable in computer vision.

In this thesis we propose a new framework for perceptual grouping in computer vision. The grouping information available to the process is represented by the grouping likelihood graph. We use this model to develop and analyze algorithms for several machine vision tasks. We present a grouping algorithm, a hypothesis verification algorithm, and a figure ground discrimination algorithm, which are based on known statistical tools, such as the Wald's Sequential Probability Ratio Test (SPRT), and the Maximum Likelihood criterion.

The most important contribution of this work is the theoretical analysis of the algorithms — mainly the grouping algorithm. The performance analysis of the grouping algorithm allows us to predict the resulting grouping quality relative to the information available for the task. Such an analysis has never been done for grouping and is rarely found in other computer vision areas.

This framework is fairly general, and does not depend on the domain. Three grouping algorithms, in three different domains, are synthesized as instances of the generic grouping method. They demonstrate the applicability and generality of this grouping method. Moreover, we show that the same perceptual information can be used for other visual tasks, such as figure ground discrimination, and hypothesis verification in object recognition. We believe that this framework could be useful for other tasks and domains in different levels of the visual system.
Chapter 1

Introduction

Perception is not a mere passive recording of information impressed upon my sensory organs by the environment. Rather, it consists of an active construction by means of which sensory data are selected, analyzed, and integrated with properties not directly noticeable but only hypothesized, deduced, or anticipated, according to available information and intellectual capacities (Gaetano Kanizsa, from [SB93]).

Visual processes deal with analyzing images and extracting information from them. They play an important role in many of today’s industries, such as medical instruments, the electronic and micro-electronic industry, the movie industry, the robotics industry, surveillance, and many other fields. All these applications require automatic extraction of visual information out of images.

One of the major difficulties in image analysis is that only a few subsets of the data items contain the useful information, while all others are not relevant for the task. The irrelevant data interferes with the visual process, and makes it more difficult, slower and less accurate. It is the ability to impose organization on sensory data which makes human perception so powerful and so versatile. This ability is called perceptual organization, and requires minimal a priori knowledge about the world. This basic knowledge is described by a set of grouping cues — some laws of organization. Perceptual grouping processes use these grouping cues to rearrange the given data by eliminating the irrelevant data items and sorting the rest into groups, each corresponding to a certain object. Doing so, they reduce the computational time of following visual tasks, and improve their accuracy. The role of perceptual grouping in computer vision is critical to success [WT83, Low85, Jac88, Gri90, SB93, HMS94].

In this thesis we propose a new framework for perceptual grouping in computer vision. The grouping information available to the process is represented by the grouping likelihood graph. The graph nodes represent data features extracted from the image, like edge points, and the arcs represent the grouping cues — functions which evaluate the mutual relation between feature pairs. We use this model to develop and analyze
algorithms for several machine vision tasks. We present a grouping algorithm, a hypothesis verification algorithm, and a figure ground discrimination algorithm, which are based on known statistical tools, such as the Wald's Sequential Probability Ratio Test (SPRT), and the Maximum Likelihood criterion.

A major contribution of this work is the theoretical analysis of the algorithms — mainly of the grouping algorithm. We have analyzed the performance of the grouping criterion, and we are able to predict its resulting grouping quality. Such an analysis has never been done for grouping, and is rarely found in other areas computer vision.

The introduction begins with a brief look at the origin of perceptual organization research, found in the field of Gestalt psychology. It proceeds with the use of perceptual organization in computer vision, in previous and current work. Finally it discusses the specific questions by which this thesis is motivated, and provides a broad description of our work and its contribution.

1.1 Gestalt Psychology

The history of research on perceptual organization, in broad outline, is a search for some underlying principle which would unify the various grouping abilities of human perception (Lowe, [Low85]). The Gestalt school of psychology was founded by Wertheimer [Wer50], Koffka, and Köhler in the early twenties of this century. They demonstrated the importance of perceptual organization to human visual perception by a number of self evident examples. These demonstrations of grouping phenomena are considered to be the major contribution of Gestalt psychology to our current understanding of visual perception [Gor89]. The first categorization of perceptual organization rules, made by Max Wertheimer, is know as Wertheimer’s Laws of Grouping (1912). Figure 1.1 illustrates some of the rules which cause elements to be grouped together [Low85, Gor89]:

- **Proximity** — elements that are close.
- **Similarity** — elements which have similar shape, color, orientation, size, etc.
- **Continuation** — elements that lie on a line or a smooth curve.
- **Closure** — elements, like lines or curves, that form a closed shape.
- **Symmetry** — elements which are placed in symmetric order.
- **Familiarity** — elements which we are used to be seen together.

Perceptual organization rules are usually denoted *grouping cues*, or just *cues*. Grouping cues may be thought as time and space regularities, which can take many
Figure 1.1: Some of Wertheimer’s laws of grouping (redraw of a figure from [Low85]).
different forms at different levels of the vision process [WT83]. The variety of grouping processes, from the simplest and fastest tasks of proximity- and co-linearity-based groupings, and up to the very difficult, knowledge-based grouping processes, is demonstrated by the two extreme examples of Figure 1.2.

A known tenet in Gestalt psychology is that “the whole is greater than the sum of its parts”. Obviously, this does not apply to the number of points on a curve, but rather states that the properties of a curve are not just the sum of the properties of all its points. Thus, a part has properties that depends on the whole in which it is included. This implies that parts are not completely independent, and hence cannot be analyzed separately. This idea, which has great significance for perceptual theory, implies that the perceived form is not determined by the individual features, but rather by the relations between them. This is known as the relational effect, and is well documented, for example, in studies of the perception of brightness. The perceived brightness depends on the relation between the illumination and brightness of the surrounding field to the brightness of the object under observation (Köhler, 1947). One of the implications of this thesis is that relations between data features should be measured and used as additional atomic pieces of information in machine vision systems.

Gestaltists describe perception as a dynamic process that interprets complete forms, rather than interpreting the individual parts which comprise them. Kanizsa suggests that the perception of form or shape can be divided into two stages: the
primary process, which locates features, or regions of spatial regularity, and the secondary process, which helps us in going beyond the given information. This is the perceptual inference of totalization, integration and completion, or inferring what is absent. Evidently, many of today’s machine vision systems include these processes as two separate stages.

The grouping process precedes, therefore, the perception of form. However, a more definite specification of the grouping goal is required. Witkin and Tenenbaum [WT83] and Lowe [Low85] have defined the grouping goal in terms of causality: the goal of grouping is to uncover causal relationships between image features. It is to partition the data features into groups, such that all the features within a group are associated with the same cause or the same object in the scene.

This definition was motivated by the following argument. It is clear that detailed analytic models of surface photometry and the imaging process are not essential for surface perception. This is evident from the human capability to derive structure directly from collections of two dimensional image features which are not a result of any physical imaging process, such as range images, line drawings, or electron micrographs [WT83]. This is also the case in random dot stereograms [Mar82], where grouping is explained by the coherence, or regularity, of disparity. These grouping tasks can be formed without any high-level knowledge of of the content of the scene, and it is remarkable that even after the scene has been recognized and understood, the same grouping are nearly always present in the final description. This phenomenon can be explained by the principle of causality — if the grouping task is to find groups of features having a common cause then such groups are unlikely to be separated later in the vision process [Low85].

Humans can perceive the existence of perceptual structures in a scene, and can extract the image elements associated with such structures, even before they recognize them as a meaningful object [Wer50, Low85, Gor89, Low85]. Lowe said that “perceptual organization refers to a basic capability of the human visual system to derive relevant groupings and structures from an image without prior knowledge of its content”. The quality of a shape highly effects the human ability to find it. This goodness of forms is called Pragnanz . There were many attempts to quantify the Pragnanz of scenes, but this was found to be very difficult, because both cues and scenes are extremely general and difficult to be model.

The quantification of Pragnanz is one of the fundamental questions raised by Gestaltists. Many Gestaltists were unsatisfied with the circular definition of Pragnanz , and by the 1950’s there was some agreement on a general principle of simplicity, also known as the “minimum principle”, or the “minimum description length” (MDL) [Low85]. This was stated as the principle that certain forms, which require the least amount of information to specify, could evoke an equal perceptual response to a stimulus (Hochberg, 1957). This idea, which is widely used in information theory
[Cov91], is not very suitable for perceptual organization, because it depends on the description language, which cannot be derived from information theory. Another major problem of the simplicity argument, stated by Lowe, is that it is assumed that the description language will perfectly encode the image. In realistic scenes, however, any visual pattern is likely to only approximate whatever ideal description is being considered. The simplicity criterion does not tell us how to solve the tradeoff between the strength of the shape and the degree of approximation. This problem can be ignored in idealized experiments, but is of crucial importance for computer vision, and for most real images. As will be shown later, there is a more recent approach, in which another property, denoted non-accidentalness, is measured. Every non-casual organization of data features in the image is most likely to have a common derivation. Without a common cause, this organization has a very little chance of being seen. If grouping by causality is the task, then simplicity and non-accidentalness are two possible measures of evidence for the existence of a common cause.

Another implication of the causality principle is that there is a unique cause, or explanation, for the appearance of a feature in the scene. Practically, this makes grouping a partitioning problem, where every data feature should be grouped only once, with other data features of the same object, or cause. This is a very common definition of grouping in computer vision (e.g., [IJ91, ZMF95, SH97, Mat77, DR92]), which we adopted in this work.

This section of the introduction describes only a small part of Gestalt theory — that which is relevant to our research in machine vision. A lot of work has been done by Gestaltists to model the working of the living brain. Unfortunately, these models, which were speculated upon and extrapolated from the simple experiments, do not have now many adherents [Low85, Gor89]. One of the main reasons for their decline is in Köhler’s psycho-neural model, which describes the brain’s working in terms of physics and chemistry, using the “field forces”. Gordon [Gor89] states that Köhler would probably have avoided this trap if knowledge of neurophysiology had been more advanced. However, subsequent research in perception owes a great debt to the movement started by Wertheimer, Koffka and Köhler, who emphasized the importance of organization, found many laws of grouping, the effect of Prägnanz, the relational effect and other phenomenon which have inspired and stimulated research in human and machine vision.

1.2 Perceptual Organization in Machine Vision

Perceptual organization has a very important role in machine vision. In the following sections we first provide the motivation, and then give some perspective on the wide range of applications.
1.2.1 Motivation

The role of perceptual organization in human vision was known long before it was first introduced in computer vision. Until 1983 there were very few works on grouping in computer vision (e.g. [Zah71, Win75, Mat77, SH79]).

Marr was one of the pioneers in using perceptual grouping with computer vision. One of Marr’s acknowledged contributions is his attempt to clarify our thinking about information-processing systems [Mar82]. His theory is that vision is organized as an information-processing system that works in successive stages; in the first stage the image is captured as a spatial distribution of the gray levels on the retina. In the second stage, certain $2D$ geometric primitives, or tokens, are derived from the image. Next, the tokens are grouped together, hierarchically, into $2\frac{1}{2}D$ structures of a higher level, which are then used for perceiving $3D$ shapes and objects. He called this token-based information representation the *primal sketch*, and proposed it as a computational model for vision. It is interesting to note that, 14 years later, Jaynes et al. [JSS+96] built an efficient model-based system for building reconstruction from aerial images, which follows Marr’s ideas in a surprisingly accurate fashion, both in the token grouping and in their gradual dimensionality from $2D$ to $3D$.

Lowe has studied the role of perceptual organization in computer vision [Low85]. Based on the observed behavior of the human vision system, he concluded that perceptual grouping operations must be viewpoint and scale invariant (i.e., stable over a wide range of viewpoints) and have a figure from ground discrimination ability (i.e., to distinguish a common cause organization from an incidental one). He suggested ways to evaluate some of the above mentioned grouping cues, which comply with these basic criteria. He further observed that uncertainty in the image makes these cue imperfect, and concluded that the quantitative goal of perceptual organization is to calculate the probability that an image relation is due to actual structure in the scene. He ranks perceptual groups according to their significance, which is inversely proportional to the *a priori* probability of the accidental occurrence of this event. Perceptual organization plays a major role in his SCERPO object recognition system [Low87].

Witkin and Tenenbaum [WT83] recognized and emphasized the importance and the usefulness of perceptual organization in computer vision. They claim that grouping processes should be a part of many tasks and processing levels in computer vision, much like the situation in the human vision system. They adopt some of the Gestalt psychology principles, and argue that perceptual organization, which can be evaluated by spatiotemporal coherency and regularity, is usually a result of a common cause, and is very unlikely to happen by coincidence. It is therefore a strong evidence for the presence of an object. Spatiotemporal coherency may be measured with different properties, forms and scales, resulting in groups of different dimensionalities, types and sizes.
Zucker has argued that perceptual grouping should be used for segmentation and for finding the properties of the grouped objects [Zuc83]. He studied the problem of contour inference from an image of edge points, and provided several practical algorithms for the task [Zuc86, PZ89, DZ90].

There is evidently much more interest in perceptual grouping since 1983 than there was before (e.g., see the review in [SB93]). This is probably due to a combination of system needs and of the great progress achieved during a short period by Marr, Lowe, Witkin and Tenenbaum, Zucker and others. Most of the work on grouping in computer vision has been done to support object recognition systems. Object recognition is one of the most important fields in computer vision. The efficacy of grouping in model-based object recognition has been shown by Lowe [Low85], Jacobs [Jac88], Grimson [Gri90], Medioni [HMS94], Zisserman [ZMF+95] and others. There are several reasons why grouping is important for the success of a recognition system. To explain them, we should first introduce the basic stages of most model based object recognition systems.

A model-based object recognition system is usually composed of four major successive stages (see Figure 1.3): first, the image is processed, and data features, such as edge points, corners, lines and curves, are extracted. Then a matching process is used to find hypotheses - potential matches between subsets of data features and library models. Each hypothesis includes the information about the identity of the observed object and its pose. In the verification stage, each hypothesis is verified against the image, and a fitness score is evaluated. The scored hypotheses are later used by a reasoning process to make the final decision, which depends on the task (e.g., it may accept the highest scored hypothesis and reject the rest).

Grouping is used in object recognition systems to reduce their computational complexity [Gri90, HMS94, MS96]. Usually, indexing is the most expensive stage in these systems. The high complexity is a result of the many existing potential matches. The minimal number of matched features, that is, the number of matched pairs of data and model features, has to be large enough (e.g., 3-5 points) in order to uniquely determine the corresponding transformation, or pose, between the model and the image. Finding a good (i.e., a correct) match is therefore a combinatorial problem.
Grimson has shown in [Gri90] that the interpretation tree indexing method is very efficient when all the data features are known to be derived from a single object, but dramatically slows down in the presence of clutter, or spurious data features. Perceptual grouping processes, which remove the irrelevant data features and sort the rest into groups, each associated with one object in the scene, help to focus the search for a good match. In the interpretation tree, for example, the matching complexity is polynomial (quadratic) if all data features belong to the same object, but it becomes exponential in the presence of clutter [Gri90]. Grouping dramatically reduces the number of wrong hypotheses that need to be evaluated, and increases recognition performance.

Grouping is used, in a very similar manner, to reduce the complexity of stereo matching algorithms, where a similar combinatorial matching problem is addressed [CWH96, MN89, HM95, Vos92, UE96]. It is worth mentioning here a work by Marr and Poggio, in which they developed a computational theory of human stereopsis [MP79]. Stereopsis is the ability to match up the two slightly different views of a scene captured by the two eyes, resulting in depth perception. In this work they showed, using random dot stereograms, that disparity is a necessary and sufficient cue for stereopsis. It was further shown that both grouping and matching can be explained by the unique cause, regularity and simplicity principles.

In addition to the grouping (clustering) task, perceptual organization is a very powerful source of information for other computer vision tasks. As a part of this thesis, perceptual organization is shown to be very useful for hypothesis verification in object recognition [AL96a]. It is shown that in the presence of high clutter and occlusions, traditional methods, which do not use perceptual organization, often fail to find the correct hypothesis. However, the correct hypotheses can be found by incorporating this additional information.

*Figure-ground discrimination* is another visual task related to grouping. It is the separation, or segmentation, of the object from the rest of the scene. Human perception of the viewed image is not of sensations of light, but of objects and surfaces. There is a tendency to distinguish between the *figure* and its *background*. Humans perceive the figure as a complete and coherent shape that appears in front of the background, which in itself is seen as less distinct, is attended to less readily, and is often seen as floating behind the figure [Gor89]. In machine vision, however, there is a major difficulty in evaluating these quantities. The definition of a figure can, therefore, take many forms, depending on the visual task.

One definition of objects is based on saliency measures. The structured groups tend to be more salient, and to attract attention. *Saliency* may be defined as a measure of local organization. A function that specifies the saliency at each point in the image is called a *saliency map*. Saliency can be defined in many different ways: salient points are points of irregularity, such as corners in a gray level image.
Figure 1.4: A synthetic dots image (left) and its saliency map (right). The bright regions, which are the more salient ones, correspond to the structure found in the dots image.

Smooth closed curves, however, are usually considered as salient curves. Regions which have a unique motion, or a unique color, are salient regions, and so on. Saliency maps are often used in computer vision for figure ground discrimination, grouping, segmentation, to guide the attention of other visual processes to specific parts of the scene, etc.

An extensive review of the perceptual organization research in computer vision has been done by Sarkar and Boyer [SB93]. They emphasize, as a result of their review, that the role of perceptual organization (in the true Gestaltic sense) has been very minor if not altogether absent in computer vision research. They further explain that although work has been done, its full potential has not been realized. The next section provides an overview of different aspects of the field, including some of the work that has been done since their review. The wide range of new applications which involve perceptual grouping emphasize the correctness of their claim, and support the thesis of Witkin and Tenenbaum, proposed ten years earlier.

1.2.2 A Perspective Overview

Vision tasks are commonly classified into low-level, mid-level and high-level tasks, according to their successive place in the vision system, the data types of their input and output, and the size of the neighborhood they consider [WT83]. Low-level vision, also known as early vision, includes tasks that accept images as input, and extract primitive data features, such as edge points, corners, and other points of interest. Mid-level vision processes receive images and/or data features as their input, and produce simple structures, like lines, curves and regions, and geometric mappings between images, based on spatial connectivity and local similarity. At this level, we count tasks such as optical flow, stereo matching and other image to image correspondence
methods, which find the relations within pairs or sequences of images. In high level vision the main goals are to discover structural relations that are not obvious without specialized (i.e., semantic) models, and to use such models to attribute meaning to previously discovered structures. Here we find tasks like shape recovery, object recognition, face recognition, image classification and retrieval (in image databases), and more. Note that in [SB93, WT83] a finer classification is proposed, in a similar manner, with more than three levels. It is not our aim here to classify vision tasks, nor to define the “correct” classes. Our purpose is to adopt this natural consecutive order to sort and discuss the rich body of prior work on perceptual organization.

1. Low-level vision

Perceptual organization is widely used in low-level vision. Applications include the evaluation of saliency maps, later used for figure-ground discrimination [SU88, GS91, HH93] and for grouping [SU90]. Several perceptual grouping methods have been suggested for the grouping of edge points into lines and curves [PZ89, SU90, DR92, MA96], for the grouping of points into clusters [Ahu82, TJA92], for curve partitioning into arc segments [CRH92], and for image segmentation [WI93]. Other grouping methods were developed for gray level and depth-based image segmentation [LCJ91], and for texture-based image segmentation [GGGD90]. As these are low level tasks, methods are fairly general and do not require restrictive constraints on the image domain.

The saliency network method of Shashua and Ullman is well-known in the grouping literature [SU88, SU90]. It evaluates a saliency map of edges, and uses it for the grouping of edge points into smooth curves. The saliency of an image point depends on the length, continuation and smoothness of the best edge curve passing through that point. This method fills small gaps, and has a high preference for closed contours. It has been used for several applications (e.g., to find networks of thin lines in medical and in aerial images [MA96]). Alter and Basri [AB95] have analyzed this method, and point out its advantages and weaknesses. Roughly speaking, the network is capable of efficiently finding the most salient curve, but is not suitable for the detection of the rest of the salient curves, and it has a geometric convergence for cycles.

Cox and Zhong have developed a grouping algorithm to find a set of edge points which form a closed contour, that optimize the ratio between the exterior boundary cost and the enclosed interior benefit [CZ96]. This algorithm is very useful when the boundary of a single object has to be found. Elder and Zucker have proposed in [EZ96] an algorithm for finding closed boundaries in an edge image. Their algorithm also looks for smooth curves, but without considering the area of the enclosed shape.
Guy and Medioni have developed a directional voting method [GM92]. It may be considered as a variation of the Hough transform, where vectors replace the usual scalar votes, and the voting accumulation method, performed in the image plane, evaluates the second moments of the vector contributions in each cell. The result was used to produce saliency maps of smooth edge lines and junctions.

2. **Mid-level vision**

In mid-level vision tasks we count the completion of occluded boundaries [Ull76, WJ96, WJ95, WJ96], the grouping of lines and curves into groups [MN89, DR92, IO92], and motion-based segmentation [AW94, JC94, Sha94, AL97b]. In these tasks, the extent of the sought for structure is usually much wider than in low-level. Although some of the saliency maps can bridge small gaps along smooth lines (e.g., [SU88, GM92]), they cannot handle large gaps, which require an explicit representation of endpoints, and more global methods for the computation of an optimal curve between these endpoint pairs. In these tasks, however, there is a greater need for prior knowledge, or assumptions, about the domain. A common assumption in many motion-based segmentation methods, for example, is that the scene includes only rigid bodies. This limits the scope of these methods and makes them domain dependent.

Parallel lines and curves provide strong evidence to the existence of a structure. This is used for the grouping of parallel lines for object recognition [HMS94, Low87], for stereo matching [MN89, UE96], and for stereo matching plus occlusion compensation [BR94].

Brautigam, Garding and Eklundh proposed a segmentation algorithm for planar surfaces using co-planar corners and stereo disparity [BGE96]. They evaluate the local orientation of planar corners, look for clusters of equal orientations, and then verify the hypothesized plane by stereo matching. This algorithm is limited to the identification of large planes, covered by an appropriate texture.

Leclerc has developed a region grouping method which uses polygons to describe the boundaries and polynomials to describe the region gray level slow variation [Lec90]. A similar approach was suggested by Jepson and Black in [JB96].

In the last few years there is a grown interest in analyzing image sequences, and motion becomes an important grouping cue. There are algorithms for the grouping of points having common rigid motion parameters [JC94, AD93, Sha94], and for motion-based image segmentation [AW94]. Cox have reviewed the statistical methods used to evaluate motion correspondence in image sequences [Cox93].

Finding illusory contours is also considered as mid-level vision task. There are some algorithms for subjective contours in machine vision, which are based on
energy minimization [Ull76, Mar82, BN90, WJ96] on distance maps [ASZ96],
directional filtering [HvdH93], and on symmetry [ZMF+95]). However, it is hard
to judge the results, as the produced contours are indeed subjective, and are
often different from humans subjective judgment.

3. High-level vision In the last few years, perceptual grouping is used also for
high-level vision tasks, such as model-based object recognition [DSB93] class-
based object recognition [HMS94, ZMF+95, JSS+96], structure-based stereo
matching [CWH96, MN89, HM95, Vos92, UE96], image representation and
classification [FMF+96], model-based 3D shape reconstruction and image fu-
sion [JSS+96], and hypothesis verification [Bre93, AL96a]. In particular, it was
shown that perceptual grouping is useful for more than segmentation, or clus-
tering.

Modayur and Shapiro use grouping of feature points into pairs to accelerate
the indexing stage in object recognition [MS96]. They show how the number of
hypotheses dramatically decreased by sorting the hypotheses according to their
likelihoods, and evaluating them in that order.

Perceptual grouping is used in class-based object recognition systems, to find
sets of symmetric curves [ZMF+95, HMS94]. These recognition systems are
capable of recognizing 3D objects of specific classes, like polyhedra, straight
homogeneous generalized cylinders (SHGC) (e.g., surface of revolution), and
canal surfaces (the envelope of a sphere swept along a curvy axis). Each of
these classes is characterized by different symmetry rules that are used in the
grouping process to locate potential subsets of curves. These subsets are then
being processed, and the symmetry is also used to complete occluded parts.
This differs from the classical model based recognition, where the exact shape
of the model is known.

Another high level grouping is the grouping of edge segments lying around closed
regions [Cle91], around convex shapes [Jac96a], or grouped by other constraints
[IO92]. The grouping of curves that coincide with the same generalized cylinder
model [HM95, ZMF+95], and the grouping of arc segments [Sau92] may also fall
into this category.

Du, Sullivan and Baker have used grouping for car tracking [DSB93]. They
initiate a small group of line segments, and then iterate this group while com-
paring it with the car model, until the best match is found. Their algorithm is
similar to Lowe’s system (SCERPO, [Low87]).

High-level vision processes usually involve restrictive assumptions about the ob-
jects in the scene (e.g., [DSB93, JSS+96, ZMF+95], the illumination conditions
(e.g., [JSS+96]), etc. As more a priori knowledge is required, these methods
can work only in specific domains. This natural evolution supports the thesis of Witkin and Tenenbaum, who argue that perceptual organization should be a part of many processes in the computer vision system. The initial work in these new interesting and promising directions should be followed by more intensive research.

A different perspective of perceptual organization methods is gained by sorting them hierarchically, according to the description level of image organization. The first layer includes methods which find the amount of local organization at every point in the image, without explicitly finding the structure, as in the evaluation of saliency maps. Saliency can be measured and described by a saliency map in the image plane. It is defined as a measure of local organization. Better organization is less likely to be found by coincidence, and is therefore more salient. It is commonly used for filling in the gaps of an occluded boundary (subjective contours) [Ull76, BN90, HvdH93], for noise reduction [SU88, GM94] and more. The second layer includes processes that find local structural features, such as figure ground discrimination methods [HH92, HH93]. Such processes look for the presence of local organization, and decide for each data feature whether to put it in the figure set or in the background set. Grouping processes belong to the next layer, where the figure should not only be explicitly separated from the background, but should also be divided into groups, each associated with a single object. The next layer is object recognition. This high description level, in which models and transformations are used to describe the image organization, imposes a partitioning which is very close to the common cause principle.

Perceptual grouping processes may also be classified by the dimensionality of the domain, or of the sought-for groups. Zucker classified them into one dimensional (1D) and two dimensional (2D) groupings [Zuc83]. The different nature of 1D and 2D structures and groups usually requires different grouping cues and computational methods. In [SB93] they have proposed a general classifactory structure for perceptual organization in computer vision, which may also be thought of as a combination of two of the above categorizations. They classify the different grouping methods with respect to the types of features being organized and the dimensions over which the organizations are sought. They define four types of features, namely the Signal level (e.g. dots, interest points), the Primitive level (e.g. regions, edge chains), the Structural level (e.g. edge and region primitives), and Assembly level (e.g. corners, polygons, closed regions). They also noticed four classes of dimensionality, which are 2D and 3D, with or without the additional time dimension. However, as is later explained, we conceive the dimensionality of the data and group as a property that mainly reflects on the grouping cues and their extent, and might have only a minor impact on the grouping method.
1.2.3 Grouping Cues

We distinguish between two components of the grouping method: the grouping cues that are used and the grouping mechanism which combines these cues into a partition of the data set, according to a grouping criterion. Perceptual grouping cues are the atoms of grouping evidence. They are the building blocks of all perceptual organization processes, and shall be treated as the only source of information available for this task. These are the organization rules used by the vision system. In general, cues are domain-specific functions which rely on the assumed organization and on the properties of the sought-for groups. Their choice is essentially made by taste and intuition, although more rigorous statistical properties are sometimes taken into account [Low85, Jac88, Cle91, CRH93, GGGD90].

Some of the most useful cues to computer vision have been imported from the Gestalt psychology (see Section 1.1.1). These include all Wertheimer’s Laws of Grouping: proximity and smoothness (also continuation, co-linearity and co-circularity) are often used for the evaluation of saliency maps [SU88, GS91, GM92], for figure-ground discrimination [HH93], for the grouping of edge points lying on smooth curves [PZ89, SU90, MA96, AL97b], for subjective contours [HvdH93], and for hypothesis verification [AL96a]. Symmetry is used for stereo matching [MN89], for class-based grouping [ZMF+95], for model-based grouping [HMS94], and for attention [RWY95]. Note that in [ZMF+95] they used different types of symmetry, not only for the grouping process, but also for the completion of occluded parts in the detected groups. Closure is used for grouping of edges [Cle91], and for evaluating subjective contours [ASZ96]. Similarity is used for grouping and segmentation in [LM96]. Stereo disparity is used for the grouping of points lying on virtual 3D smooth surfaces in dot stereograms [MP79]. Common motion is often used for segmentation and grouping [JC94, AW94, AL97b]. Parallelism is used for the grouping of lines in 2D [Low85, MN89], and co-planarity is used for the grouping of surfaces in 3D [JSS+96, BGE96].

Jacobs has developed an efficient algorithm for the grouping of lines into convex groups [Jac88, Jac96a]. Convexity is an important grouping cue to the human visual system. In [LJB96], Liu, Jacobs and Basri have shown that grouping the two parts of a shape, partially occluded by a wide horizontal strip, is stronger when the shape is convex. This was found by providing a competitive grouping cue - a stereoscopic depth perception. It was shown that convex shapes are grouped over larger depth differences than non convex shapes, thus implying the importance of convexity.

While the Gestaltists successfully identify the qualitative organization rules that work surprisingly well for the human vision system, one still has to find applicable computational methods to evaluate them, in order to plug them into a machine vision system. This question has no definite answer, even in probabilistic terms, because we do not have a model for the real world, and therefore we cannot tell the probability
of an accidental occurrence. Almost every discriminating function may work, but the overall system performance may be highly dependent on the choice of function (as we shall see later, in Section 5.4). An important observation is that there is no consensus in the computer vision literature about what should be measured for a given grouping task, and how it should be evaluated. The well-studied task, of grouping edge points lying on a smooth boundary, is a good example for the variety of perceptual grouping cues; the typically used cues are proximity and co-linearity [WB86], co-circularity [Sau92], curvature and length [SU90, PZ89], and some combinations of these [DR92, HH93, GM92]. The same problem occurs in other domains as well, and is one of the reasons that makes it so difficult to compare the performance of different grouping systems.

1.2.4 The Non-Accidentalness Criterion

Grouping cues are functions which evaluate the amount of organization in the local structure, and use it to discriminate between causal to casual occurrences. The measure of simplicity relates to the a priori expectation that simple descriptions are more likely than complex ones [Low85]. The description implies a single casual explanation, common to all features in the perceived structure, or organization. Let $Pr\{\text{Organization}\}$ denote the a priori probability of a structure to be seen, then $\frac{1}{Pr\{\text{Organization}\}}$ is a measure of complexity (this value is related to the description length. For example, consider the Huffman code [Huf52, GW87] as a description language for a finite set of observable structures).

In more recent work, Prägnanz is measured as the degree of non-accidentalness, also denoted as surprising simplicity. This explanation for the preference of one perceived form over another was independently proposed by Witkin and Tenenbaum [WT83], Lowe [Low85], and Rock [Roc83]. Witkin and Tenenbaum state that the strength of good grouping methods does not derive from their a priori expectations, but from a non-accidentalness measure. They suggested the use of regularity and coherence in some feature spaces as a general form for grouping cues. Lowe gives, as an example, the strength of parallelism as a grouping cue: if two parallel curves are considered to be highly significant, it is not due to the fact that structures which evoke parallel curves are more common than structures which do not, but rather to the very small probability that two independent curves would happen to be parallel by accident. Of course, it is also true that any other exactly specified relation between two curves would be very unlikely to happen by accident, so the non-accidentalness argument still relies on a priori expectations. However, the most important consideration is not the expectation for parallels versus non parallels, but the expectation for parallels arising by accident from some expected distribution of the constituent features.
Sarkar and Boyer [SB93] have suggested an explicit statistical measure to the importance of organization. They express the conditional probability of causality given an observed organization as

\[
Pr\{Causality|Organization\} = \frac{Pr\{Organization|Causality\}Pr\{Causality\}}{Pr\{Organization\}}
\]

where \(Pr\{Organization|Causality\}\) is the probability that we will observe this organization among a set of features given that they are coming from the same cause, and \(Pr\{Causality\}\) is the a priori probability that a set of features comes from a common cause. However, \(Pr\{Causality\}\) and \(Pr\{Organization\}\) are usually unknown, as they might rapidly change between scenes.

In this work, a different non-accidentalness measure is used. It is assumed that there are two different populations of subsets for which the cue is applied: consistent and inconsistent ones. A set is said to be consistent if all its features have a common cause (hence consistency is equivalent to causality). It is assumed that the two different conditional probabilities are known: \(Pr\{Organization|Consistency\}\) and \(Pr\{Organization|Inconsistency\}\). Note that, given only these two conditional probabilities, one cannot conclude about \(Pr\{Organization\}\) without knowing the probability of \(Pr\{Consistency\}\). However, the likelihood ratio,

\[
\frac{Pr\{Organization|Consistency\}}{Pr\{Organization|Inconsistency\}}
\]

can be used as a measure of non-accidentalness, or Pragnanz. The likelihood ratio is a known discrimination measure, often used in hypothesis testing (see, e.g., [Fuk90]).

Using this definition, the a priori probabilities, \(Pr\{Causality\}\), \(Pr\{Organization\}\) are not required. The proposed model is described in the body of this work.

### 1.2.5 Grouping Mechanisms

The grouping criterion is usually defined as the maximization of some consistency function between the group assignments and the given data. This maximization is done by various grouping mechanisms, including dynamic programming [SU90, MA96], relaxation labeling [PZ89, TJA92], simulated annealing [HH93], constrained simulated annealing [GGG90], and different graph clustering techniques (we devote Section 4.2.1 for these techniques).

Grouping may also be carried hierarchically [HMS94, MN89, DR92], or in a split-and-merge approach [OPR78]. The grouping criterion proposed here is also a maximization of a function: the likelihood of the available data, relative to the grouping decision. The crucial difference which exists between the proposed method and previous work is in the analysis we provide, which predict the performance of the proposed grouping criterion, based on the reliability of the data.
1.3 The Contribution of This Work

In this thesis we propose a new framework for perceptual grouping in computer vision. The grouping information available to the process is represented by the grouping likelihood graph. The graph nodes represent data features extracted from the image, like edge points, and the arcs represent the grouping cues — functions which evaluate the mutual relation between feature pairs. We use this model to develop and analyze algorithms for several machine vision tasks: a generic grouping algorithm, a hypothesis verification algorithm, and a figure ground discrimination algorithm, which are based on known statistical tools, such as the Wald’s Sequential Probability Ratio Test (SPRT), and the Maximum Likelihood criterion.

In our opinion, the most important contribution of this work is the theoretical analysis of the algorithms — mainly of the grouping algorithm. Although many grouping methods have been suggested and tested, it seems that no solid theoretical background has been established. The performance of grouping algorithms in early work, was assessed by implementing the algorithm, testing it on a small number of simulated or real examples, and then visually evaluating the results. This methodology shows that some of the grouping methods perform well on the examples tested and indeed succeed in partitioning the image elements into seemingly correct subsets. It does not allow us, however, to predict the performance of these algorithms on other images or to compare algorithms which have not been tested the same way. In this thesis, we analyze the power of the maximum likelihood criterion to provide guaranteed grouping quality. This analysis quantifies the expected grouping performance as a function of the quality of the available data, the reliability of the cue, and the connectivity of the graph. Similar analyses exist for object recognition (e.g. [GH91a, Lin94, SR89]). The proposed grouping criterion is, in general, computationally hard to solve. Therefore, we also provide a heuristic practical algorithm, which was successfully used in our experiments.

The proposed grouping algorithm is fairly general. It is applicable to several domains, and we implemented and tested it in three of them. Most, if not all, previous algorithms were domain specific. The analysis is domain independent, and thus applies for all the specific cases.

Although good cues are essential for successful grouping, finding them is not our aim here. Instead we assume that the cue function is given, model it by a random variable, quantify its reliability, and focus on the relation between this reliability and the expected grouping quality. In many situations this reliability is predetermined and the grouping algorithm designer can only choose the more reliable cues from the available variety.

Here we show how the reliability of a grouping cue can be significantly improved by using statistical evidence accumulation techniques. We propose a general cue en-
The contribution of this work involves a **hancement procedure** (CEP), incorporating **multi-feature cues** (operating on more than two data elements and relying on higher order statistics), which provides very reliable bi-feature cues. This procedure takes a random sample of multi-feature cues, sharing a pair of data features, and uses the Sequential Probability Ratio Test (SPRT) to reach a decision about this pair. This procedure implies a tradeoff between the computational effort and the achieved reliability of the enhanced cue. This dependency is explicitly calculated and is used to optimize the expected running time, for any desired cue reliability. This method is not restricted only to improving our grouping algorithm, and can be used to improve other grouping algorithms as well.

Grouping is mainly used in computer vision to reduce computational effort of other visual algorithms. However, most grouping algorithms are rather complicated, and require high computational effort. The figure ground discrimination process, which splits the data features into only two sets: the **figure** set and the **background** set, provides an alternative to grouping algorithms. As the task involved is simpler, it is likely to have simpler and more efficient algorithms than grouping. We propose a new, efficient, “Ground From Figure Discrimination” algorithm. Our method can be classified as an asymmetric relaxation labeling process. At every stage, the data features are classified into either “background” or “still unknown” classes, thus emphasizing the background detection task, as implied in its name. A bootstrap mechanism improves performance in very cluttered scenes. A fast implementation using a kd-tree allows work on large, realistic images.

Witkin and Tenenbaum have already argued that, like the human vision system, many processing levels of a computer vision system may benefit from grouping [WT83]. Here we follow this argument and use grouping also for hypothesis verification in object recognition, which is certainly a high level process. We proposed a new verification paradigm, which does not rely on the size of the consistent features set. It rather relies on grouping information, which is extracted from the image independently of the object hypothesizing mechanism. The method is controlled by one parameter, which should be selected based on the expected error mechanism. We proved that this method is consistent, in the statistical sense; that is, in the presence of only one object in the scene, the correct hypothesis is expected to get the highest score. Our experimentation shows that while additive approaches (based on the consistent set size) for verification work well for easy tasks, our approach works significantly better in many difficult cases.

All the above mentioned algorithms use the same grouping information, available to the process by the **grouping likelihood graph**. We believe that this framework could be useful for other visual tasks, as an important additional source of information. As this is an abstract, domain independent model, it could be useful for many different applications. Computer vision algorithms should not be fed by data features only, but also with grouping cues, which evaluate the relations between data features.
Chapter 2

The Grouping Task and its Graph Representation

2.1 The Grouping Task

We consider the grouping task as a set partitioning problem. Let \( S = \{v_1, v_2, \ldots, v_N\} \) be the set of data elements, which may consist of the boundary points in an image, previously detected line segments, etc. \( S \) is naturally divided into several \emph{groups} (disjoint subsets) so that all data elements in the same group belong to the same object, lie on the same straight line, or are associated with each other in some other manner. Let

\[
S = S_0 \cup S_1 \cup S_2 \cup \ldots \cup S_M
\]

denote the natural true partition, where \( S_1, \ldots, S_M \) are groups in the sense described above, and \( S_0 \) is the set of non-important background elements. In the context of the grouping task, the data set is given but its partition is unknown and should be inferred from indirect information given in the form of grouping cues.

Many researchers define grouping as a partitioning problem (e.g., [LCJ91, ZMF+95, SH79, Mat77, DR92]). This definition coincides with the \emph{Causality} principle, i.e., there is a unique cause, or explanation, for the appearance of each feature in the scene. Hence the grouping process is an attempt to identify subsets of data features which have a common cause. We should also mention, however, that according to another grouping concept, the hypothesized groups are not necessarily disjoint. In some cases it is claimed that a data feature can be associated with more than one object (e.g., the edge pixel at the center of a T junction). In other cases the reason is to postpone decisions - the grouping process provides a few alternatives and a later process chooses between them [Jac96a, HMS94, Sau92]. We do not discuss this possibility, but we believe that at least some of the tools developed here are useful for analyzing it too.
CHAPTER 2. THE GROUPING TASK AND ITS GRAPH REPRESENTATION

2.2 Grouping Cues

Grouping cues are the building blocks of the grouping process. They may be described as functions operating on a subset of data elements, and revealing some information about the grouping of this subset. More formally, let \( A \subset S \) denote a subset of \( m \) data elements \((m \geq 2)\). We shall say that \( A \) is consistent if it is a subset of some single true group, and denote it by \( t(A) = 1 \). Similarly, we say that \( A \) is inconsistent if \( A \) includes data features from more than one true group, and denote it by \( t(A) = 0 \).

Unfortunately, we cannot measure the consistency of a subset, \( t(A) \), directly from the image. A grouping cue is a scalar function \( C_m(A) \), which tells us something about the consistency of the subset \( A \).

**Example 2.2.1** Consider the problem of grouping points which lie approximately on a straight line (e.g., see Figure 2.2.1). Let \( A = \{(x_1, y_1), \ldots, (x_m, y_m)\} \) be a small subset of \( m \) points \((m > 2)\).

We use the second eigenvalue of the 2nd-order central-moments matrix of \( A \) as a colinearity grouping cue (see, e.g. [GM92]). Let

\[
m_{01} = \frac{1}{m} \sum_{A} x_i \quad m_{10} = \frac{1}{m} \sum_{A} y_i
\]
denote the first order moments. Let

\[
cm_{20} = \frac{1}{m} \sum_{A} (x_i - m_{10})^2
\]

\[
cm_{02} = \frac{1}{m} \sum_{A} (y_k - m_{01})^2,
\]

\[
cm_{11} = \frac{1}{m} \sum_{A} (x_i - m_{10})(y_i - m_{01})
\]

(2.1)
denote the second order central moments. Let \( d = \sqrt{(cm_{20} - cm_{02})^2 + 4 cm_{11}} \), then

\[
\lambda_1 = 0.5(cm_{20} + cm_{02} + d)
\]

\[
\lambda_2 = 0.5(cm_{20} + cm_{02} - d)
\]

(2.2)
denote, respectively, the first and the second (smaller) eigenvalues of the matrix \[
\begin{bmatrix}
  c_{m11} & c_{m20} \\
  c_{m02} & c_{m11}
\end{bmatrix}.
\]

The co-linearity, multi-feature binary cue is

\[
C_m(A) = \begin{cases} 
1 & \text{if } \lambda_2 < T \\
0 & \text{else.}
\end{cases}
\]

(2.3)

where \( T \) is a threshold which specifies the allowed perturbation of the points from the straight line. If the points in \( A \) are approximately co-linear, then \( \lambda_2 \) is small and \( C_m(A) = 1 \), otherwise \( C_m(A) = 0 \). The selection of \( T \) is considered in Section 5.4.

As apparent from the example, grouping cues are domain-dependent. Good cues should be discriminative, and preferably, should also be pose invariant and robust to noise [Low85]. Many of the grouping cues considered in the literature are functions defined over data subsets including only two data elements. We shall refer to such functions as bi-feature cues. However, more general and often more reliable multi-feature cues may be defined over larger data subsets, including three data features or more. The above co-linearity cue, a convexity cue [Jac96a], and a U-shape cue [MN89], are examples for cues which require at least three data elements (points, line segments). Although bi-feature cues are usually easier to calculate and are more straightforward to use, cues which test larger data subsets have several significant advantages. Many useful cues are simply not defined when only one pair of elements is considered (e.g., line-segment convexity), or become degenerate (every two points are co-linear). Bi-feature cues usually have corresponding multi-feature cues with improved reliability. For example, accidental co-linearity is less likely if more points are considered, while the miss probability should decrease only slightly in this case. More generally, the reliability of the shape-based multi-feature cue of “consistent with some instance of a particular object” clearly increases with the number of data features [GH91a, Lin94]. Multi-feature cues are considered in Chapter 5, where we provide a cue enhancement procedure (CEP), which integrates the evidence available from them into very reliable bi-feature cues. In the rest of this section, we shall consider only bi-feature cues, which may be either the cues used by common grouping processes or the result of the CEP. We shall denote a bi-feature cue by \( c(v_1, v_2) \), or just \( c(e) \), where \( e = (v_1, v_2) \) denotes a pair of data features.

In accordance with one of the main goals of this work, which is to provide a general framework for grouping processes, we would like to determine grouping performance in a domain-independent way. Therefore, we shall not consider the domain dependent properties of cues, and shall characterize them only by their reliability. This reliability is quantified by considering the cue function to be a random variable, the distribution of which depends on the consistency of the tested pair of data features, \( t(e) \in \{0, 1\} \). A given cue function is therefore modeled by two probability density functions. Lowe has used a similar approach to define the measure of non-accidentalness [Low85]. He used
the Bayesian approach to evaluate the probability of causality according to the given measurements, assuming that these measurements could be either the result of an accidental occurrence or a causal reason. His approach, however, requires additional knowledge about the prior probability for the causal occurrence of a grouping cue. This type of information, which can change rapidly between scenes, is not required by our method.

In the context of this work we shall consider only binary cues, which return either 0 or 1. Binary cues can be easily obtained from any other, more general bi-feature cue, by setting a threshold on its result (see, e.g., (2.3)), as illustrated in Figure 2.2. For binary cues, the two pdfs may be simply described by the two corresponding error probabilities: $\epsilon_{miss}$ is the probability that the cue indicates a wrong negative answer ($c(e) = 0$ while in fact $t(e) = 1$), and $\epsilon_{fa}$ is the probability that the cue indicates a false positive answer (a false alarm: $c(e) = 1$ while in fact $t(e) = 0$). If both $\epsilon_{miss} = 0$ and $\epsilon_{fa} = 0$, then $c(e)$ is an ideal cue. It is important to note that $\epsilon_{miss}, \epsilon_{fa}$ describe two different distributions, defined over two disjoint populations (one pdfs defined over the consistent pairs, $Pr\{c(e)|t(e) = 1\}$ and the other over the inconsistent feature pairs, $Pr\{c(e)|t(e) = 0\}$). This characterization can sometimes be calculated using analytical models (e.g., [Low85]), and can always be approximated using Monte-Carlo experimentations (see, e.g., [Jac88] and Figure 6.3(a)).

![Figure 2.2](image_url)

**Figure 2.2:** A binary grouping cue, characterized by its two error probabilities $\epsilon_{miss}, \epsilon_{fa}$, can be obtained by setting a threshold to a more general cue.

From Figure 2.2 it is apparent that this simple threshold approach forces a tradeoff between the two error probabilities, $\epsilon_{miss}$ and $\epsilon_{fa}$. As we shall see in Chapter 5, it is possible to improve (decrease) $\epsilon_{miss}$ and $\epsilon_{fa}$ simultaneously, if we are willing to pay in computational effort.
2.3 Representing Groups and Cues Using Graphs

Our approach to the grouping process is to convert the partition problem into a graph clustering problem. We use two graphs to represent the perceptual information provided by the cue, and another graph to represent the grouping result. A fourth graph is used for the analysis to represent the ground-truth partition (which is unknown to the grouping process). The nodes of all four graphs are the observed data elements, \( V = S \), but the arcs may have different meanings. We shall use arcs as a natural way to describe feature pairs. An arc, \( e = (v_1, v_2) \), represents the pair of nodes (data features) it connects. Let \( E_c(A), A \subseteq V \) denote the set of all possible arcs between nodes in \( A \). Thus a sub-graph defined by \( (A, E_c(A)) \) is a clique (an example for these graphs is given together with the algorithm flowchart, in Figure 3.1).

The ground-truth partition, which is to be determined, is represented by the target graph, \( G_t = (V, E_t) \), composed of several disconnected cliques. Every such clique represents a different object (or group). There is no connection (arcs) between nodes which belong to different cliques. A graph with this characterization is called a clique graph and the class of such graphs is denoted \( \mathcal{G}_c \). The nodes of this graph are available to the grouping algorithm, but its arcs, which contain the grouping information, are hidden and are not directly observable. In fact, the arc set of the target graph is the set of all consistent feature pairs: \( e \in E_t \iff t(e) = 1 \). Knowing that \( G_t \) belongs to the class of clique graphs, \( \mathcal{G}_c \), the grouping algorithm should provide a hypothesis graph, \( G_h = (V, E_h) \in \mathcal{G}_c \), which should be as close as possible to \( G_t \).

Perceptual grouping information is extracted from the image using a (binary) cue function and is represented by two graphs; the underlying graph and the measured graph. The underlying graph, \( G_u = (V, E_u) \), specifies, by its arcs, the feature pairs which should be evaluated by the cue function. The measured graph, \( G_m = (V, E_m) \), specifies the information provided by these cues. That is, an arc belongs to \( G_m \) if and only if it belongs to \( G_u \) and the result of the binary cue function indicates that the feature pair belongs to the same group (ie. \( c(e) = 1 \)).

While the underlying graph is specified by the designer, depending on the domain and on computational effort limitations (see Section 3.1), the measured graph is a result of the cue evaluation process and a part of the grouping process. Consider, for example, a sequence of edgels lying on a smooth curve. All these edgels belong to one true group and thus form a clique in the target graph (each graph node denotes an edgel). As we shall see in the next section, the appropriate underlying graph for this grouping task is a locally connected graph, which contains only a small subset of the arcs of this possibly large clique. Yet, ideally, all the arcs included in the projection of the clique on \( G_u \) should be detected by the cue function, and are expected to be found in \( G_m \). Other arcs in \( G_u \), connecting a node of the group to an alien node, are ideally expected to be eliminated by the cue from \( G_m \). Therefore, we may infer
on the unknown clique graph by observing the underlying graph and the measured graph.
Chapter 3

The Generic Grouping Method

The generic grouping method described in this section consists of two main stages (see Figure 3.1): cue evaluation (for many feature pairs) and the maximum likelihood graph clustering (partitioning) criterion (MLGC). The two stages are general and do not depend on the particular grouping domain, except for the choice of a domain-dependent cue and some associated decisions made before the process. The essence of the grouping method is described in the next chapter, while the implementation details are left to Section 7.8 and to the technical report [AI94a].

3.1 The User Decisions: Cue and Connectivity Selection

In order to use the proposed generic method, several decisions must be made. First, a grouping cue must be chosen, which naturally depends on the domain and on the assumed characterization of the sought-for groups. A related decision corresponds to choosing the feature pairs for which the cue should be evaluated. In principle, all feature pairs, corresponding to a complete underlying graph, $G_u = (V, E_c(V))$, should be considered. However, some cues are only meaningful for near or adjacent data elements and are not adequate for evaluating every feature pair. Therefore, cue evaluation can be restricted to a subset of the feature pairs, specified by the spatial extent of the available cue. We use a locally connected graph for the task of grouping edge points lying on a smooth curve. In this graph, every one of the data features is connected to its $K$ nearest neighbors (see the illustration in Figure 3.1, Figure 6.5, and Figure 6.6). On the other hand, a complete graph is used for grouping of co-linear points, and for motion-based segmentation (see Figure 6.7).

Another consideration which influences the choice of the underlying graph is the reliability of the grouping process and the computational effort invested in it. As we shall see, reliability increases with the density of the graph, but so does the computational effort, so some compromise should be made. Although we provide
CHAPTER 3. THE GENERIC GROUPING METHOD

Figure 3.1: The proposed grouping process: The image is a set of data features (edgels in this illustration), each of which is represented by a node of a graph. The designer should decide about a cue and about the set of feature-pairs to be evaluated using this cue. This set of feature-pairs is specified by the arcs of the underlying graph \( G_u \). The first stage of the algorithm is to build \( G_u \) and then use grouping cues to decide, for every feature pair in \( G_u = (V, E_u) \), if both data features belong to the same group. These decisions are represented by the measured graph \( G_m = (V, E_m) \); every arc corresponds to a positive decision (hence \( E_m \subseteq E_u \)). The known reliability of these decisions is used in the second stage to find a maximum likelihood partitioning of the graph, which is represented by the hypothesized (clique) graph \( G_h \). A main issue considered in this thesis is the relation between this hypothesis, \( G_h \), and the ground truth target graph, \( G_t \), which is unknown.

some quantitative tools for choosing between alternative cues, we do not focus here on optimizing the decisions concerning the cue selection and the underlying graph design. We assume that both the cue and the associated adequate “topology” are either given or chosen intuitively.

3.2 First Stage: Evaluating the Grouping Cues

The image provides a set of data features, which may be edge points, line segments, etc. First, the underlying graph is built, according to the chosen cue and the guidelines described above. Then, all feature pairs corresponding to arcs in \( G_u = (V, E_u) \) are considered, one pair at a time. The cue function is used to decide whether the two data features belong to the same group. We show how to make this decision reliably, based on multi-feature cues (see Chapter 5). Positive decisions are represented by
the measured graph $G_m = (V, E_m)$. After all decisions are made, $E_m$ is an estimate of $E_t \cap E_u$, the projection of the target graph $G_t$ on the underlying graph $G_u$ (the projection of a graph $G_1 = (V, E_1)$ on a graph $G_2 = (V, E_2)$ denotes the graph $G = (V, E_1 \cap E_2)$). The measured graph carries the information accumulated at this stage to the graph clustering stage.

Note that it is also possible to postpone the decisions, mark each arc with a pair of likelihoods, and continue with the same partitioning stage. While the additional information may yield better results, it requires that the actual non-binary cue distributions be given. This is rarely the case, and we do not further consider this issue.

### 3.3 Second Stage: Maximum Likelihood Graph Clustering (MLGC)

Recall now that every decision made in the first stage is modeled as a value of a binary random variable, the statistics of which depends on whether or not the two data features belong to the same group. Therefore, the likelihood of this decision depends on the consistency of the corresponding arc.

We may look now for the grouping hypothesis which best agrees with the cue information. Note that every hypothesized grouping, or graph partitioning, implies a hypothesis on the consistency of every arc in the underlying graph. Two data features are hypothesized to be consistent if and only if they are assigned by the partition to the same part (group). Therefore, the joint likelihood of all the decisions made in the previous stage (represented by the measured graph $G_m$) depends on the hypothesized grouping. Following the common tradition of estimation, we propose to choose a clique graph $G_h$ (a partition), which maximizes the overall likelihood, as an approximation of the required unknown target graph $G_t$.

$$G_h = \arg \max_{G \subseteq G_{\text{c}}} L\{G_m | G\}. \quad (3.1)$$

Note that if $G_u$ is not a complete graph, then we can only expect to find the hypothesis up to its projection on the underlying graph. If two hypotheses, $G_{h_1}, G_{h_2}$ have the same projection on $G_u$, that is $E_{h_1} \cap E_u = E_{h_2} \cap E_u$, they cannot be distinguished by the available cue information, and obviously have the same likelihood.

In the context of this work, the cue decisions are assumed to be independent and are subject to two types of errors specified uniformly by two error probabilities $^1$:

$$\epsilon_{\text{miss}} = P r\{c(e) = 0|t(e) = 1\} \quad \epsilon_{\text{f.a}} = P r\{c(e) = 1|t(e) = 0\}. \quad (3.2)$$

$^1$Making these probabilities non-uniform, and thus associating every arc of $G_u$ with an individual pair of error probabilities, may be a more accurate model, but requires more knowledge about the error mechanism.
These error probabilities can also be expressed in terms of the graphs’ arc sets:

\[ \epsilon_{\text{miss}} = \Pr\{ e \in (E_t \cap E_u) \setminus E_m \} \]
\[ \epsilon_{fa} = \Pr\{ e \in E_m \setminus (E_t \cap E_u) \} \]  

(3.3)

where \( \setminus \) denotes the set difference operator: \( A \setminus B = A \cap \neg B \). Let \( G = (V, E) \) denote a candidate hypothesis \( G \in \mathcal{G}_c \), and let \( h(e) \) denote the indication function such that if \( e \in E \) then \( h(e) = 1 \), otherwise \( h(e) = 0 \). The likelihood of the measured graph, \( G_m \), for every \( G \in \mathcal{G}_c \) is then given by

\[ L\{G_m | G\} = \prod_{e \in E_u} L\{e | h(e)\}, \]

(3.4)

where the likelihood of each edge, \( L\{e | h(e)\} \), is given in Table 3.1. Note that the likelihoods of an arc to be consistent or inconsistent are not complementary, and do not sum to 1.

| \( L\{e | h(e)\} \) | \( h(e) = 0 \) | \( h(e) = 1 \) |
|-----------------|------------|------------|
| \( \epsilon_{fa} \) | \( 1 - \epsilon_{fa} \) | \( \epsilon_{\text{miss}} \) |
| \( \epsilon_{fa} \) | \( \epsilon_{fa} \) | \( 1 - \epsilon_{\text{miss}} \) |

Table 3.1: The likelihood, \( L\{e | h(e)\} \), assigned to each of the arcs given \( e(e) \) and the hypothesis \( G = (V, E) \).

The maximum likelihood graph clustering criterion (MLGC), defined by (3.1), specifies the grouping result, \( G_h \), but is not a constructive algorithm. Moreover, this problem belongs to the class of NP-hard problems (in the worst case analysis). We therefore address the theoretical and the practical aspects separately.

Considering the theoretical aspect, we assume that the (not necessarily unique) hypothesis \( G_h \), which maximizes the likelihood, can be found, and address the following question: “what is the relationship between the result \( G_h \) and the unknown target graph \( G_t \)?” This question is interesting and important, because it concerns the prediction of the grouping performance. If we can show that these two graphs are close in some sense, then it means that grouping algorithms which use this maximum likelihood principle may have predictable expected behavior. That is, although we cannot know \( G_t \), the grouping hypothesis \( G_h \) produced by such an algorithm is close enough to the true partitioning. This question is considered in detail in Chapter 4.

From the practical point of view, one should ask if this optimization problem can be solved in reasonable time. We shall show that this problem is NP-hard, in the worst case analysis. Similar problems have been solved by simulated annealing methods [GGGD90, HH93]. However, these algorithms are usually very slow. Heuristic
algorithms have also been tried [SH79, Vos92]. For our implementations, we use a heuristic algorithm, by which the optimal solution is not guaranteed.

Some other works also use graphs for grouping [UE96, EZ96, CZ96]. They look for groups that form closed paths (circles) in the graph, which much simplifies the optimization problem (polynomial time). These methods are suitable for the case where the groups are known to form circles in the graph. However, we do not make any assumption on the form of the groups in the graph. Furthermore, even in the following second example, the grouping of edge points lying on smooth curves, the groups are not closed curves, and a curve-like group of \( n \) edgels is definitely not just a path of \( n - 1 \) arcs in the graph. It is an actual subgraph which includes \( n \) nodes and many more arcs, connecting both successive and not successive nodes along the path. These additional arcs provide more perceptual information for the grouping process.

In the context of this thesis we use the binary cue model, which is useful because the reliability of a grouping cue is given by only two numbers. It requires only a little knowledge about the distribution of the measured property (e.g., the second eigenvalue \( \lambda_2 \), used in the above co-linearity example, (2.3)), hence making the method robust to changes in this distribution between different scenes. It also allows us to analyze the grouping quality in terms of cue reliability, and to compare the results of the analysis with the experimental results. However, for the more general case of a non-binary grouping cue, we shall define the Grouping Likelihood Graph.

### 3.4 The Grouping Likelihood Graph

The result of a non-binary grouping cue, \( c(e) \), cannot be described by the so-called measured graph. However, the generalization is straightforward. Let \( e \) denote an arc in \( G_u \). Note that the binary cue decision, \( c(e) \), provides two likelihoods for \( e \):

\[
L_c(e|h(e) = 1) \quad L_c(e|h(e) = 0),
\]

where \( h(e) \) is the hypothesis of \( e \) (i.e., \( h(e) = 1 \) if and only if \( e \in E \), where \( G = (V, E) \) is the hypothesized graph). Hence the result of a non-binary grouping cue, \( c(e) \), is represented by the two conditional likelihoods of (3.5), which depend on the (non binary) value of the cue function. Hence, the binary cue is a special case of a grouping likelihood graph (see Table 3.1).

Chapter 4

Analysis of the Grouping Method

The proposed maximum-likelihood graph clustering (MLGC) criterion is analyzed in the first section. The analysis quantifies the similarity between the (unknown) ground truth grouping, $G_t$, and the hypothesized grouping, $G_h$, obtained by this method. As we shall see, the dissimilarity depends on the error probabilities of the individual arcs, $\epsilon_{miss}, \epsilon_{fa}$, and on the connectivity, or the density, of $G_u$.

The second section provides the complexity analysis of the MLGC criterion. Some related work in graph clustering is discussed, and compared with our criterion from the algorithmic point of view.

4.1 Analysis of the Grouping Quality

In general, grouping performance is good for groups which are densely connected within the underlying graph, and is expected to be worse for loosely connected groups. If, for example, a node (data feature) is connected to its group by only one arc in the underlying graph, it may be separated from this group in the hypothesized partition with probability $\epsilon_{miss}$, which may be high.

The first result shows that good solutions are not rejected.

Claim 4.1.1 If $\exists G^* = (V, E^*) \in \mathcal{G}_c$ s.t. $E^* \cap E_u = E_m$, then

$$L\{G_m|G^*\} = \max_{G \in \mathcal{G}_c} L\{G_m|G\},$$

provided that $\epsilon_{miss}, \epsilon_{fa} < 0.5$.

Proof: For every clique graph, $G = (V, E) \in \mathcal{G}_c$,

$$\frac{L\{G_m|G\}}{L\{G_m|G^*\}} = \left( \prod_{e \in E_u \cap (E \setminus E^*)} \frac{\epsilon_{miss}}{1 - \epsilon_{fa}} \right) \left( \prod_{e \in E_u \cap (E^* \setminus E)} \frac{\epsilon_{fa}}{1 - \epsilon_{miss}} \right) < 1. \quad (4.2)$$

Arcs of $E_u$, which exist in both (or none) of the two sets, $E$ and $E^*$, do not affect that ratio, and therefore are not counted. \[\square\]
To borrow the terminology of parameter estimation, this claim shows that the MLGC is a consistent estimator. That is, arbitrarily reliable labeling of the underlying graph, associated with very good cues, leads to a correct decision. From now on we assume that $\epsilon_{\text{miss}}, \epsilon_{\text{fa}} < 0.5$; otherwise, consistency is not ensured. In the more realistic case, some of the cues disagree with the hypothesized groups. As we shall see, the grouping performance degrades gracefully with the quality (reliability) of the cues.

We now turn to prove a fundamental claim on which most of the other results rely. It is a necessary condition, satisfied by any MLGC partition. Consider two disjoint node subsets $V_i, V_j \subset V$ of the graph $G = (V, E)$, and denote their cut by

$$J(V_i, V_j) = \{ e = (v_1, v_2) \mid v_1 \in V_i, v_2 \in V_j \}.$$  

Let

$$l_u(V_i, V_j) = | J(V_i, V_j) \cap E_u |$$

denote the cut width relative to the underlying graph. Similarly, let

$$l_m(V_i, V_j) = | J(V_i, V_j) \cap E_m |$$

denote the cut width relative to the measured graph ($l_m \leq l_u$) (see Figure 4.1(left)).

Then,

**Claim 4.1.2 necessary condition:**

Let

$$G_h = (V, E_h), \ V = V_1 \cup V_2 \cup \ldots, \ E_h = E_c(V_1) \cup E_c(V_2) \cup \ldots$$

be the MLGC hypothesis (satisfying (3.1)), and let

$$\alpha = \left( 1 + \frac{\log(\epsilon_{\text{fa}}/(1 - \epsilon_{\text{miss}}))}{\log(\epsilon_{\text{miss}}/(1 - \epsilon_{\text{fa}}))} \right)^{-1}. \quad (4.3)$$

Then,

1. For any bisection of any group $V_i = V_i^l \cup V_i^u$ ($V_i^l \cap V_i^u = \emptyset$),

$$l_m(V_i^l, V_i^u) \geq \alpha l_u(V_i^l, V_i^u).$$

2. For any two groups $V_i, V_j$, $i \neq j$

$$l_m(V_i, V_j) \leq \alpha l_u(V_i, V_j).$$

**Proof:**
Figure 4.1: Left: the cut involved in splitting one group into two groups Right: the value of \( \alpha(e_{\text{miss}}, e_{\text{fa}}) \) (see Claim 4.1.2 for both left and right).

To prove the first part, consider the likelihood ratio between two hypotheses: one is \( G_h \), and the other, denoted \( \tilde{G}_h \), is constructed from \( G_h \) by separating \( V_i \) into two different groups, \( V_i' \) and \( V_i'' \). Denote \( l_m = l_m(V_i', V_i'') \), \( l_u = l_u(V_i', V_i'') \). Then

\[
\frac{L\{G_m | G_h\}}{L\{G_m | \tilde{G}_h\}} = \prod_{e \in J(V_i', V_i'') \cap E_u} \frac{P_r\{e | E_h\}}{P_r\{e | \tilde{E}_h\}} = \left( \frac{1 - e_{\text{miss}}}{e_{\text{fa}}} \right)^{l_m} \left( \frac{e_{\text{miss}}}{1 - e_{\text{fa}}} \right)^{l_u - l_m}.
\]

This likelihood ratio is a non-decreasing function of \( l_m \) and is greater than 1 for \( l_m \geq \alpha l_u \). Therefore, if the claim is not satisfied, then \( \tilde{G}_h \) is more likely than \( G_h \), which contradicts the assumption of (3.1). The second part of the claim is proven in a similar manner.

This claim has several interesting implications. It shows, for example, that if a true group is erroneously split by the MLGC into two smaller groups, then a substantial subset of the arcs connecting the two is “missed”. Recalling that \( e_{\text{miss}} < 0.5 \) is the probability for missing each one of these arcs, then for a reasonably small \( e_{\text{miss}} \) probability, such a split would happen with an extremely low probability. Another implication relates to addition errors, that is merging a group \( V_i \) with an alien node \( v^* \). Here the claim requires that a substantial fraction of the edges in the cut \( J(V_i, \{v^*\}) \), of which none is in \( E_t \), will be included in \( E_m \). That is, it requires many false-alarm cues. We shall return to these examples when discussing the theoretical results. The parameter \( \alpha \), specifying the fraction of cut edges required to merge two subsets, reflects the expected error types; if the false alarm probability is equal to
the miss probability then $\alpha = 0.5$, but if the false-alarm probability is higher, then so is $\alpha$ (see Figure 4.1(right)). We shall use this necessary condition to show that choosing a sufficiently dense underlying graph can significantly improve the grouping performance. We will consider two cases: a complete underlying graph and a locally connected underlying graph.

### 4.1.1 A Complete Underlying Graph

A complete underlying graph connects every data feature with all others and provides the maximal information to the graph clustering stage. Therefore, it may lead to excellent grouping accuracy. On the other hand, as mentioned above, it is useful only for global grouping cues, such as being on the same straight line, being consistent with an affine motion model, etc. The quality of the grouping result may be evaluated by several measures, and the following claims consider some of them.

**Claim 4.1.3** Let $S_i$ and $V^*$ denote, respectively, a true group and a MLGC hypothesized group containing $k$ nodes of $S_i$. Then the probability that the group $V^*$ contains a particular additional alien node, $v^* \notin S_i$, is

$$P_{\text{addition}} = \sum_{i=k_{\min}}^{k} \binom{k}{i} \epsilon_{fa}(1 - \epsilon_{fa})^{k-i}$$  (4.4)

where $k_{\min} = \lceil \alpha k \rceil$.

**Proof:** Use Claim 4.1.2 with $V_i = V^* \setminus v^*$ and $V_j = \{v^*\}$, and note that $l_u = k$. Merging $V_i$ and $V_j$ requires that at least $\alpha l_u$ of the edges connecting them are included in $E_m$. This event happens with a binomial distribution. \qed

**Claim 4.1.4** Let $S_i$ and $V^*$ denote, respectively, a true group and a MLGC hypothesized group containing at least $k$ nodes of $S_i$. Then the probability that $V^*$ contains $k'$ nodes or more which are alien to $S_i$ is at most

$$P_{k'-\text{aliens}} \leq \sum_{j=k'}^{N-k} \binom{N-k}{j} \sum_{i=k_{\min}(j)}^{k_j} \binom{k_j}{i} \epsilon_{fa}(1 - \epsilon_{fa})^{k_j-i}$$  (4.5)

where $k_{\min}(j) = \lceil \alpha k j \rceil$.

**Proof:** Use Claim 4.1.2 with and $V_i = V^* \cap S_i$ and $V_j = V^* \setminus S_i$ to find the probability that a particular data subset $V^*_j \setminus S_i$ merges. Then take a worst-case approach, and sum these probabilities over all subsets of a particular size $j$ and over all sizes higher than $k'$. \qed
Figure 4.2: Upper bound on the probability for adding any $k'$ alien data features to a group of size $k$, using a complete underlying graph (Claim 4.1.4). The error probability is negligible for $k > 20$ (Here $\epsilon_{\text{miss}} = \epsilon_{\text{fa}} = 0.2$).

Claim 4.1.5 Let $S_i$ and $V^*$ denote, respectively, a true group and the MLGC hypothesized group containing the maximal number of data features from $S_i$. Then, the probability that $V^*$ contains $|S_i| - k'$ data features from $S_i$ is at most

$$P_{k'-\text{deletions}} \leq \binom{|S'_i|}{k'} \sum_{i=k_{\text{min}}}^{\alpha(|S_i|)-k'} \binom{|S'_i|-k'}{i} \epsilon_{\text{miss}}^i (1-\epsilon_{\text{miss}})^{|S_i|-k'i} \quad (4.6)$$

where $k_{\text{min}} = \lceil \alpha(|S_i|)-k' \rceil$.

Proof: For any particular deleted subset $S' \subseteq S_i$, $|S'| = k'$, use Claim 4.1.2 with $V_i = S_i \setminus S'$ and $V_j = S'$, and note that $l_u = (|S_i| - k')k'$. Such a split of $S_i$ requires that $l_m \leq \alpha l_u$. This event happens with a binomial distribution. To find the probability that some subset of size $k'$ is deleted, we sum over all subsets, ignoring the dependency between the events, which can only decrease this probability.

Claims 4.1.3, 4.1.4, and 4.1.5 simply state that if the original groups $S_i$ are large enough and the miss and false alarm probabilities are small enough, then it is very likely that the maximum likelihood partition will include one group for each object, containing most of its elements, and very few aliens. One example is given by the crude bound, plotted in Figure 4.2(right). It shows that the probability for hypothesizing highly mixed subsets is small, even for substantial cue errors, provided that the group is large enough ($k \geq 20$).
Another performance measure, evaluated using some approximations, is the expected number of deletion errors,

\[
E\{k_{\text{delete}}\} = k \sum_{i=k_{\text{min}}}^{k-1} \binom{k-1}{i} \epsilon_{\text{miss}}^i (1 - \epsilon_{\text{miss}})^{k-1-i}
\]  \hspace{1cm} (4.7)

where \(k_{\text{min}} = \lceil \alpha(k-1) \rceil\). Similarly, the expected number of addition errors is

\[
E\{k_{\text{add}}\} = (N - k) \sum_{i=k_{\text{min}}}^{k'} \binom{k'}{i} \epsilon_{\text{fa}}^i (1 - \epsilon_{\text{fa}})^{k'-i}
\]  \hspace{1cm} (4.8)

where \(k\) is the group size, \(k' = k - E\{k_{\text{delete}}\}\) and \(k_{\text{min}} = \lceil \alpha k \rceil\). Figure 4.3 shows how \(E\{k_{\text{delete}}\}\) and \(E\{k_{\text{add}}\}\) are affected by the reliability of the grouping cue. Note that \(E\{k_{\text{add}}\}\) increases while \(\epsilon_{\text{miss}}\) and \(\epsilon_{\text{fa}}\) increase, as one could intuitively expect. However, \(E\{k_{\text{delete}}\}\) decreases while \(\epsilon_{\text{fa}}\) increases. The evaluation of \(E\{k_{\text{delete}}\}\) by (4.7) ignores arcs of \(G_m\) connecting elements of the true group, say \(S_i\), to elements of other groups and clutter. It is a reasonable assumption when \(\epsilon_{\text{fa}}\) is small enough, because there is a small chance that another group will “pull” an element from \(S_i\). In (4.7), \(\epsilon_{\text{fa}}\) has an indirect effect, by changing \(\alpha\) (and therefore changing the grouping decision criterion). When \(\epsilon_{\text{fa}}\) increases, \(\alpha\) increases, \(k_{\text{min}}\) increases, and therefore \(E\{k_{\text{delete}}\}\) decreases.

Complete underlying graphs provide the maximal information to the grouping process, and therefore improve the grouping quality. They have two disadvantages: one is the computational time needed to evaluate all the grouping cues, which might be impractical in large graphs. However, the major difficulty we see with the use of a complete underlying graph is that it does not apply to all types of grouping cues. It is especially concerned with cues that have only local meaning, such as co-circularity and proximity for smooth curve detection. Therefore, another option, the locally-dense underlying graph, is also proposed.

### 4.1.2 Locally Dense Underlying Graphs

An intuitive choice of an underlying graph which is less dense than the complete graph is to connect every data feature only to those data features in its neighborhood, either to the closest \(k\) data features, or to all data features in a certain radius. This type of graph is used, for example, for the grouping of edge points lying on a smooth curve (see the 2nd grouping example in the experimentation section). When specifying such a graph, it is important to keep a substantial connectivity between the data features of objects so that accidental deletion will be less likely. This connectivity demand is quantified by requiring the projection of every group on the underlying graph, \((V_i, E_c(V_i) \cap E_a)\) to be \(k\)-connected. That is, if any \(k-1\) nodes are eliminated, then this projected subgraph remains connected. A nice property of \(k\)-connected graphs is
that every cut in them contains at least \( k \) edges. Therefore, deleting a node requires at least \( \alpha k \) miss errors. Alien data features are either densely connected to a group, implying that their incorrect addition to a group is prevented with high confidence, or are not connected enough and are not considered at all as candidates for addition.

A significant change from the case of complete graph is that \( \alpha k \) miss errors can cause the deletion of a subgroup containing more than one data feature. Therefore, we have chosen to characterize the grouping performance by another measure: the expected number of “large” subgroups into which the group decomposes. Consider a particular cut of size \( k \) in the projection of some object on the underlying graph. The probability that the object is divided in this cut into two parts is

\[
P_{k-\text{divide}} = \sum_{i=k_{\text{min}}}^{k} \binom{k}{i} e_{\text{miss}}^i (1 - e_{\text{miss}})^{k-i}
\]

(4.9)

where \( k_{\text{min}} = \lfloor \alpha k \rfloor \).

Suppose now that we can estimate the number of “potential cuts”, and denote this number by \( N_{\text{cut}} \). Then, the expected number of group separations will be simply \( N_{\text{cut}} P_{k-\text{divide}} \). Fortunately, such an estimate may be done for the interesting case of curve-like groups. A group is said to be curve-like if its elements can be sorted along a curve in the image plane. The “length” of the group is its size, \( \| S_i \| \). A typical example is again the grouping of edge pixels lying on a smooth curve. Here
the grouping process is 1D, in contrast to area based segmentation [Zuc83]. Let $S_i$ be a long $k$-connected curve-like group. This group is likely to break into a number of shorter (smaller) groups, in $N_{\text{cut}} = \|S_i\| - 1$ different potential places. Each break is associated with a cut of size $k$ or more in the graph. Therefore, the expected number of parts into which the curve decomposes is not higher than

$$N_{\text{cut}} P_{k-\text{divide}} = (\|S_i\| - 1) \sum_{i=k_{\text{min}}}^{k} \binom{k}{i} \epsilon_{\text{miss}}^i (1 - \epsilon_{\text{miss}})^{k-i}.$$ 

This number, plotted in Figure 4.4(Left), generally decreases when the cut size $k$ increases (but is not strictly monotonic, due to the non-constant and non-monotonic nature of the ratio $\frac{k_{\text{min}}}{k} = \left[ \frac{\alpha k}{k} \right] \neq \alpha$). It is useful for choosing a practical value for $k$.

![Figure 4.4](image)

**Figure 4.4**: A $k$-connected curve-like group (e.g. smooth curve) is likely to break into a number of sub-groups. The graph shows an upper bound on the expected number of sub-groups versus the minimal cut size in the group, $k$ ((4.9)). Here the group size (length) is 400 elements (i.e., edge points), $\epsilon_{\text{miss}} = 0.14$ and $\epsilon_{\text{false}} = 0.1$ (typical values for images like Figure 6.7). It shows how increasing connectivity quickly reduces the false division of this type of group.

### 4.2 Complexity Analysis of The MLGC Criterion.

In this section, the maximum likelihood graph clustering problem (MLGC) is discussed from the algorithmic point of view, and compared with other known graph clustering problems. For convenience, we shall first express the MLGC by the common notations used in graph theory. The MLGC problem, defined in Chapter 3, is as follows: given an undirected graph $G = (V, E)$ with nonnegative pairs of arc weights, $w_0(e), w_1(e), \ e \in E$, find a positive integer $k$ and a partition $S = S_1 \cup S_2 \cup \ldots S_k$ of $G$ into $k$ connected components, with a minimum total weight.
\[ wt(S) = \sum_{i=1}^{k} \sum_{u < v, u \in V_i, v \in V_j} w_1(e) + \sum_{j=1}^{k} \sum_{u \in V_i, v \in V_j} w_0(e). \] (4.10)

**Remark 4.2.1:** This slightly different formulation replaces the previous notations of underlying graph and measured graph. The arc weights are the negative log-likelihoods of the grouping cue. For the simple case of a binary bi-feature grouping cue, one should assign the pair of weights \( w_0(e), w_1(e) \) of each arc \( e \in E_m \) with

\[ w_0(e) = -\log(\epsilon_a) \quad w_1(e) = -\log(1 - \epsilon_{miss}), \] (4.11)

and of each arc \( e \notin E_m \) with

\[ w_0(e) = -\log(1 - \epsilon_a) \quad w_1(e) = -\log(\epsilon_{miss}). \] (4.12)

Note that in this context, the “noise” data features, which do not belong to any of the grouped objects, should intuitively be considered as many “connected components”, composed of one (or very few) elements. The number of groups, \( k \), specified for later use, is a natural result of the MLGC process.

This form can be simplified as follows: let \( w(e) = w_0(e) - w_1(e) \). Let \( wt_1 = \sum_{e \in E} w_1(e) \) denote the total weight of the trivial partition into one group (which is not optimal, in general). Then the objective function takes the form

\[ wt(S) = wt_1 + \sum_{i=1}^{k} \sum_{u < v, u \in V_i, v \in V_j} w(e). \]

In (4.2) the weights \( w(e) \) correspond with log-likelihood ratios:

\[ w(e) = w_0(e) - w_1(e) = \begin{cases} -\log(\frac{\epsilon_a}{\epsilon_{miss}}) & e \in G_m \\ -\log(\frac{1 - \epsilon_a}{\epsilon_{miss}}) & e \notin G_m \end{cases}. \]

In the next sections we use this form to compare the MLGC criterion with other graph clustering techniques, and to prove that the MLGC criterion is NP-hard (like some other graph clustering criterion). A heuristic algorithm for the MLGC, which finds a partitioning that is not guaranteed to be the optimal one, is presented in Chapter 6.

**4.2.1 Related Work on Graph Clustering Methods**

The form of (4.2) is similar to the well known \( k \)-cut problem (e.g., see the work and references of Saran and Vazirani [SV95]): given an undirected graph \( G = (V, E) \) with nonnegative arc weights and a positive integer \( k \), find a set \( E_0 \subseteq E \), of minimum
weight, the removal of which leaves $k$ connected components. The $k$-cuts problem is NP-Hard, and there are polynomial time variants of the $k$-cuts problem for a fixed $k$ (see [SV95] for references). However, these variants are not efficient for large $k$-s, as can be ascertained from the above remark. Saran and Vazirani proposed an approximation to the $k$-cuts problem, which finds a $k$ cut weighting within a factor of $(2 - 2/k)$ of the optimal. They show that this is a tight bound of their algorithm. This approximation algorithm is based on the Gomory-Hu multi-terminal max-flow algorithm [GH61], which needs to solve $n$ max-flow problems, and therefore has a running time of $O(|E||V|^2 + |V|^3 + \epsilon)$, for any fixed $\epsilon > 0$.

It is interesting to note that the very same algorithm was developed independently by Wu and Leahy, for image segmentation [WL93], two years before it was introduced and analyzed by Saran and Vazirani. Wu and Leahy also implemented the Gomory-Hu algorithm, and show good experimental results on synthetic and real images. However, they wrongly claim that this algorithm finds the global optimum partition of the graph into $k$ clusters. As we know from Saran and Vazirani, this algorithm only finds a good approximation for the optimal $k$-cuts problem.

Herault and Niez [HN89] have considered another $k$-partitioning problem which is a more restricted form of the $k$-cuts problem: given a graph $G = (V, E)$ with positive weights on its arcs, and $k$ integers, $N_1 + N_2 + \ldots + N_k = |V|$, find a partition of the graph nodes into $k$ disjoint subsets of these specified sizes, which minimizes the total weight of the cut. This problem is NP-complete, and they compare several optimization methods for solving it. Most of these methods are annealing-based, stochastic or deterministic, and one is a neural-network approach. Their approach, however, is less suitable for grouping than the $k$-cuts, as the number of the groups and their sizes are usually unknown.

Matula has made a comprehensive study of graph clustering methods (e.g., [Mat77]). He has modeled relations between data features by random graphs. Every arc in the graph is associated with a probability, which represents the belief that this pair of nodes belongs to the same object. The task is to find connected clusters in the graph. There are different definitions of connectivity, i.e., simple connected components, $k$-bond (nodes’ degrees in the cluster are $k$ or more), $k$-connected components (every cut width in the cluster is of $k$ arcs or more), and $k$-clique components. These connectivity definitions are hierarchical: every $k+1$-clique is a subgraph of a $k$-connected component, which is a subgraph of a $k$-bond component, which is a subgraph of a simple connected component. It is shown that each of these criterion leads to a stratified clustering method, that is, the $k$-x groups are refinements of $k - 1$-x groups. The main result deals with the properties of stratified clustering methods, and how the clustering can be evaluated in a hierarchical order. In addition, it is argued that a good connectivity criterion must be robust to a small number of missed arcs, and therefore cannot be based on simple connected components, or on cliques. Both are
too sensitive to the presence or the absence of even a single arc.

Zahn has used other graph clustering methods for perceptual grouping [Zah71]. He used methods that are based on the minimal spanning tree, and demonstrated their efficiency on synthetic dot patterns. One method is to build a k-nearest neighbors graph, based only on proximity, and then to divide the graph into strongly connected components. A second method is to evaluate a minimal spanning tree, and then use it to divide the graph nodes into two or more clusters. Urquhart [Urq82] pointed out some of the problems with these simple methods, and suggested an improved approach, using the Gabriel graph. Ahuja [Ahu82] has also considered the Gabriel graph, and compared it with the Voronoi neighborhood and the k-nearest neighbors. Unlike with the minimal spanning tree, where arcs are removed by a global criterion of length threshold, the improved approaches consist of an adaptive, local clustering criterion. These methods are more robust, and adaptive to clusters having different point densities. Although some of these methods might look very naive, their simple and efficient implementation makes them attractive whenever grouping is based solely on a local proximity criterion (in some metric).

Jerrum and Sorkin [JS93] have used simulated annealing for graph bisection into two equal-size parts. They consider random graphs in $G_{npr}$, which contain two $n/2$-vertex sets, connected by internal arcs with density $p$ and cut arcs of a slightly lower density $r$. They show that for some range of $r$ and $p$, this problem can be solved with a high probability using the Metropolis algorithm (a variant of simulated annealing, using a fixed temperature). The MLGC may be considered as a variant of this problem, where the number of groups is larger than 2 (a generalization), and the groups may not be of the same size. The later described MLGC heuristic algorithm can be modified to a simulated annealing algorithm in a straightforward manner, by adding a random acceptance criterion. Moreover, their analysis, which is based on a Markov chain of bisections, may also be adopted to analyze our heuristic algorithm. These are interesting directions for future study.

Gotsman [Got91] has used the theory of percolation in random graphs to find clusters of points, embedded in a slightly less dense background set of points. In a sense, this is a harder case of point clustering than was studied by Zahn and by Urquhart, who assume that the clusters are more or less isolated from each other, with no spurious points between them. Gotsman constructs a graph by connecting each point to all other points within radius $r$. All connected components of the resulting graph of size $\leq t$ are discarded. The remaining components are valid clusters. This algorithm applies Zahn's connectivity criterion to graphs similar to those used by Urquhart. However, it differs from the above methods in that the values of $t$ and $r$ are not tuned manually, but are explicitly determined from the given set of points. This is done by modeling the background and the clusters of points by Poisson processes. Gotesman's interesting result arises from percolation theory in random
graphs (more about percolation theory and on graph theory can be found in [Gri89, Bol85]). It is shown that there is a high probability that the clusters be connected in the slightly denser regions, while at the same time there is a low probability of separated clusters being wrongly connected by spurious background points. The practical results, demonstrated on synthetic images, are very promising. It is not clear, however, how this approach can be used for the grouping of points having other distributions, or with data features other than points.

Cox and Zhong [CZ96] have proposed an algorithm to find a single closed edge contour which minimizes the ratio between the exterior boundary cost and the interior benefit. A closed contour in the image is a circle in the corresponding graph. The pinned ratio algorithm is used to find the best circle passing thru a particular node in the graph. They apply it to every node in the graph and choose the best result. This polynomial-time algorithm can easily incorporate spatial constraints given interactively by a user. It provides good results on medical images. The main reported problem is how to determine the ratio between increasing the benefit of the interior to increasing the cost of the boundary, with respect to a given practical domain. Despite this inherent limitation, this algorithmic approach might be also useful with other graph clustering methods.

Kortsarz and Peleg [KP93] discuss other criterion for graph clustering, like the heaviest subgraph (HS) and the densest subgraph (DS). The heaviest $k$-nodes subgraph of a given undirected weighted graph is the $k$-nodes subgraph with highest weight. This problem is NP-hard, even in the unweighted case. They present a number of polynomial time approximation algorithms for HS. The density of a subgraph is equal to the ratio between its weight and the number of its nodes. The DS problem is concerned with choosing a subset with maximum density. This problem can be solved in polynomial time. However, the density criterion is less appropriate for grouping than the heaviest subgraph. For example, the union of two disjoint subgraphs having the same density will also have the same density. Furthermore, in a large graph of a bounded degree, this criterion is very insensitive to the addition/deletion of nodes to a group (as the arcs-to-nodes ratio remains nearly constant). The $k$-DS problem, where the size of the subgraph should be at most $k$, is NP-complete. In general, one would like to have a graph clustering criterion that prefers larger groups, which are also highly connected and dense, and can be computed or approximated in a relatively weak polynomial time.

Shapiro and Haralick [SH79] have used a graph theoretic method to decompose the boundary of a shape into “intuitively pleasing” simple parts. Their method is motivated by previous work, where clusters are found by iteratively merging cliques having enough overlap. However, finding maximal cliques is NP-complete. In [SH79] a graph is constructed using a visibility criterion - every two points such that the line segment joining them is inside the shape are connected with an arc in the graph.
They define a connectivity measure that serves as an upper bound on the maximal size of a clique having these two nodes. The iterative algorithm merges initial clusters with a lot of overlap, until it naturally stops with a few clusters having not many overlapping parts. A similar connectivity criterion is used in our implementation for finding good seeds and growing new groups.

It is worth noting that some of the above graph clustering methods, which have been successfully applied to point clustering in the plane, might not work in other grouping domains (e.g., [Zah71, Urq82, Got91]). The reason is that either the graph or the clustering method depends on the spatial location of the points, and is not suitable for data features other than points. The MLGC algorithm is designed for abstract graph representation, independent of the features’ type and location, and is therefore useful in many grouping domains.

The above graph clustering techniques may be divided into two classes; the local clustering methods and the global optimization methods. Usually, the global optimization methods are more computationally demanding than the local clustering methods. The MLGC is a global optimization method, based on the maximum likelihood criterion. The \( k \)-cuts and the \( k \)-partitioning are the closest well-studied graph clustering problems to MLGC that we know\(^1\). There are, however, two main issues in which they differ from MLGC: In MLGC the arc weights might be negative, and \( k \) is not specified, but is ascertained as a part of the result (obviously the sizes of the groups in not known in advance, as required by the \( k \)-partitioning method). The main difference arises from the arcs’ signed weights. With only positive weights, it is clear that if \( k_1 < k_2 \), then the weight of the optimal \( k_1 \)-cut is smaller than the weight of the optimal \( k_2 \) cut. Therefore, if one solves for every \( k = 2, 3, \ldots, N - 1 \), there is will be no non-trivial preference for a specific optimal \( k \). In the method proposed here, the negative and positive weights provide a balance which directs the clustering process to find the optimal number of clusters, \( k_{opt} \).

### 4.2.2 The MLGC criterion is NP-hard

We shall prove that MLGC is NP-hard by showing a reduction from \( k \)-colorability to MLGC. The \( k \)-colorability problem is as follows: given a graph and an integer \( k \), find a partition of the graph nodes into \( k \) sets such that there is no arc in \( E \) connecting two nodes in the same set (in other words, this is a maximal \( k \)-cut).

Given \( k \) and \( G = (V, E) \) for \( k \)-colorability, a we shall solve the following MLGC problem: let \( n = |V| \) and \( m = |E| \) denote the number of nodes and the number of arcs in the graph. Let \( G' = (V', E', w(e) : E' \to \mathbb{R}) \) denote an MLGC problem, defined as follows: the nodes set is

\[
V' = V \cup \{v'_1, v'_2, \ldots, v'_k\}
\]

\(^1\)Another related problem is the \( k \)-separator [ENRS96]
where \( v'_i \) denote \( k \) additional nodes. The arcs set is
\[
E' = E \\
\cup \{(v, v'_i) | v \in V, \ i = 1, 2, \ldots, k \} \\
\cup \{(v'_i, v'_j) | i = 1, 2, \ldots, k-1, \ i < j \},
\]
and the arc weights are
\[
w(e) = \begin{cases} 
1 & e \in E \\
-m & e = (v, v'_i), \ v \in V, \ i = 1, 2, \ldots, k \\
2nm & e = (v'_i, v'_j) \ i \neq j
\end{cases}
\]

The graph \( G' \) can be constructed in polynomial time \( (O(n^2)) \). Let \( S' = \{S'_1, S'_2, \ldots, S'_l\} \) denote the optimal partition of \( G' \), found by the MLGC (as defined in (4.2)). This partition has the maximum cut weight in \( G' \). First we shall show that the optimal partition of \( G' \) has exactly \( k \) connected components (i.e., \( l = k \)), each including one of the additional nodes \( v'_i \). Then we shall show that its total weight, \( wt(S') \), is \( m + (k-1)^2 nm \) if and only if \( G \) is \( k \)-colorable.

**Lemma 4.2.2** The MLGC optimal partition \( S' \) has exactly \( k \) connected components, each containing one of the additional nodes \( v'_i \).

**Proof:** Suppose that there is a group \( S'_j \) which includes \( k' > 1 \) additional nodes. If we split \( S'_j \) into two groups, the first including only one of the additional nodes, and the second including the other \( k' - 1 \) additional nodes, then the effect on the overall weight \( wt \) (4.2) would be
\[
\Delta wt \geq (k' - 1)2nm - (k' - 1)|S'_j \cap V|m \tag{4.13}
\]

The splitting would increase \( wt \), in contrast to the assumption that \( S' \) is the maximum. Therefore, \( l \geq k \). On the other hand, suppose that there is a group, say \( S'_j \), that contains none of the additional nodes \( v'_i \). By merging it with another group which contains at least one additional node, the effect on the overall weight \( wt \) would be
\[
\Delta wt \geq |S'_j|m - m > 0.
\tag{4.14}
\]

The merging would increase \( wt \), in contrast to the assumption that \( S' \) is the maximum. Therefore, each of the groups \( S'_j \) in the optimal partition \( S' \) must include at least one of the additional nodes, i.e., \( l \leq k \). Hence we have \( l = k \), and each of the groups \( S'_j \) must contain exactly one additional node. \( \Box \)

**Lemma 4.2.3** The total weight of the optimal partition, \( S' \), found by MLGC in \( G' \) is \( wt(S') = m + (k-1)^2 nm \) if and only if \( G \) is \( k \)-colorable.
Proof: From Lemma 4.2.2 we can assume, without loss of generality, that \( v'_j \in S'_j \). The arc set of the cut, found by the MLGC includes three types of arcs:

- For every node \( v \in S'_j \cap V \), the cut contains \( k - 1 \) arcs of the form \((v, v'_i)\), \( i \neq j \) having a total weight of \(-(k - 1)nm\).

- The cut contains all the arcs of the form \((v'_i, v'_j)\), \( i = 1, \ldots, k - 1, i < j \), having a total weight of \( \frac{1}{2}k(k - 1)2nm = k(k - 1)nm \).

- The rest of the cut arcs are a subset of \( E \). Let \( 0 \leq W \leq m \) denote their number (also their weight).

By summing the weights of these three edge sets that compose the cut we have \( wt(S') = W + (k - 1)^2nm \).

If \( G \) is \( k \)-colorable, then there exists a \( k \)-cut \( S = \{S_1, S_2, \ldots, S_k\} \) of weight \( m \) in \( G \). By selecting \( S'_j = S_j \cup \{v'_j\} \) we have \( W = m \), and therefore \( wt(S') = m + (k - 1)^2nm \).

The second direction is proved similarly.

The next theorem is a direct result of Lemma 4.2.3:

**Theorem 4.2.4** The \( k \)-colorability problem is polynomially transformable to MLGC.

The MLGC problem is therefore NP-hard. This result was not surprising to most of the researchers in the field of graph theory to whom we spoke. It is important to note, however, that the above analysis holds for the worst case. In practice, the problem might be much simpler. If, for example, the grouping cue is very reliable, that is both \( \epsilon_{miss} \) and \( \epsilon_{fa} \) are small, and the graph connectivity is high enough, then finding the best clustering could be relatively simple. In fact, if one defines a distribution over the population of measured graphs, then some of the worst case examples are very unlikely to be seen. Finding the conditions in which this problem is simple (if any), either in worst case or in average case, and finding an approximation algorithm to the MLGC are subjects for future research.
Chapter 5

The Cue Enhancement Procedure

The performance of the grouping algorithm depends strongly on the reliability of the cues available to it. In many situations this reliability is predetermined and the grouping algorithm designer can only prefer the more reliable cues from the available variety. This section, however, shows how the reliability of a grouping cue can be significantly improved by using statistical evidence accumulation techniques. This procedure integrates the evidence available from raw multi-feature cues into very reliable bi-feature cues. Two of the three domain specific grouping algorithms that we implemented as examples (co-linearity and smoothness) use this procedure. This method is not restricted only to improving our grouping algorithm, and can be used to improve other grouping algorithms as well.

5.1 Overview

The CEP considers one pair of data features at a time, and tries to use the other data features in order to estimate the consistency of this pair. The idea behind the following process of evidence accumulation is that a random data subset $A$ that contains the data pair $e = (v_1, v_2)$, as well as at least one additional data feature, may be consistent only (but not necessarily) if $e$ itself is consistent (see Figure 5.1(left)). Therefore, a multi-feature cue, operating on a feature subset $A$, carries statistical information on the consistency of $e$. Recall that $c_m(A)$ denotes the raw multi-feature cue (e.g. Figure 5.1(right), and $c(e)$ denotes the enhanced bi-feature cue. Thus, the CEP uses $c_m(A)$ cues to determine the value of the cue $c(e)$.

The algorithm is conceptually simple: for every data pair $e = (v_1, v_2)$ in the underlying graph, the algorithm draws several random data subsets $A_1, A_2, \ldots$, of size $m > 2$, which contain the pair $e$. Then the corresponding multi-feature cues $c_m(A_1), c_m(A_2), \ldots$ are extracted. The cue values are deterministic functions of the subsets $A_1, A_2, \ldots$, but may also be considered as instances of a random variable, the statistics of which depend on the data pair $e$, and in particular on its consistency. The
Figure 5.1: An illustration of the CEP for obtaining a very reliable smoothness cue: The left image contains several edgels, some of which belong to the same group: one smooth curve. Subsets from this group, such as \( e \) and \( A_j \) but not \( A_i \), are denoted consistent. The CEP extracts a raw multi-feature cue for several feature subsets which contain the feature pair \( e \), and decides about the consistency of \( e \) from these cues. (In this example the cue is extracted for several subsets of three edgels such as \( A_i \) and \( A_j \)). One of the possible raw three-feature smoothness cues, which is used in the experiments section, is illustrated in the right image. This cue tests co-circularity by testing the consistency of the edgels’ directions with the tangent to a circle drawn through their centers. 

\[
\phi = \max \{ \phi_1, \phi_2, \phi_3 \}.
\]

The number of random data subsets (and their associated cues) required for a conclusive reliable decision on the consistency of \( e \), is determined adaptively and efficiently by a well-known method for statistical evidence integration: Wald’s SPRT test.

### 5.2 Wald’s SPRT and its Application to Cue Enhancement

Consider a random variable, \( x \), whose distribution depends on an unknown binary parameter, \( \Omega \), which takes a value of \( \omega_0 \) or \( \omega_1 \). Every instance of the random variable carries statistical information on this parameter. Integrating this information along a sequence of instances of the random variable will eventually lead to a reliable inference about \( \Omega \). An efficient and accurate procedure for integrating the statistical evidence is the *Sequential Probability Ratio Test* (SPRT) suggested by Wald [Wal52]. This procedure quantifies the evidence obtained from each trial by the log-likelihood ratio function of its result

\[
h(x_i) = \log \frac{P_1 \{ x_i \}}{P_0 \{ x_i \}}, \quad (5.1)
\]
where
\[ P_0\{x_i\} = Pr\{x_i|\omega_0\} \]
\[ P_1\{x_i\} = Pr\{x_i|\omega_1\} \quad (5.2) \]
are the probability functions of the two different populations and \( x_i \) is the value assigned to the random variable \( x \) in the \( i \)th trial.

The log-likelihood ratio is high when the value of the instance \( x_i \) of the random variable \( x \) is likely for one hypothesis (\( \omega_1 \)) and is not likely for the other (\( \omega_0 \)). It is negative and low when the situation is reversed. If the probabilities of obtaining \( x_i \) under both hypotheses are close, then \( x_i \) carries only a little information and \( h(x_i) \approx 0 \). When several trials are carried out, the log-likelihood function of the composite event \( \mathbf{x} = (x_1, x_2, \ldots, x_n) \) should be considered. If, however, the trials are independent, then this composite log-likelihood function is equal to the sum of the individual log-likelihood functions,

\[ \sigma_n = \sum_{i=1}^{n} h(x_i). \]

The sum \( \sigma_n \) serves as the statistic by which the decision is made. Wald's procedure specifies upper and lower limits, denoted \( a \) and \( b \), respectively. If the cumulative log-likelihood function crosses one of these limits, a decision is made. Otherwise, more trials are carried out. More formally, denote the decision made by the procedure (which is also the resulting binary enhanced cue) by \( D(\mathbf{x}) = D(x_1, x_2, \ldots, x_n) \in \{0,1\} \), and let the allowed probabilities of a decision error be

\[ \epsilon_{\text{miss}} = Pr\{\omega_1 \land D(\mathbf{x}) = 0\} \quad \epsilon_{fa} = Pr\{\omega_0 \land D(\mathbf{x}) = 1\}. \quad (5.3) \]

In the context of the CEP, these probabilities quantify the reliability of the enhanced cue. Therefore, using the same notations (\( \epsilon_{\text{miss}}, \epsilon_{fa} \)) as in Section 2.2 is justified (where \( c(e) \) is the CEP result). The algorithm is given simply by this iterative rule:

\[ \begin{align*}
  & if \sigma_n \geq a \quad \Rightarrow \quad \text{stop with } D(\mathbf{x}) = 1 \\
  & if \sigma_n \leq b \quad \Rightarrow \quad \text{stop with } D(\mathbf{x}) = 0 \\
  & else \quad \text{test for another subset}
\end{align*} \quad (5.4) \]

The upper and lower limits, \( a > 0 > b \), depend only on the allowed error probabilities, \( \epsilon_{\text{miss}}, \epsilon_{fa} \) and do not depend on the distribution of the random variable \( x \). We calculate \( a, b \) using a practical approximation, proposed by Wald, which is very accurate when \( \epsilon_{\text{miss}}, \epsilon_{fa} \) are small:

\[ a = \log\left(\frac{1 - \epsilon_{\text{miss}}}{\epsilon_{fa}}\right) \quad b = \log(\epsilon_{\text{miss}}(1 - \epsilon_{fa})). \quad (5.5) \]
The basic SPRT algorithm terminates with probability 1, and is optimal in the sense that it uses the minimum expected number of tests necessary to obtain the required decision error. This expected number of tests is given by:

\[
E \{n | \omega_0 \} = \frac{[a \epsilon_{fa} + b(1 - \epsilon_{fa})]}{\eta_0} \quad E \{n | \omega_1 \} = \frac{a(1 - \epsilon_{miss}) + b \epsilon_{miss}}{\eta_1},\]  

(5.6)

where \(\eta_0, \eta_1\) are the conditional expected amounts of evidence from a single test: \(\eta_0 = E\{h(x)| \omega_0 \}\) and \(\eta_1 = E\{h(x)| \omega_1 \}\). Despite its average-case optimality, the worst-case number of trials required by the SPRT algorithm is not bounded. To deal with this disadvantage, the modified Truncated SPRT [Wal52], which uses a predefined upper bound \(n_0\) on the number of tests, is used. We set \(n_0\) to be a few times larger than \(\max\{E \{n | \omega_0 \}, E \{n | \omega_1 \}\}\).

In the context of the cue enhancement procedure, the unknown parameter \(\Sigma\) is the consistency of the pair \(e = (v_1, v_2)\): the hypothesis \(\omega_0\) stands for \(t(e) = 0\), and the hypothesis \(\omega_1\) stands for \(t(e) = 1\). The random variable is \(x_i = \epsilon_m(A_i)\). In order to use the CEP, the user should specify the two distributions of the raw multi-feature cue, \(P_0 \{\epsilon_m(A)\}, P_1 \{\epsilon_m(A)\}\), and the desired reliability \((\epsilon_{miss}, \epsilon_{fa})\) of the enhanced cue. Specifying the raw cue distributions involves some technical details and is given in Section 5.3. Then, (5.5),(5.1) and (5.6) should be used to calculate the limits \(a, b, h\) and \(n_0\). The SPRT-based cue enhancement procedure is summarized in Figure 5.2. A cue optimization procedure is described in Section 5.4.

---

For every feature pair \((v_1, v_2)\) in the underlying graph:

1. Set the evidence accumulator, \(\sigma\), and the trials counter, \(n\), to 0.
2. Randomly choose \(k - 2\) data features \(v_3, \ldots, v_k \in S \setminus \{v_1, v_2\}\)
3. Calculate the multi-feature cue \(c = C_m(\{v_1, v_2, v_3, \ldots, v_k\})\).
4. Update the evidence accumulator \(\sigma = \sigma + \log \frac{P_0(c)}{P_k(c)}\).
5. if \(\sigma \geq a\) or \((n \geq n_0\) and \(\sigma > 0\)), output: \((v_1, v_2)\) is consistent.
   if \(\sigma \leq b\) or \((n \geq n_0\) and \(\sigma < 0\)), output: \((v_1, v_2)\) is inconsistent.
   else, repeat (2)-(5)

Figure 5.2: The cue enhancement algorithm (CEP)

The success of the CEP relies on the validity of the statistical model, and particularly on two assumptions which are necessary for the following claim:
Claim 5.2.1 Given that

(a) the statistics of the cue values evaluated over all data subsets containing a consistent (inconsistent) arc is approximately the same, and

(b) the cues extracted from two random consistent (inconsistent) subsets including the same feature pair are independent identically distributed (iid) random variables,

then the cue enhancement procedure can identify the consistency of the feature pair within any specified error tolerance, $\epsilon_{\text{miss}}, \epsilon_{\text{fa}}$, irrespective of the reliability of the raw cue, $c_m(A)$.

This surprising conclusion seems to contradict intuition, according to which arbitrarily low identification errors are impossible due to the finite amount of data in the image. Indeed, arbitrarily high performance is not possible, for it requires a large number of trials, which leads to a contradiction of the independence assumption.

Therefore, the reliability of the basic cue, $P_0\{c_m(A)\}, P_1\{c_m(A)\}$, is important because it leads to a lower expected number of trials, $E\{n\}$, which is both computationally advantageous and important to the validity of the statistical independence assumption. Indeed, our experiments show that the SPRT significantly improves the cue reliability but that the achievable error rate is not arbitrarily small (see Section 7.8). Both assumptions may be violated in practice. It may be, for example, that assumption (a) fails for some particular data features. In Figure 5.2, all four of the marked arcs, $e_{ij}$, are of the same population of the inconsistent pairs ($\omega_0$). For every one of these pairs, the SPRT tests triplets of points which contain this feature pair, using a co-linearity cue. It is assumed that the distribution of the co-linearity cue does not depend on the particular feature pair. However, more false co-linearities are expected to be associated with $e_{23}$, since it is closer to a co-linear group, and many of its tested triplets are therefore likely to be co-linear. Indeed, in our experiments, there were a few addition errors like this, but they do not have any major effect.

5.3 Calculating $P_0\{c_m(A)\}, P_1\{c_m(A)\}$

In the context of the cue enhancement procedure, the cue value is regarded as a random variable. Apart from specifying the desired reliability ($\epsilon_{\text{miss}}, \epsilon_{\text{fa}}$) and using (5.5) to calculate the two thresholds $a$ and $b$, one must supply the two distributions (for consistent and inconsistent feature pairs) from which the log-likelihood ratio can be determined. These distributions should be evaluated carefully: the distributions of the multi-feature cues, taken over the consistent and inconsistent populations and denoted respectively by $P_{\text{con}}\{c_m(A)\}$ and $P_{\text{incon}}\{c_m(A)\}$, are usually quite different. It is important, however, to observe that even if a feature pair $(v_1, v_2)$ is consistent,
Figure 5.3: Example for a failure of assumption (a), Claim 5.2.1 (see text).

A random set $A$ which includes it may not be consistent (see Figure 5.1). Therefore, these distributions should be modified as follows: a random set $A = \{e = \{v_1, v_2\} \cup \{v_3, \ldots, v_m\}\}$, containing a consistent feature pair $\{v_1, v_2\} \subset S_j$ and additional $m-2$ randomly selected data features $v_3, \ldots, v_m$, is consistent with probability

$$p^{(m)} \overset{\text{def}}{=} Pr\{t(\{v_1, v_2, \ldots, v_m\}) = 1 \mid t(\{v_1, v_2\}) = 1\} \geq \frac{s_i^{m-2}}{(m-2)^{s_i}}$$

where $N = \|S\|$ and $s_i = \|S_j\|$ denote the number of data features and the size of the true group, respectively. Therefore, the modified cue distributions, conditional relative to the consistency of the first two features, are

$$Pr\{c_m(A)\} = Pr\{c_m(A) \mid t(\{v_1, v_2\}) = 1\}$$

$$P_0\{c_m(A)\} = Pr\{c_m(A) \mid t(\{v_1, v_2\}) = 0\}.$$  \hfill (5.8)

These two distributions can either be estimated using a Monte-Carlo process, or calculated by

$$P_1\{c_m(A)\} = p^{(m)} P_{\text{con}}\{c_m(A)\} + (1 - p^{(m)}) P_{\text{incon}}\{c_m(A)\}$$

$$P_0\{c_m(A)\} = P_{\text{incon}}\{c_m(A)\}.$$  \hfill (5.9)

Unfortunately, these distributions are more similar than $P_{\text{con}}, P_{\text{incon}}$, and therefore difficult to distinguish (see Figure 6.3(a) for such a pair of distributions considered in our experiments). We shall restrict the rest of the discussion to binary multi-feature cues, the distribution of which is specified by the two error probabilities, $p_{\text{miss}}, p_{\text{fa}}$:

$$p_{\text{miss}} \overset{\text{def}}{=} P_{\text{con}}\{c_m(A) = 0\}$$
In this case, the conditional distributions \( P_0\{c_m(A)\} \), \( P_1\{c_m(A)\} \) are

\[
P_1\{c_m(A) = 1\} = p^*
\]
\[
P_1\{c_m(A) = 0\} = 1 - p^*
\]
\[
P_0\{c_m(A) = 1\} = p_{f_a}
\]
\[
P_0\{c_m(A) = 0\} = 1 - p_{f_a}
\]

where

\[
p^* = p^{(m)}(1 - p_{miss}) + (1 - p^{(m)})p_{f_a}.
\]

Note that \( p^* > p_{f_a} \). The log-likelihood ratio of the \( i \)-th randomly-selected subset, \( A_i \), becomes:

\[
h(c_m(A_i)) = \begin{cases} 
\log\left(\frac{p^*}{p_{f_a}}\right) & c_m(A_i) = 1 \\
\log\left(\frac{1 - p^*}{1 - p_{f_a}}\right) & c_m(A_i) = 0.
\end{cases}
\]

### 5.4 Optimizing the Raw Cue.

Apparently, it is not clear how grouping cues should be chosen and optimized. This is evident from the many cues used by different people to capture the same perceptual phenomenon. In particular, a cue function may include some built-in constants, like thresholds, which affect the overall balance between the number of false positive decisions and the number of misses made by the cue. In this section we focus on this aspect of cue optimization and provide a quantitative method for selecting such constants optimally in the context of the proposed cue enhancement method.

By Claim 5.2.1 we know that raw cues with various reliabilities can be enhanced to the same required reliability, \( \epsilon_{miss}, \epsilon_{fa} \), as long as the cue assumptions hold. Yet, a better raw cue requires fewer tests by the SPRT, and is thus more computationally efficient and less likely to violate the independence assumption. Therefore, we suggest using the raw cue which minimizes the expected number of SPRT trials, \( E\{n\} \). We shall use the multi-feature co-linearity cue (2.3) to illustrate this technique, and then discuss its more general applicability.

Using some rough model of the imaged scene and the underlying graph that is used, we may evaluate the a-priori probabilities for consistent and an inconsistent pairs, denoted \( Pr\{\omega_1\} \) and \( Pr\{\omega_0\} \), respectively. Then, the expected number of
The two conditional expected numbers of trials are influenced by the statistics of the cue only by \( \eta_0 \) and \( \eta_1 \). For binary raw cues these are given by

\[
\eta_k(T) = E\{h(c_m(A))|t(\epsilon) = 1\}(T) = \sum_{j=0}^{1} P_j\{c_m(A) = j\}(T) \log \frac{P_j\{c_m(A) = j\}(T)}{P_0\{c_m(A) = j\}(T)}. \tag{5.15}
\]

The two pdf-s of the cue functions are calculated as explained in appendix A, but with respect to the optimized parameter \( T \). For the co-linearity example, given in (2.3), we first use a Monte-Carlo simulation to find \( Pr\{\lambda_2(A)|t(\epsilon) = 0\} \) and \( Pr\{\lambda_2(A)|t(\epsilon) = 1\} \), which are independent of \( T \) (see Figure 6.3(a)). Then, we used these distributions to calculate

\[
P_1\{c_m(A) = 0\}(T) = \int_T^{\infty} Pr\{\lambda_2|t(\epsilon) = 1\} d\lambda_2
\]

\[
P_0\{c_m(A) = 1\}(T) = \int_0^T Pr\{\lambda_2|t(\epsilon) = 0\} d\lambda_2. \tag{5.16}
\]

Note that if \( T \) increases, then \( P_1\{c_m(A) = 0\}(T) \) decreases and \( P_0\{c_m(A) = 1\}(T) \) increases. Thus the optimal combination is found numerically at the minimum of \( E\{n\}(T) \), as shown in Figure 6.3(b).

The required reliability of the enhanced cue, \( \epsilon_{miss}, \epsilon_{fa} \), also affect the optimal \( T \) by changing \( a \) and \( b \) (5.5). Thus, the threshold value should be optimized after the required reliability of the enhanced cue was specified. (Note, however, that the statistics of the raw cue, \( Pr\{\lambda_2(A)|t(\epsilon) = 0\} \) and \( Pr\{\lambda_2(A)|t(\epsilon) = 1\} \) are independent of \( \epsilon_{miss} \) and \( \epsilon_{fa} \) and have to be evaluated only once.)

The key-point here is that the reliability of the CEP result, \( \epsilon_{miss}, \epsilon_{fa} \), is not affected by changing \( T \). Only the expected number of trials is affected. Thus, the optimization may be considered to be a maximization of the information obtained in every test (trial). This is, in fact, true also in the information theoretic sense because \( E\{n\} \) is minimized by maximizing the Kullback Leibler distance (known also as relative entropy, or divergence), between the two pdf-s, as \( \eta_0(T) = -D(P_0(T), P_1(T)) \) and \( \eta_1(T) = D(P_1(T), P_0(T)) \).

This technique is applicable for optimizing other cues' parameters, and also for choosing the best cue from some finite set of alternatives.

### 5.5 Conclusions

In this chapter we provide the cue enhancement procedure. A statistical test is used to improve the reliability of grouping cues, and to incorporate multi-feature cues.
Many grouping cues, like convexity, require more than two data features to evaluate. It is worth to note that other statistical tools were used for raw grouping cues, like the Kolmogorov-Smirnov statistics used in [GGGD90], or the feature vector used in [GS91]. However, none of them has suggested it as a general method for improving grouping cues. Rather than, they have focused on the specific measurements that are needed in a specific domain.

The CEP algorithm is efficient: for a given cue function and a constant specified reliability $\epsilon_{miss}, \epsilon_{fa}$, the expected run-time of the CEP is constant. Therefore, the expected total run-time for evaluating all the arcs of the underlying graph, $G_u$, is linear in the number of arcs.

The analysis of this method allows us to optimize the cue’s free parameters and minimize the run time, i.e., maximize the amount of information evaluated by the cue.

We emphasize that this cue enhancement method is completely general and independent of the grouping mechanism which uses it. It may use any cue that satisfies the benign assumptions stated above, and it relies only on its abstract characterization, according to the distribution.
Chapter 6
Grouping Experimentation

This section presents three different grouping applications, implemented in different domains, as instances of the generic grouping algorithm described above. To the best of our knowledge, it is the first time that a generic grouping algorithm is used in multiple domains. For each implementation, the domain, the data features, and the grouping cue are different, but the same grouping mechanism (and computer program) is used (see Table 6.1). The aim of these examples is to show that useful grouping algorithms may be obtained as instances of the generic approach, and to examine the performance predictions against experimental results.

We do not expect that our general algorithm will perform as well as domain specific algorithms which were tailored for that domain. Still, in all tested domains, we got grouping results comparable to those obtained from existing, domain specific methods. This is remarkable because, except for the choice of the cues (and the associated underlying graph determined by their extent), the process did not depend on the domain. Moreover, although some of the analysis may help in selecting between different available cues, we did not focus on choosing the best cues, but rather on testing our approach using reasonable ones. Therefore, we expect that even better performance will be possible by optimizing cues and their corresponding underlying graphs. The reported run times (5 to 15 minutes) include much self-statistics and graphics, and besides, are not the CPU time, but the overall time it took the program to run in a multi-user environment.

6.1 A Heuristic Algorithm for the MLGC Criterion

As was shown in Section 4.2, the MLGC problem in its general form is NP-H. In our experiments we have therefore used a heuristic algorithm. The grouping algorithm is composed of the three main stages (see Figure 3.1 in Chapter 3). The first stage is to construct the underlying graph $G_u$. In the second stage, the grouping cue is applied to
<table>
<thead>
<tr>
<th></th>
<th>The 1st example</th>
<th>The 2nd example</th>
<th>The 3rd example</th>
</tr>
</thead>
<tbody>
<tr>
<td>data elements</td>
<td>points in $R^2$</td>
<td>edgels</td>
<td>patches of Affine optical flow</td>
</tr>
<tr>
<td>grouping cues</td>
<td>co-linearity</td>
<td>co-circularity</td>
<td>consistency with Affine motion</td>
</tr>
<tr>
<td></td>
<td></td>
<td>and proximity</td>
<td></td>
</tr>
<tr>
<td>Cue’s extent</td>
<td>global</td>
<td>local</td>
<td>global</td>
</tr>
<tr>
<td>Enhanced cue</td>
<td>subsets of 3 points</td>
<td>subsets of 3 edgels</td>
<td></td>
</tr>
<tr>
<td>underlying graph</td>
<td>complete graph</td>
<td>locally connected graph</td>
<td>a complete graph</td>
</tr>
<tr>
<td>grouping mechanism</td>
<td>maximum likelihood graph clustering (same program)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6.1: The three instances of the generic grouping algorithm

each of the arcs of the underlying graph, and the measured graph, $G_m$, is evaluated. The last stage is the graph clustering process, which is the most computationally demanding stage. We have already shown (in Section 4.2) that graph clustering by the maximum likelihood criterion is an NP-hard problem. We have therefore developed and used a heuristic algorithm for the experimentation.

The algorithm receives the graphs, $G_u$ and $G_m$, and the error probabilities of the grouping cue, $\epsilon_{miss}, \epsilon_{fa}$. It produces a partition of the graph nodes which is a local maxima of the likelihood criterion. However, in our extensive experiments, this algorithm performs nicely, both quantitatively and visually. The same algorithm is used in all three of the implemented domains.

The graph clustering algorithm is based on finding seeds of clusters (groups), which form (almost) cliques in $G_m$, and then growing and modifying them iteratively. The modification stage is comprised of two phases: in the greedy phase these seeds are iteratively modified by making small changes (such as moving one element from one group to another, or merging two groups), using a greedy policy, until a (local) maximum of the likelihood function is obtained. In the constraint phase the result is “corrected”, by applying some constraints on it (e.g., dismissing all the small groups). This phase usually causes a temporary decrease in the likelihood. This process is iteratively repeated, until the likelihood of the result can no longer be improved. The algorithm is summarized in Figure 6.1.

Random graphs theory implies that cliques of a certain size are most likely to be found inside an object, and are very unlikely to be found elsewhere in the graph. One of Matula’s important contributions to the field of random graphs theory is the discovery of the clique number phenomenon [Pal85, Mat72] (For more about random graphs see, e.g., [Bol85]). Consider a random graph $G_{n,p}$, composed of $n$ nodes, where each of the $n(n-1)/2$ arcs is present independently with probability $p$. Let $z_{n,p}$ denote the largest clique size found in a random graph. The distribution of $z_{n,p}$
The MLGC heuristic algorithm

0. Input perceptual information \((G_u, G_m, \text{ and } \epsilon_{\text{miss}}, \epsilon_f)\).
1. Create new group seeds.
2. Evaluate the best node move.
3. Evaluate the best groups join.
4. Make a greedy modification phase.
5. If the likelihood was not improved since previous iteration, exit with the last partition.
6. Make a constraint modification phase, and return to 1.

Figure 6.1: Summary of the Maximum-Likelihood Graph-Clustering algorithm.

has a remarkably sharp value. Let \(d = d(n)\) be defined by:

\[
d(n) = 2 \log n - 2 \log \log n + 1 + 2 \log \frac{e}{2}
\]

where logarithms are to the base \(1/p\). Then the maximum clique size \(z_{n,p}\) of the random graph \(G_{n,p}\) satisfies

\[
\lim_{n \to \infty} \Pr\{|d(n) - \epsilon| \leq z_{n,p} \leq |d(n) + \epsilon|\} = 1,
\]

where \([x]\) denotes the greatest integer less than or equal to \(x\). Therefore, there is a very high chance to find a clique of a specific size \(d' < d(n)\) in the random sub-graph which represents an object (\(n\) is small, but \(p\) is high). At the same time, there is a small probability to find a \(d'\) clique in the subgraph which corresponds with the background (\(n\) is large but \(p\) is small). If one finds a \(d'\) clique, then it is a part of an object with a very high probability. Although \(z_{n,p}\) slowly converges to this asymptotic tight bound, this result is important to the understanding of the nature of random graphs, and the type of clusters which are likely to be found. The first part of our heuristic grouping algorithm, is motivated by this result.

A new seed is found by locating the pair of nodes which has the highest number of common free (ungrouped) neighbors in \(G_m\). The number of common neighbors is an upper bound on the maximal clique size, which this pair could be a part of. A similar process has been proposed by Shapiro and Haralick for decomposition of 2D shapes into simple parts [SH79].

In our experiments, described in the next section, this algorithm provides good results in all three tested grouping domains. Its main weakness is that whenever two neighboring groups are joined during the process, the algorithm is unable to separate
them. This is why we prefer to start the iterations with smaller group seeds. Typical run-times are given in Figures 6.5, 6.6, 6.7. A detailed description of the algorithm, including the data structures needed to accelerate the calculation, are included in the technical report [AL94a] \(^1\).

### 6.2 Example 1: Grouping of Points Into Co-Linear Sets

Given a set of points in the plane, the algorithm should partition the data into co-linear groups (and one background set). To remove any doubt, we do not intend to propose our grouping approach as an efficient (or even reasonable) method for detecting co-linear clusters. Several common solutions (e.g., Hough transform, RANSAC) exist for this particular task. We have chosen this example as a case-study because it is a characteristic example of grouping tasks associated with globally valid cues (and complete underlying graphs). Moreover, it provides a convenient way for measuring grouping performance, the quantification and prediction of which is our main interest here.

The co-linearity multi-feature grouping cue, defined over data subsets containing three data features, is given in detail in Section 2.2. It was used as the raw cue for the CEP. The cue is global; hence, the underlying graph is the complete graph. We consider synthetic random images containing randomly drawn points (e.g., Figure 6.2(a)). The points are drawn according to a distribution specified by a collection of arbitrary straight lines which are the “objects” associated with the given data, and some additional, uniformly distributed, aliens. With this data source, it is easy to automatically create many data sets with known noise distributions and grouping ground truth. See Figure 6.2 for a typical image and a grouping result. This image is associated with five lines, each containing 30 points in their vicinity, and 150 uniformly distributed additional data features. Unfortunately, this ground truth is not perfect, due to the (few) alien points located very close to the “objects”.

We used the co-linearity example to test the performance of the grouping algorithm against its predictions. First, we considered the cue reliability and the improvement that was achieved with the CEP. The reliability of the raw, multi-feature cue is estimated by the two cue-value distributions, \(P_0(\lambda_2), P_1(\lambda_2)\) (see (5.8)). This is done by a Monte-Carlo process over randomly-selected feature-triplets. These two distributions tend to be very similar, as shown in Figure 6.3(a). The value of the threshold \(T\) (2.3) determines the error probabilities associated with the binary cue, and thus effects the expected number of SPRT trials through \(\eta_0\) and \(\eta_1\) (5.6). (See Section 5.4, for an explicit derivation of this dependency.)

\(^1\)The TR is available at [http://www.cs.technion.ac.il/~arnon/publications.html](http://www.cs.technion.ac.il/~arnon/publications.html)
Figure 6.2: Example 1: grouping of co-linear points. Quantitative results of this experiment are shown in Figure 6.4.
As apparent from Figure 6.3(b), the expected number of trials is minimized for some threshold value, and this threshold is selected. Note that the selection of $T$ does not affect the grouping quality, but only the computational time needed for the SPRT to reach the desired error of the enhanced cue, $(\epsilon_{miss}, \epsilon_{fa})$.

As described above, the cue enhancement result depends on the computational efforts invested. The measured average number of subsets needed for the SPRT, $E\{n\}$ is given by the labels in Figure 6.4(c), for 100 different pre-specified $(\epsilon_{miss}, \epsilon_{fa})$ values, and remarkably agrees with the predicted average (5.6), shown by the labeled curves in this graph (here $T$ remains constant). It is also shown that the enhanced cue reliability can exceed 95% (i.e. $\epsilon_{miss} < 5\%$, and $\epsilon_{fa} < 5\%$), even with the simple cue we used, which has a very low discrimination power by itself.

Now we turn to the overall grouping quality. Regardless the choice of $(\epsilon_{miss}, \epsilon_{fa})$, all 5 lines were always detected as the 5 largest groups in our experiments. The selection of $(\epsilon_{miss}, \epsilon_{fa})$ does affect, however, the overall grouping quality. This is measured by counting the addition errors and the deletion errors, as shown in Figure 6.4(a) and Figure 6.4(b), respectively. Note that while the deletion error is very low, as expected, the addition error is higher than expected for groups of that size. This discrepancy is caused by some alien data features which are very close to one of the lines and are erroneously added to it. The tradeoff between grouping quality and the computational time of the CEP is obtained by these three graphs; As $E\{n\}$ increases (Figure 6.4(c)), the errors decrease (Figure 6.4(a) and Figure 6.4(b)). The highest quality is reached near the origin, where $\epsilon_{miss} = \epsilon_{fa} = 4\%$ and the average number of tested subsets is $E\{n\} = 84$. 

Technion - Computer Science Department - Ph.D. Thesis PHD-1997-01 - 1997
(a) The distribution of the co-linearity cue values, for subsets including a consistent feature pair, $P_0\{\lambda_2\}$ (solid) and for subsets including an inconsistent feature pair, $P_1\{\lambda_2\}$, (dashed). Although these two are very similar, their populations can be distinguished with less than 5% error, as shown in Figure 6.3.

(b) The expected number of trials needed for the CEP as a function of the selected cue threshold. The optimal cue threshold is $T = 0.5 \cdot 10^{-4}$, corresponding to the minima of this curve. The experimental results are shown in Figure 6.3.

Figure 6.3: Optimizing the enhanced grouping cue.
Figure 6.4: The tradeoff between the enhanced cue reliability and the computational effort invested is clearly demonstrated in this figure. Every point in each of the three graphs represents a complete grouping process. Its coordinates are the specified enhanced cue reliability measures, $E_{\text{miss}}$, $E_{\text{fa}}$, and it is labeled by the number of deleted points (deletion error) from all 5 lines (a), and added points (addition error) to all 5 lines (b). The experimental average number of trials, $E\{n\}$ is shown in (c). The solid lines show the predicted average time, for $E\{n\} = 5, 15, 25, 35, 46, 55, 65$ (the lines’ labels). It shows a good match between predicted time to experimental time.
6.3 Example 2: Grouping of Edgels Lying on Smooth Curves

Starting from an image of edgels, (data feature = edge location + gradient direction), the algorithm should group edgels which lie on the same smooth curve. This is a very useful grouping task, considered by many researchers (see, e.g., [GM92, ZMF+95, HH93, SU90, CRH93]). A crude co-circularity cue function, operating on edgel triples, is used within the CEP. It is calculated as the maximal angular difference between the gradient direction and the corresponding normal direction to the circular arc passing through the three points (see above Figure 5.1(right)). The underlying graph is locally connected and is constructed by connecting every edgel to its $K \in [10, 40]$ nearest edgels ($K$ is a constant).

We test this procedure both on synthetic and real images, and the results are very good in both cases (see Figure 6.5 and Figure 6.6). Synthetic images are created by detecting the edges of piecewise constant images which contain grey level smooth blobs (e.g., Figure 6.5(a)). Edge detection and gradient where calculated on image (a). 50% of the true edge pixels were randomly removed, and 10% of the background pixels were added, as aliens, with uniformly distributed gradient directions. Total number of edgels is about 5,000, and about 110,000 arcs in $G_u$. The processing time is about 3 minutes on a Super-Spark CPU.

In the synthetic example, we found that the boundary of each of the two big blobs splits into 3-4 groups (see Figure 6.5(e)). This happens in places where the connectivity in $G_u$ is low, the minimal connectivity assumption fails, and the split probability increases (see Figure 4.4).

The brain computerized tomography (CT) image includes many small details (Figure 6.6). The grouping result is shown in Figure 6.5(f). Most of the non-smooth edges were removed, and the remaining smooth edges are sorted into groups. The underlying graph, $G_u$, is made of 10,400 edgels and 230,000 arcs ($k = 40$). The processing time is about 10 minutes on a Super-Spark CPU.

6.3.1 Creating a Saliency Map

Saliency maps are very useful for figure ground discrimination, perceptual grouping, and other visual tasks. Using the grouping likelihood graph, the degree of a node in the measured graph, $G_m$, can serve as its likelihood; if the number of arcs connected to a node is higher then this edge point is more likely to be in a structure. In Chapter 8 we use this measure for figure ground discrimination.

A second definition of saliency is shown in Figure 6.5(d) and Figure 6.6(d). These figures show the measured graphs of the smooth curves grouping example. To create these maps we draw a straight line between every two edge points that are connected.
by an arc in $G_m$. The pixel gray level indicates the number of arcs passing thru. Note that the bright groups in the measured graph not only correspond to the local density of $G_u$, but also to smoothness.
Figure 6.5: Example 2-1: Grouping of smooth curves in a synthetic image.
CHAPTER 6: GROUPING EXPERIMENTATION

(a) Original image: A brain image.

(b) Edge detection of (a). The associated data features are edgels.

(c) Underlying graph $G_u$: locally connected (40 nearest nbrs).

(d) Measured graph $G_m$. The pixel gray level indicates the number of arcs passing thru.

(e) The five largest detected groups.

(f) All the detected groups, superimposed on the original image.

Figure 6.6: Example 2-2: Grouping of smooth curves in a brain image.
6.4 Example 3: Motion-Based Segmentation

The third grouping algorithm is based on common motion. The data features are pixel blocks, which should be grouped together if their motion obeys the same rule, that is if the given optical flow over them is consistent with one Affine motion model [JC94, AW94]. Technically, every pixel block is represented by its location and six parameters of the local Affine motion model (calculated using Least Squares). The grouping cue is defined over pairs of blocks, and its value is the sum of the optical flow errors of each block when calculated using the Affine model of the other block. The cue is global and hence a complete underlying graph is used. No cue enhancement is used here, and the cue is not very reliable: typical error probabilities are $\epsilon_{miss} = 0.35$ and $\epsilon_{fa} = 0.2$. Still, the results are comparable to those obtained by a domain specific algorithm [AW94]. The final clustering result, shown in Figure 6.7(f), was obtained after a post-processing stage: the obtained grouping is used to calculate an Affine motion model for every group, which is used to classify all the individual pixels in the image into groups. (The same post-processing method used in [AW94],) Note that the underlying graph is a complete graph of about 600 nodes (180,000 arcs), and the runtime is about 5 minutes. This result shows that even the groups (e) are not visually nice, they can still capture the correct motion clustering of the image.
CHAPTER 6. GROUPING EXPERIMENTATION

(a) Original image: Flowers sequence.

(b) Associated data features: Optical flow (blocks).

(c) Underlying graph $G_u$: A complete graph.

(d) Measured graph $G_m$.

(e) The 3 resulted groups, each in a different gray level. Black regions were either eliminated for high error with their Affine model (e.g. on the tree border), or not grouped to any of the groups.

(f) Resulted motion-based segmentation (Black pixels were not classified).

Figure 6.7: Example 3: Image segmentation into regions consistent with the same Affine motion parameters.
Chapter 7

Grouping-Based Hypothesis Verification

7.1 Introduction

Verification is the second and last part of most object recognition systems. It comes after the hypothesis drawing stage, which usually supplies several alternative hypotheses. Verification proceeds by evaluating the consistency of these candidate hypotheses with the image, giving a score as a measure of “goodness” for every hypothesis (e.g, [GH91a, GH91b, MZ92]). In this part of the work we focus on the verification stage, and suggest a new score measure, which leads to a more reliable recognition result.

Most of the research in object recognition was dedicated to the hypothesis drawing stage. Much less work has been done about the verification stage. Lowe has emphasized the importance of the verification module in his 3D model-based object recognition system (SCERPO), saying that in many ways this is the most powerful component of his system [Low85]. The verification stage in SCERPO simultaneously refines the object pose and evaluates its goodness, in order to find the best match with the image data. Lowe states that adding more information to the verification stage could greatly decrease the ambiguity in the presence of noise. He suggested to extend the verification process by testing surface properties, such as color, shading, and texture measures. Here we suggest to use the laws of organization, measured by grouping cues, to improve the verification reliability.

Usually the score is calculated by counting the number of data features that are consistent with the hypothesized object instance [HU90, Wol90, BB82]. Many algorithms build, for example, on the object boundary; in this case the score is equal to the number of edge points in the image which lie close enough (within the error tolerance) to the boundary of the projection of the hypothesized object instance. More elaborate methods rely on an error model (usually Gaussian), implying that the contribution of every data feature to the score depends on its distance from the
hypothesized boundary (see e.g. [SG93, MHL+94]). Some recognition algorithms do not contain an explicit verification stage, but still rely on a similarly calculated score. In the Hough transform [DH72, BB82], for example, the number of votes in every cell is equal to the number of features consistent with an associated hypotheses. We shall refer to all methods which evaluate some consistency measure for every data feature and calculate the score as the sum of these measures as additive score methods.

In a statistical context, the additive score approach corresponds to an implicit independence assumption on the data feature errors. Such verification procedures were analyzed theoretically, and the intuitive observation, that a higher additive score indeed results in a more reliable hypothesis, was quantified in a meaningful way [GH91b, Lin95, Lin94]. For many recognition tasks, the additive score is indeed sufficient for choosing the correct hypothesis from those suggested by the hypothesizing mechanism. However, in cluttered scenes it often happens that some large data subset is consistent with some wrong object instance hypothesis, leading to the wrong conclusion that this object is indeed in the scene (see Figure 7.1(d)). In fact, the number of data features consistent with the wrong hypothesis may even be larger than the number of data features consistent with the correct hypothesis, especially if the latter is occluded. Such failure of additive score methods in an adverse context is predicted by theory. In [GH91a, Lin94] it is shown that the minimal score threshold necessary for rejecting false hypotheses with high confidence, may be higher than the maximal score threshold required to accept the correct hypothesis with high confidence. In this case, no score level is a reliable indication of the true hypothesis, and methods based on additive score cannot succeed with the required confidence. Such failures are very common also in practical situations, and limit the reliability of recognition systems [MHL+94, Zis, Bre93]. They happen even when it is visually clear to a human observer which object instance is indeed present in the scene (see the simple examples described in Figure 7.1).

The basic theme of this work is that verification should measure the consistency of the hypothesis in the context of the whole image data, and should take into account especially its mutual dependencies. Our basic observation is that the hypothesized object instance induces a partition of the data feature set into two subsets: those data features associated with the hypothesized object and those which are not. This partition should be consistent with the natural segmentation (or grouping) of the image, which is not related to the hypothesized object and is evaluated only by general perceptual grouping cues. Therefore we intend to reject hypotheses associated with unreasonable partition, even if they are supported by many data features. We shall show that this approach yields a more reliable verification, a result consistent with the superior verification performance of human observers, which probably relies on their effective grouping abilities.

Usually, grouping is used as a low or medium level process, which effectively de-
Figure 7.1: Four hypotheses of a circle which may be associated with the same additive score. Every small line segment denotes an edge pixel, with its direction. The first case, (a), stands for a half-occluded circle, and is expected to get the highest score out of the four. In (b) it is not clear why some parts of the boundary are missing, although the hypothesis may look correct. The hypotheses in cases (c) (partially match with different object/s) and (d) (match with clutter, noise edges) are clearly incorrect.

Increases the computational costs of subsequent higher level process (e.g. object recognition [Gri90, HMS94], stereo matching [MN89], or scene interpretation [JSS+96]), and for image classification and categorization [FMF+96]. Witkin and Tenenbaum have already argued that, like the human vision system, many processing levels of a computer vision system may benefit from grouping [WT83]. Here we follow this argument and use grouping for verification, which is certainly a high level process.

To describe the grouping information available to the verification process we use the same graph representation described in Chapter 3. Here, we adhere to the stochastic approach and treat the grouping cues as random variables. We use the likelihood of the partition, induced by the hypothesis, as a measure of consistency between the image and the hypothesized object instance. This consistency measure does not depend on the number of consistent data features alone, but also on their relative locations and orientations, captured by the grouping cues. In fact, the consistency of one data feature with the hypothesized object cannot be evaluated independently of the other data features. Therefore, such a consistency measure may be considered as a non-additive score to the hypothesis.

Two other verification methods already used score measures which are not additive. Breuel uses a non-additive measure and shows that it performs better than methods that use additive score [Bre93]. He rightfully claims that the nature of edge detection processes implies that continuous edge segments are more likely than edgel segments which are scattered along the boundary. Therefore, the consistency of the former edgel subset yields a stronger evidence for the presence of the object in the scene, and should be weighted higher. In a very recent paper, Rothwell suggests using topological constraints in the verification process [Rot96]. He claims that all parts of the
hypothesized object boundary should be supported by the presence of image edgels unless some part is occluded or the hypothesis is incorrect. He discriminates between these two possibilities by trying to detect occlusion events (e.g. T-junctions). If such occlusion events are found, then the missing data is explained and the hypothesis score is increased by the number of edgels expected in the occluded part. If there is no evidence for such an occlusion, the hypothesis score is lowered.

Although the idea of using higher order statistics, suggested by Breuel [Bre93], is fairly general, both approaches (of [Bre93] and [Rot96]) base their score calculation mostly on intuition and experimentation. Therefore, although they considerably improve the verification reliability, they can serve only as a first step in using the non-additive score principle in a wider context. The framework we suggest is rigorously based on statistics. The score calculation is based on the reliability of grouping cues which is quantified by error probabilities and may be estimated analytically or experimentally. More specifically, the score is just the joint likelihood of the available grouping cues, treated as random variables. In a way, the systematic and quantitative method proposed here generalizes and unifies some of the criteria suggested by Breuel and by Rothwell. For example, it weights higher the evidence coming from a continuous set of edgels and penalizes edgel curves, which smoothly leave the hypothesized object boundary (and therefore, are not T-junction).

7.2 Grouping Cues and Their Representation

The suggested verification method is based on testing the agreement of the object instance hypothesis with the available grouping information, which consists of grouping cues. A grouping cue may provide, for example, strong evidence that two data features belong to the same object. Such evidence will work against hypotheses which associate only one of these data features with the hypothesized object. On the other hand, if the cue evidence indicates that the two data features do not belong to the same object, then it should support such hypotheses.

To formulate and to accumulate such evidence we shall recall the framework developed in Chapter 3 (see Figure 7.2). In this framework, grouping cues are considered to be random binary functions of data feature pairs (e.g. edge points). For a pair \( e = (u, v) \), the grouping cue provides a decision which may be either that the two features which form the pair belong to the same group \( (c(e) = 1) \), or that they do not \( (c(e) = 0) \). In the context of the following examples, the grouping cue is a smoothness criterion (see Figure 5.1), a function which decides whether or not two edge points belong to the same smooth curve. The cues are characterized by their reliability, independently of the domain for which they are extracted. Let \( t(e) \) denote the true association between the features of the pair \( e \). That is, \( t(e) = 1 \) if both features indeed belong to the same group, and \( t(e) = 0 \) if they do not. Using this notation, the
cues’ reliability may be quantified by the probabilities that the cues provide wrong decisions,

$$
\begin{align*}
\epsilon_{fa} & = \text{Prob}(c(e) = 1 \mid t(e) = 0) \\
\epsilon_{miss} & = \text{Prob}(c(e) = 0 \mid t(e) = 1).
\end{align*}
$$

This model is easily extendible to more general cue functions which provide a non-binary output, such as the likelihood of the feature pair to be in the same group.

![Diagram of the proposed grouping-based verification process](image)

**Figure 7.2:** The proposed grouping-based verification process: the image provides a set of data features (edgels in this illustration), every one of which is represented by a node of a graph. The first step is to decide about a cue and about the set of feature-pairs to be evaluated using this cue. This set of feature-pairs is specified by the arcs of the underlying graph $G_u = (V, E_u)$. The second step is to use the grouping cue to decide, for every feature pair in $E_u$, if both data features belong to the same group. These decisions are represented by the measured graph $G_m = (V, E_m)$: every arc corresponds to a positive decision (hence $E_m \subseteq E_u$). The known reliability of these decisions is used in the last step to evaluate the likelihood of every hypothesis.

To describe the set of feature pairs for which the cues should be calculated, we use the so called underlying graph, $G_u = (V, E_u)$, which contains a node for every feature in the image and an arc for every calculated cue. Although a cue function may operate on any feature pair in a given image, we may not wish to calculate it for every pair. In addition to the general reasons discussed earlier in Section 3.1, here we are interested in the regions of the image which are relevant for any of the given hypotheses. Non relevant regions, which are far from any hypothesis, shall not be explored at all.
The measured graph, $G_m = (V, E_m)$, describes the results of the cue calculations. It is constructed from the underlying graph, by keeping the arcs of which the corresponding cue value is “1” (i.e. $c(e) = 1$) and deleting all others (for which $c(e) = 0$).

The verification process starts by building $G_u$, then applying the grouping cue on each one of its arcs, and evaluating $G_m$. The generic grouping algorithm, proposed in Chapter 3, uses the same abstract graph structure and the available cue reliability. For the grouping task, we look for the partition of the data feature set that maximizes the joint likelihood of the cues. Here, the (computationally expensive) partition stage is not required.

### 7.3 The Grouping Based non-Additive Score

#### 7.3.1 The Hypothesis-Induced Partition of The Data

Consider an image containing $M$ data features, and some hypothesis drawing algorithm which examines it and provides several (alternative) hypotheses regarding the identity and the pose of the object in the imaged scenes. Every such hypothesis divides the data feature set into two parts, one of which is related to the object, $O$, and another of which is related to the background, $B$. The background is not necessarily uniform, and may contain objects as well. In the examples considered herein, the data features are edgels, and the object-related edgels-subset, $O$, consists of those edgels which lie close to the hypothesized object boundary.

This feature partition induces a further partition, of the feature pairs and their corresponding arcs of the underlying graph, into three disjoint subsets:

1. **Object-Object set**, $OO$: the set of arcs corresponding to two data features in the hypothesized object.

2. **Object-Background set**, $OB$: the set of arcs corresponding to one data feature in the hypothesized object and one data feature outside the hypothesized object.

3. **Background-Background set**, $BB$: the set of arcs corresponding to two data features outside the hypothesized object.

#### 7.4 The Likelihood of an Hypothesis

If the hypothesis is correct, then it is likely that cues operating on feature pairs of the $OO$ type yield a “1” decision (the feature pair belongs to the same object) more often than a “0” decision. Similarly, cues operating on feature pairs of the $OB$ type are expected to yield a “0” decision more often. The cue information (decisions) related to the third type of data feature pairs, which are included in the $BB$ set, does
not carry information about the hypothesis, as the presence of other objects is not excluded.

To use these observations, we suggest treating the grouping information supplied by the cues as instances of random variables. Then, we may measure the joint likelihood of all the cues relative to the different partitions and choose the partition which maximizes the likelihood. First, consider some particular cue which corresponds to some feature pair \( e \). If this feature pair belongs to the \( OO \) set, it means that the hypothesis places this pair of data features in the same object. In this case, the likelihood of the value “0” is \( L\{c(e) = 0|e \in OO\} = \epsilon_{\text{miss}} \) and the likelihood of the value “1” is \( L\{c(e) = 1|e \in OO\} = 1 - \epsilon_{\text{miss}} \). Similarly, if \( e \) belongs to the \( OB \) set, then \( L\{c(e) = 1|e \in OB\} = \epsilon_{fa} \) and \( L\{c(e) = 0|e \in OB\} = 1 - \epsilon_{fa} \). Calculating the likelihood of cues corresponding to feature pairs in the \( BB \) set (both features belong to the background) is not straightforward and will be considered below. For the moment, assume that the likelihood may be calculated for all cues.

Note that the likelihood of the cue result (or just the cue) depends on the hypothesis, which in turn determines the data partition and the \( OO, OB \) and \( BB \) sets. Therefore, we shall use the notation \( L\{c(e)|h\} \) for the likelihood of the cue corresponding to the edge \( e \) relative to the membership of \( e \) as determined by the hypothesis \( h \).

To calculate the joint likelihood of all the cues, we assume that the cues are independent random variables. This assumption may be challenged, but note that it is different (and weaker) from the assumption (made by all additive-score verification methods) that the location of the data features is determined by a random iid process. Here we allow, and in fact build on the dependencies between data features. We simply assume that the grouping tests (cues) fail (or succeed) independently on different pairs. Using this independence assumption, the joint likelihood of all the cues is just the product of all individual likelihoods. This joint likelihood is the consistency measure proposed in this work.

To make the expressions simpler, we use the logarithm of this likelihood as the actual score. Recall that every one of the cues corresponds to some arc, \( e \), of the underlying graph \( G_u = (V, E_u) \), this log-likelihood is

\[
\log L_h \overset{\text{def}}{=} \sum_{e \in E_u} \log L\{c(e)|h\}.
\]

The likelihood values, \( L\{c(e)|h\} \), were specified above for feature pairs (arcs) belonging to either the \( OO \) set or the \( OB \) set. Cues which are associated with \( BB \) type feature pairs do not work for or against the associated hypothesis. The background may contain only “clutter”, but may also contain objects; the hypothesis does not indicate whether some feature pair in \( BB \) belongs to the same object or not. On the other hand, ignoring these cues in the joint likelihood is not acceptable because it sets a bias on the score and makes it higher for smaller object hypotheses.
In fact, it implies that the empty hypothesis (no object, \( O = \emptyset \)) is the most likely. Therefore, the number of terms in the score calculation (7.2) is equal to the total number of cues available, \( N = |E_u| \), (independently of the hypothesis). We shall show later that this score, which takes all feature pairs into account, is in fact equivalent to a “local” score which corresponds only to object features and their direct neighbors.

Since, in general, we do not know the number of objects in the background and their sizes, it is not possible to characterize the distribution (and the associated likelihood) of these cues exactly. Still, we found it useful to characterize these cues by the probability \( \epsilon_{bg} \) that a feature pair from the background set \( BB \) gets the “1” value. Later, in Section 7.7, it is shown that \( \epsilon_{bg} \) may be used to optimize the verification score calculation, especially if we have some idea about the expected failure mechanism. Note also that \( \epsilon_{bg} \) cannot be smaller than \( \epsilon_{fa} \) and cannot be larger than \( 1 - \epsilon_{miss} \). These bounds, which correspond to the cases where the entire \( BB \) set is just “clutter” or just one object, are sufficient for proving certain theoretical properties on the log-likelihood score.

Plugging the different likelihoods for all three arc types into the log-likelihood sum of (7.2), the log-likelihood of an hypothesis \( h \) becomes

\[
\log L_h = \sum_{e \in E_u} \log L\{c(e)|h\}, \tag{7.3}
\]

where

\[
\log L\{c(e)|h\} = \begin{cases} 
\log(\epsilon_{miss}) & \text{if } e \in O O \text{ and } c(e) = 0 \\
\log(1 - \epsilon_{miss}) & \text{if } e \in O O \text{ and } c(e) = 1 \\
\log(1 - \epsilon_{fa}) & \text{if } e \in O B \text{ and } c(e) = 0 \\
\log(\epsilon_{fa}) & \text{if } e \in O B \text{ and } c(e) = 1 \\
\log(1 - \epsilon_{bg}) & \text{if } e \in B B \text{ and } c(e) = 0 \\
\log(\epsilon_{bg}) & \text{if } e \in B B \text{ and } c(e) = 1.
\end{cases} \tag{7.4}
\]

### 7.5 A More Compact Form of The Log-likelihood Score

The likelihood of every cue \( c(e) \) depends on its value and on the set to which \( e \) belongs. As apparent from (7.3), only 6 likelihood values are possible. Therefore, all arcs may be classified into 6 categories. Table 7.1 summarizes the notations for the number of arcs in these categories.

Using these notations, (7.3) becomes

\[
\log L_h = N^{O O}_0 \log(\epsilon_{miss}) + N^{O B}_0 \log(1 - \epsilon_{fa}) + N^{B B}_0 \log(1 - \epsilon_{bg}) + N^{O O}_1 \log(1 - \epsilon_{bg}) + N^{O B}_1 \log(\epsilon_{fa}) + N^{B B}_1 \log(\epsilon_{bg}). \tag{7.5}
\]

For the special empty hypothesis, which associates all the data features with the background set \( B = V \), the object set is empty, \( O = \emptyset \), \( N^{O O}_0 = N^{O O}_1 = N^{O B}_0 = \)
A MORE COMPACT FORM OF THE LOG-LIKELIHOOD SCORE

Table 7.1: The set of feature-pairs, tested by the grouping cue, is divided into $2 \times 3$ subsets: each of the feature pairs is associated with its grouping cue result, which is either “0” or “1”. Every hypothesis infers a further partition of $E_u$ into three disjoint sets, denoted as $OO$, $OB$, $BB$. Therefore, we have 6 disjoint subsets. This table specifies the notations for the number of arcs in each category.

$$N^{OB} = 0,$$ and the log-likelihood expression shrinks to

$$\log L_{\emptyset} = N^0_{BB} \log(1 - \epsilon_{bg}) + N^1_{BB} \log(\epsilon_{bg})$$

$$= N_0 \log(1 - \epsilon_{bg}) + N_1 \log(\epsilon_{bg}). \quad (7.6)$$

The log-likelihood of the empty hypothesis, $\log L_{\emptyset}$, does not depend on any other hypothesis $h$, and thus may be regarded as a constant bias level. We shall use the relative log-likelihood,

$$score(h) \overset{\text{def}}{=} \log L_h - \log L_{\emptyset}$$

$$= -N^0_{OO} \log \frac{1 - \epsilon_{bg}}{\epsilon_{miss}} + N^1_{OO} \log \frac{1 - \epsilon_{miss}}{\epsilon_{bg}}$$

$$+ N^0_{OB} \log \frac{1 - \epsilon_{fa}}{1 - \epsilon_{bg}} - N^1_{OB} \log \frac{\epsilon_{bg}}{\epsilon_{fa}}, \quad (7.7)$$

as the verification score, for evaluating any other hypothesis. This final form differs from the original log-likelihood score, $\log L_h$ (7.2), only by a constant, $\log L_{\emptyset}$. Note that this criterion is “local”, and depends only on feature pairs associated with the hypothesized object (only the $OO$ set and the $OB$ set). This score may be interpreted as the joint log likelihood ratio, comparing some hypothesis, $h$, against the empty hypothesis. In this comparison, the feature pairs corresponding to the $BB$ set get the same likelihood for both options, and are therefore absent from the score.

This form is also computationally more practical, because it avoids the need to count $N^0_{BB}$ and $N^1_{BB}$, and relies only on the much smaller sets of arcs, $OO$ and $OB$, connected to the nodes of the hypothesized object. Therefore, the grouping cues associated with the other feature pairs do not have to be evaluated at all. This dramatically decreases the computational cost, especially if only a small number of hypotheses are considered. In our implementation, where the degree of each node in the graph is bounded, the number of evaluated grouping cues for each hypothesis is therefore linearly bounded relative the hypothesis size, $|O|$.
7.6 Some Properties of The Log-Likelihood Score

Every hypothesis \( h \) may be interpreted as a modification of the empty hypothesis, made by taking a set of arcs (or cues) from the \( BB \) set and moving them either to \( OO \) or to \( OB \). To see the impact of these moves, assume that they are done sequentially in some arbitrary order. The hypothesis score starts at zero, \( \text{score}(h) = 0 \), and changes with every arc-move. If the cue gives a correct decision, then it is straightforward to show that indeed the score change is positive if and only if feature pairs associated with \( c(e) = 1 \) move to the \( OO \) set and feature pairs associated with \( c(e) = 0 \) move to the \( OB \) set. We shall show now a slightly more general result which is not concerned with the actual cue result: that the expected score change (relative to the cue distribution) is positive if and only if the corresponding feature pair moves to the true class.

We shall consider the four possible cases, associated with feature pairs that are either true (\( t(e) = 1 \)) or false (\( t(e) = 0 \)), and which move to either \( OO \) or \( OB \). The following notation from statistics will be useful: a difference between two discrete distributions, \( p(x) \) and \( q(x) \), is often quantified by the Kullback Leibler (KL) distance (known also as relative entropy, or divergence), \( D(p(x)||q(x)) = \sum_x p(x) \log(\frac{p(x)}{q(x)}) \) (see, e.g., [Cov91]). The KL distance is non-negative for any two discrete distributions, \( D(p(x)||q(x)) \geq 0 \), and equality is achieved only if the distributions are identical, \( p(x) \equiv q(x) \). The notation \( D_b(p||q) \) is used here to denote the KL distance between two binary distributions associated with success probabilities \( p \) and \( q \), respectively.

**Claim 7.6.1** If \( 0 < \epsilon_{fa} < \epsilon_{bg} < 1 - \epsilon_{miss} < 1 \) then the expected score change is positive if and only if the corresponding feature pair moves to the true class. More specifically,

1. The expected score change, associated with the move of an arc \( e \) with \( t(e) = 1 \) from \( BB \) to \( OO \), is \( E[\Delta \text{score}] = D_b(1 - \epsilon_{miss}||\epsilon_{bg}) > 0 \).
2. The expected score change, associated with the move of an arc \( e \) with \( t(e) = 0 \) from \( BB \) to \( OO \), is \( E[\Delta \text{score}] < -D_b(\epsilon_{bg}||1 - \epsilon_{miss}) < 0 \).
3. The expected score change, associated with the move of an arc \( e \) with \( t(e) = 1 \) from \( BB \) to \( OB \), is \( E[\Delta \text{score}] < -D_b(\epsilon_{bg}||\epsilon_{fa}) < 0 \).
4. The expected score change, associated with the move of an arc \( e \) with \( t(e) = 0 \) from \( BB \) to \( OB \), is \( E[\Delta \text{score}] = D_b(\epsilon_{fa}||\epsilon_{bg}) > 0 \).

**Proof:** The conditions specified by the claim imply that

\[
\log(\frac{\epsilon_{fa}}{\epsilon_{bg}}) < 0 \quad \log(\frac{\epsilon_{miss}}{1 - \epsilon_{bg}}) < 0 \quad \log(\frac{1 - \epsilon_{fa}}{1 - \epsilon_{bg}}) > 0. \tag{7.8}
\]

Consider the first case. Since the feature pair is indeed included in the object (\( t(e) = 1 \)), then the cue value is “1” with probability \( 1 - \epsilon_{miss} \). In this case, it increments \( N_{tOO}^{e} \).
and increases the score by \( \log \frac{1-\epsilon_{\text{miss}}}{\epsilon_{bg}} \). Similarly, with probability \( \epsilon_{\text{miss}} \), it increments \( N_{00} \) and increases the score by \( \log \frac{\epsilon_{\text{miss}}}{1-\epsilon_{bg}} \). Therefore,

\[
E\{\Delta \text{score}_{1}\} = (1 - \epsilon_{\text{miss}}) \log \left( \frac{1 - \epsilon_{\text{miss}}}{\epsilon_{bg}} \right) + \epsilon_{\text{miss}} \log \left( \frac{\epsilon_{\text{miss}}}{1 - \epsilon_{bg}} \right) = D_{b}(1 - \epsilon_{\text{miss}}|\epsilon_{bg}) > 0.
\]

(7.9)

The other parts of the claim are similarly proved.

The above claim is concerned with the expected score gain of arc-moves (from say the \( BB \) set to the \( OO \) set) relative to the distribution of the cues. It shows that the correct moves (cases 1,4), as for example the move of an arc \( e \), belonging to a true object (i.e \( t(e) = 1 \)) from \( BB \) to \( OO \), increase the expected score. Similarly, it shows that incorrect moves decrease the expected score (cases 2,3). Note that in all cases the membership of the feature pair \((t(e))\) is unknown to the verification procedure. Yet, the claim implies that the expected score change for every one of them is in the correct direction. It also provides some lower bounds on the amount of score change, which relates to the amount of information provided by this arc-move. Note also that larger hypotheses are likely to get a higher score than smaller ones, if both are equally “good”. In particular the claim implies the following simple corollary:

**Corollary 7.6.2** If there is only one object instance in the image then the corresponding hypothesis is associated with the maximal expected score.

### 7.7 The Possible Failure Modes and Their Relation to The \( \epsilon_{bg} \) Parameter

It turns out that the \( \epsilon_{bg} \) parameter, used to characterize the unclassified feature pairs, has a lot of influence of the resulting score. Recall that \( \epsilon_{bg} \) is bounded by \( \epsilon_{fa} < \epsilon_{bg} < 1 - \epsilon_{miss} \). Note that if \( \epsilon_{bg} = \epsilon_{fa} \), then the last two terms of the score, (7.7), corresponding to the \( OB \) set, vanish, and the score depends only on the object-object arcs in \( OO \). Similarly, if \( \epsilon_{bg} = 1 - \epsilon_{miss} \), then the first two terms, corresponding to the \( OO \) set, vanish, and the score depends only on the object-background arcs. Therefore, decreasing \( \epsilon_{bg} \) is equivalent to desensitizing the likelihood criterion of the arcs between the hypothesized object and its background, and to increase its sensitivity to the \( OO \) arcs, and vice versa. Therefore, specifying \( \epsilon_{bg} \) appropriately helps to tune the score function, enabling to better discriminate between true hypotheses and wrong ones.

Consider the following two types of wrong hypotheses and corresponding failure mechanisms:

1. A collection of objects with smooth boundaries may be accidentally placed, so that large parts of their boundaries coincide with the boundary of an instance of the object we are looking for (See Figure 7.1(c), and Figure 7.4). In such a
case, it is expected that most of the $OO$ arcs will be associated with a smooth curve, and that a smoothness detecting cue will specify them as belonging to the same object ($c(e) = 1$). In this situation, the discrimination between such a wrong hypothesis and the true one should be based on the presence of too many $OB$ arcs for which the cue is “1”. Therefore, $\epsilon_{bg}$ should be high in order to reject such wrong hypotheses.

2. Image clutter, after edge detection, may appear as a large collection of randomly placed points or short lines, and may accidentally give false evidence for the presence of some object instance (See Figure 7.1(d), and Figure 7.3). In such a case, the discrimination between such a wrong hypothesis and the true one should be based on the presence of too few $OO$ arcs for which the cue is “1”. Therefore, $\epsilon_{bg}$ should be low.

Therefore, we recommend to tune $\epsilon_{bg}$ according to the knowledge available on the type of expected failure, which depends on the scene. However, we are able to show, in the next section, that the grouping based score gives the correct result, even if the $\epsilon_{bg}$ parameter is not finely tuned.

### 7.8 Experiments

Our experiments focus on the verification stage of the object recognition process. We are not concerned with the need to draw the hypotheses efficiently. Therefore, we can use a simple Hough transform, parameterized in translation and in scale (3 parameters), to draw hypotheses of 2D shapes. In our experimentation, it is assumed that the projected boundary of the object forms a smooth curve in the image plane. The data features are edgels (edge points with their gradient direction). We used a grouping-cue function which tests for the smoothness of data feature pairs, using some measures for co-circularity, curvature and proximity, and the cue enhancement algorithm (technical details can be found in [AL94a]). We expect similar results if other implementations of the cue are used, provided that they share the same statistical characterization, $(\epsilon_{f_s}, \epsilon_{miss})$. Gradient direction was measured by applying the Sobel operators on the given gray level images.

Our implementation of the Hough transform roughly counts the number of edge points which are located in the $\epsilon$-vicinity of the projected object boundary. We refer to this $\epsilon$-dilated boundary as the “extended boundary”. This number, normalized by the shape perimeter, is the additive score to which we compare our grouping-based score.

The first two experiments (see Figure 7.3 and Figure 7.4) use synthetic data and were constructed in order to demonstrate the two failure mechanisms described above.
In the first experiment, the object is a small 2D pigeon, partially occluded, and hidden in a lot of clutter. The edge image (a), of size $438 \times 337$, consists of a small, partially occluded, pigeon 2D shape, (like the one in (f)), and high background noise. About half of the object’s edge points had been randomly removed, to make the verification task harder. A few of the wrong hypotheses (d), and the correct one (e), are presented by their associated edge points, superimposed on the original image. Five of the wrong hypotheses, numbered 1 – 5, got an additive score, higher than the correct one (the 6th hypothesis)(g). Such a failure of the additive score may often happened in the presence of occlusion and the high clutter also in real scenes. The grouping information, represented by the underlying graph (b), and the measured graph (c), emphasize the smooth edges. The grouping-based score for the same 45 hypotheses is given (in the same order) in (h), for $\epsilon_{bg} = 0.3$. The correct hypothesis (no. 6), got the highest score, which is clearly much higher than all others. In this context the erroneous hypotheses are created by a random accumulation of clutter, as illustrated in (f), which may be discriminated from the correct hypothesis mainly by the negative contribution of the $OO_{b}$ term to the score ((7.7)). The grouping-based score depends on $\epsilon_{bg}$, as shown in (i). The correct hypothesis is associated with the easily identified distinguished curve.

In the second experiment (Figure 7.4), the object is a circle, which is half occluded. Another object in the image, a hexagon, is similar to this circle in the sense that many parts of the hexagon are included in the extended boundary of a circle. Therefore, a circle hypothesis which roughly covers the hexagon (d), gets a large additive score relative to the correct hypothesis (e). A graph of the scores of the first 25 best hypotheses, according to the additive score, is shown in (g). With the grouping based score, however, the features consistent with this false hypothesis have many neighboring features which, though not consistent with the hypothesis, are strongly grouped with the consistent features. This is apparent from the grouping information, represented by the underlying graph (b), and the measured graph (c), which emphasizes the smooth edges. This inconsistency between the hypothesis and the grouping information rightfully lowers the non-additive score, which is now largest for the true hypothesis. The grouping-based score for the same hypotheses is given (in the same order) in (h), for $\epsilon_{bg} = 0.35$. This time, the correct hypothesis (the occluded circle) got the highest score. The hexagon is penalized by the large number of $OB_{1}$ arcs it has in $G_{m}$, which is a result of the lines tangential to the hypothesis boundary, as illustrated in (f). As $\epsilon_{bg}$ increases, as shown in (i), the $OB_{1}$ arcs become more dominant in the score, and the correct hypothesis remains the highest one.

The score of the correct hypothesis in the second example is not as high as the score associated with the correct hypothesis in the previous example. This is because it also includes a large number of noise points, which decrease its score.

The third experiment (see Figure 7.5) shows that the hypotheses preferred by the
CHAPTER 7: GROUPING-BASED HYPOTHESIS VERIFICATION

Figure 7.3: Example 1: The image consists of a small, partially occluded, pigeon 2D shape, (like the one in (f)), and high background noise. Many large subsets of the clutter are consistent with wrong instances of the pigeon and thus get high additive scores. The non-additive score for these groups is low, however, because there is no natural grouping between them.
Figure 7.4: Example 2: The data edge image, (a), consists of an occluded circle, an hexagon, and some background noise. The model is a circle. The additive score of the wrong hypothesis, (d), is higher than of the correct hypothesis (e). The grouping based score discriminates the correct hypothesis from all the wrong ones.
grouping-based score are closer to those preferred by intuition. Here, we consider a realistic medical image (CT) and look for a circle, although we do not expect a perfect circle to be found in the image. The proposed score, which prefers shapes having smooth edges, is higher for the more or less circular shapes. The input is a real CT image, and its edges are shown here in (a). The grouping information, represented by the underlying graph (c), and the measured graph (d), emphasize the smooth edges. An exhaustive search in a 3D parametric space was used to generate 20 hypotheses of circles, in some limited radius range (which causes it to miss some smaller circles). Five of the hypotheses, labeled 1, 2, 3, 4, 10, by their additive score order, are given in (b, c, d, e, f). The additive scores of the first 20 best hypotheses are all in the range of 7.8 – 9.21, as shown in a form of a graph (g). The first and third hypotheses, which do not make any sense, resulted from the high density of cluttered edges at the bottom of the image. The intuitively correct hypotheses, which are more or less circular, are the 2nd, 4th and the 10th, which indeed got the highest grouping-based scores, as shown in (h) (here $\epsilon_{bg} = 0.4$). These three hypotheses correspond to the three upper curves in (i). Although there is no true circle in the image at all, one can still learn from this example that the hypotheses preferred by the grouping-based score are intuitively better than those which got higher additive scores.

The last experiment (see Figure 7.6 and Table Table 7.8) demonstrates the utility of the method for 3D realistic objects, an apple in this case, in a very cluttered image. The object’s boundary is modeled by a simple hand-drawn 2D spline. The input is a real image (a), and its edges are shown in (b). The object is an apple, its boundary modeled by a simple hand-drawn 2D spline. An exhaustive search in a 3D parametric space (translation and scale) is used to derive the 45 best (additive-score) hypotheses. A few of the wrong hypotheses, (e), and the correct one, (f), are presented by their associated edge points, superimposed on the original image. 31 of the wrong hypotheses, numbered 1 – 31, got a higher additive score than the correct one (the 32nd hypothesis), as shown by the graph (g) (see also Table 7.8). The grouping information, represented by the underlying graph (c), and the measured graph (d), emphasize the smooth edges. The grouping-based score for the same 45 hypotheses is given (in the same order) in (h) ($\epsilon_{bg} = 0.3$). The correct hypothesis (no. 32), got the highest score, which is clearly much higher than all others. In this real situation, the erroneous hypotheses are created by a random accumulation of clutter in some regions of the edge image. This is a combination of the two failure mechanisms presented in Figures 7.3, 7.4. Still, the incorrect hypotheses are rejected, mainly because of the negative contribution of the $OO_0$ term to the score ((7.7)). The grouping-based score depends on $\epsilon_{bg}$, as shown in (i). The correct hypothesis is associated with the easily identified upper curve.

The verification process is concerned with comparing the 2D projection of the hypothesized object on the image with the data available from the image. It is
essentially the same process for both 2D and 3D objects. Therefore, we argue that the results of our experimentation, done mainly with 2D objects, should apply to both domains.

It is worth noting that in the case of less complex scenes, both the additive score and the grouping based score perform very well, and the correct hypotheses usually got the highest score. Therefore, the synthetic scenes in our experiments are designed to provide difficult tasks, by occluding the real object, randomly removing half of its edge points, adding other objects, and adding many edge points to the background (about 10% of the background pixels).

Note that for all experiments we calculated the grouping-based score as a function of the parameter $\epsilon_{bg}$. In all cases the proposed score preferred the correct hypothesis over the incorrect hypotheses for a wide range of this parameter, showing the method’s relative insensitivity to it. Still, some knowledge about the expected failure mechanism helps in setting this parameter and yielding a more robust score.
CHAPTER 7. GROUPING-BASED HYPOTHESIS VERIFICATION

Figure 7.5: Example 3: The input is a real CT image. Although there is no true circle in the image at all, one can still learn from this example that the hypotheses preferred by the grouping-based score are intuitively better than those which got higher additive score.
a. The data image.  
b. The edges image.  
c. Underlying graph $G_u$. 
d. Measured graph $G_m$.  
e. A few wrong hypotheses.  
f. Hypothesis No. 32.  
g. Additive score for hypotheses 1 – 45.  
h. Grouping-based score for hypotheses 1 – 45.  
i. All 45 grouping-based scores as functions of $\epsilon_{bg}$.

Figure 7.6: Example 4: The input is a real image (a), and its edges are shown in (b). The object is an apple, its boundary modeled by a simple hand-drawn 2D spline. 31 of the wrong hypotheses, numbered 1 – 31, got a higher additive score than the correct one (the 32nd hypothesis) (g). The grouping-based score for the same 45 hypotheses is shown in (h). The correct hypothesis (no. 32), got the highest score, which is clearly much higher than all others.
Table 7.2: This table refers to the hypotheses of apples in the real image given in Figure 7.6. The hypotheses are sorted according to their additive score, which is compared to the grouping-based score. The additive score is evaluated from the number of consistent edge points, $|O|$, divided by the perimeter of the hypothesized boundary. In this example, the additive score does not match our expectations and does not give the highest score to the correct hypothesis (no. 32). However, the grouping-based score does discriminate between the correct hypothesis and the wrong ones. Note that hypotheses which have a negative score, $score(h) = log L_h - log L_\emptyset < 0$, are less likely than the empty hypothesis. This practically suggests to rejecting such hypotheses.

<table>
<thead>
<tr>
<th>Additive score order</th>
<th>Additive score</th>
<th>consistent edge points</th>
<th>Grouping-based score $score(h)$</th>
<th>Grouping-based order</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9.75</td>
<td>269</td>
<td>-493.402</td>
<td>18</td>
</tr>
<tr>
<td>2</td>
<td>9.61</td>
<td>223</td>
<td>-586.124</td>
<td>27</td>
</tr>
<tr>
<td>3</td>
<td>9.50</td>
<td>251</td>
<td>-544.752</td>
<td>25</td>
</tr>
<tr>
<td>4</td>
<td>9.30</td>
<td>186</td>
<td>-463.470</td>
<td>14</td>
</tr>
<tr>
<td>5</td>
<td>9.30</td>
<td>186</td>
<td>-500.928</td>
<td>20</td>
</tr>
<tr>
<td>6</td>
<td>9.25</td>
<td>185</td>
<td>-600.557</td>
<td>29</td>
</tr>
<tr>
<td>7</td>
<td>9.24</td>
<td>244</td>
<td>-617.642</td>
<td>31</td>
</tr>
<tr>
<td>8</td>
<td>9.15</td>
<td>227</td>
<td>-610.690</td>
<td>30</td>
</tr>
<tr>
<td>9</td>
<td>9.07</td>
<td>263</td>
<td>-835.062</td>
<td>42</td>
</tr>
<tr>
<td>10</td>
<td>9.05</td>
<td>358</td>
<td>-854.814</td>
<td>45</td>
</tr>
<tr>
<td>11</td>
<td>9.00</td>
<td>180</td>
<td>-434.750</td>
<td>13</td>
</tr>
<tr>
<td>12</td>
<td>8.88</td>
<td>263</td>
<td>-488.392</td>
<td>16</td>
</tr>
<tr>
<td>13</td>
<td>8.85</td>
<td>177</td>
<td>-382.588</td>
<td>9</td>
</tr>
<tr>
<td>14</td>
<td>8.85</td>
<td>177</td>
<td>-575.962</td>
<td>26</td>
</tr>
<tr>
<td>15</td>
<td>8.75</td>
<td>175</td>
<td>-428.885</td>
<td>12</td>
</tr>
<tr>
<td>16</td>
<td>8.70</td>
<td>188</td>
<td>-272.744</td>
<td>3</td>
</tr>
<tr>
<td>17</td>
<td>8.70</td>
<td>174</td>
<td>-400.383</td>
<td>17</td>
</tr>
<tr>
<td>18</td>
<td>8.69</td>
<td>209</td>
<td>-659.095</td>
<td>35</td>
</tr>
<tr>
<td>19</td>
<td>8.60</td>
<td>172</td>
<td>-378.243</td>
<td>6</td>
</tr>
<tr>
<td>20</td>
<td>8.57</td>
<td>199</td>
<td>-643.697</td>
<td>34</td>
</tr>
<tr>
<td>21</td>
<td>8.56</td>
<td>185</td>
<td>-320.489</td>
<td>4</td>
</tr>
<tr>
<td>22</td>
<td>8.50</td>
<td>170</td>
<td>-411.086</td>
<td>11</td>
</tr>
<tr>
<td>23</td>
<td>8.50</td>
<td>170</td>
<td>-233.257</td>
<td>2</td>
</tr>
<tr>
<td>24</td>
<td>8.44</td>
<td>223</td>
<td>-694.588</td>
<td>39</td>
</tr>
<tr>
<td>25</td>
<td>8.42</td>
<td>209</td>
<td>-516.099</td>
<td>21</td>
</tr>
<tr>
<td>26</td>
<td>8.41</td>
<td>249</td>
<td>-661.987</td>
<td>36</td>
</tr>
<tr>
<td>27</td>
<td>8.40</td>
<td>222</td>
<td>-356.816</td>
<td>5</td>
</tr>
<tr>
<td>28</td>
<td>8.31</td>
<td>246</td>
<td>-838.824</td>
<td>43</td>
</tr>
<tr>
<td>29</td>
<td>8.31</td>
<td>246</td>
<td>-519.697</td>
<td>23</td>
</tr>
<tr>
<td>30</td>
<td>8.29</td>
<td>219</td>
<td>-535.638</td>
<td>24</td>
</tr>
<tr>
<td>31</td>
<td>8.29</td>
<td>272</td>
<td>-472.900</td>
<td>15</td>
</tr>
<tr>
<td>32</td>
<td><strong>8.28</strong></td>
<td><strong>285</strong></td>
<td><strong>32.549</strong></td>
<td><strong>1</strong></td>
</tr>
<tr>
<td>33</td>
<td>8.27</td>
<td>245</td>
<td>-717.771</td>
<td>40</td>
</tr>
<tr>
<td>34</td>
<td>8.27</td>
<td>245</td>
<td>-670.973</td>
<td>38</td>
</tr>
<tr>
<td>35</td>
<td>8.26</td>
<td>205</td>
<td>-374.395</td>
<td>8</td>
</tr>
<tr>
<td>36</td>
<td>8.25</td>
<td>218</td>
<td>-516.811</td>
<td>22</td>
</tr>
<tr>
<td>37</td>
<td>8.25</td>
<td>284</td>
<td>-843.353</td>
<td>44</td>
</tr>
<tr>
<td>38</td>
<td>8.25</td>
<td>284</td>
<td>-635.004</td>
<td>32</td>
</tr>
<tr>
<td>39</td>
<td>8.24</td>
<td>178</td>
<td>-400.930</td>
<td>10</td>
</tr>
<tr>
<td>40</td>
<td>8.23</td>
<td>270</td>
<td>-631.626</td>
<td>33</td>
</tr>
<tr>
<td>41</td>
<td>8.20</td>
<td>164</td>
<td>-498.146</td>
<td>19</td>
</tr>
<tr>
<td>42</td>
<td>8.18</td>
<td>321</td>
<td>-786.062</td>
<td>41</td>
</tr>
<tr>
<td>43</td>
<td>8.18</td>
<td>216</td>
<td>-588.243</td>
<td>28</td>
</tr>
<tr>
<td>44</td>
<td>8.17</td>
<td>258</td>
<td>-363.501</td>
<td>7</td>
</tr>
<tr>
<td>45</td>
<td>8.17</td>
<td>308</td>
<td>-662.709</td>
<td>37</td>
</tr>
</tbody>
</table>
7.9 Conclusions

We proposed here a new verification paradigm, which does not rely on the size of the consistent features set. It relies, rather, on grouping information, which is extracted from the image independently of the object hypothesizing mechanism. The method is controlled by one parameter, which should be selected based on the expected error mechanism. We proved that this method is consistent, in the statistical sense; that is, in the presence of only one object in the scene, the correct hypothesis is expected to get the highest score.

A compact form of the grouping-based score function is suggested, which is also more intuitive. This score may be interpreted as the joint log likelihood ratio, comparing the tested hypothesis against the empty hypothesis. This criterion is “local”, and depends only on data features associated with the hypothesized object, and their direct neighbors. In addition, it provides an efficient way to calculate the score, in linear time (with respect to the number of data features associated with the hypothesis). This avoids the need to compute the grouping cues for the entire graph.

Our experimentation shows that while additive approaches (based on the consistent set size) for verification work well for easy tasks, our approach works better in many difficult cases. The integration of the proposed grouping based verification criterion with the more traditional consistent set size criterion, is currently under study.

One may argue that the additive score to which we compared our results is not the best one, and that Gaussian error based scores [SG93], or some score that also takes into account the edge direction, is more powerful in rejecting the incorrect hypotheses. Indeed, better additive scores can be probably constructed. An additive score that uses gradient direction depends on the neighborhood of the edge points. This implies an implicit assumption about the dependency between edge points, and may also be considered as a (simple) grouping based score. However, such methods may not be effective enough to reject all false hypotheses. Note for example, that adding the direction information would not help much when a similar object is present (as in the example shown in Figure 7.4), or in the presence of partial occlusion (see in [Rot96]). We claim here that any verification process must take into account not only the consistency of data features along the hypothesized boundary, but also their “inconsistency” with its larger neighborhood. In terms of perceptual grouping, a good hypothesis should form a “good group”. Therefore, any score method may benefit substantially from using perceptual grouping information. The non-additive score proposed here shows a significant improvement relative to the use of a simple additive score. It should improve other methods, especially when difficult verification scenes are considered.

The method can be applied, in principle, to many perceptual grouping cues. This
method does not depend on any specific grouping cue, and allows the use of any cue, as long as we know its statistics. Other grouping cues may measure symmetry properties, color, texture, motion parameters etc.
Chapter 8

Ground From Figure Discrimination

8.1 Introduction

Visual processes deal with analyzing images and extracting information from them. One reason these processes are so difficult is that the useful information is contained in only a few subsets of the data features. The additional data features, which are not relevant for the task, only serve to slow the process down and decrease its accuracy, it harder, slower and less accurate. One way to address this problem is to use a grouping algorithm. However, most grouping algorithms, including the one proposed in Chapter 3, are rather complicated, requiring high computational effort. The figure-ground discrimination process, which splits the data features into only two sets: the figure set and the background set, provides an alternative to grouping algorithms. As the task involved is simpler, it is likely to have simpler and more efficient algorithms than grouping. This chapter proposes a new, efficient, figure from ground method (first reported in [AL97a]). The proposed method can be classified as an asymmetric relaxation labeling process. At every stage, the data features are classified into either “background” or “still unknown” classes, thus emphasizing the background detection task, as implied in the chapter title. A bootstrap mechanism improves performance in very cluttered scenes. A fast implementation using a kd-tree allows work on large, realistic images.

The human visual system has a powerful figure-ground discrimination ability, first demonstrated by the Gestalt psychologists in the early twenties (see Section 1.1). Gestaltists have searched for the geometric properties and rules that are used by the human vision system for this task. The definition of a figure can take many forms, depending on the visual task. Gordon summarizes some work in this subject in his book [Gor89]. In his words, “figure tends to be complete, coherent and in front of ground, which is seen as less distinct, is attended to less readily, and is often seen as
floating behind the figure”. The human visual system uses completeness, coherence, depth, motion and many other grouping cues to segment the figure out of the received image. Current computer vision systems try to mimic this ability, but usually limit themselves to particular domains.

In the context of this work, the goal is to improve the result of edge detectors by pruning some of the detected edgels. An illustration is shown in Figure 8.1: the data features are edgels (i.e. edge points and their associated gradient directions). The figure set includes all the edgels which lie on smooth curves, and the background set includes the other, irrelevant edge points, which are associated with clutter, texture and noise. This filtering process is useful for many computer vision tasks. These visual processes, like object recognition, can focus on the figure set and ignore the background (or save it for later use). This way, the number of data features to be processed is reduced, saving computational time and space. It also improves the signal to noise ratio of the data, calling for better, more accurate, results (in contrary to random sub-sampling, sometimes used for similar reasons, e.g., with the Hough transform [SYK96], but with no effect on the signal to noise ratio). Figure ground discrimination is, therefore, a useful pre-processing filtering stage for many computer vision applications [Gri90, SB93, DI96].

![Figure 8.1: A figure ground discrimination process splits the data set (left) into a figure set (middle) and a background set (right).](image)

### 8.1.1 Related Work

Herault and Horaud have used simulated annealing for figure ground discrimination [HH93]. Their interpretation of the binary relations between data features, represented by probabilities on the graph arcs, is inspired by the spin model in particle physics. They do not use the notion of a graph, but an “interacting spin” system, described by the \( n \times n \) adjacency matrix (\( n \) is the number of data features). They consider combinatorial optimization problems, where the cost function is analogous.
to a global energy, and compare a few different annealing methods, such as simulated annealing and mean field annealing. However, simulated annealing is a relatively slow process.

The detection of saliency is closely related to figure ground discrimination. A saliency map assigns a saliency measure to every pixel in the image. The figure points are expected to have higher saliency, and could be classified this way. Relevant work that falls into this category is that of Shashua and Ullman [SU88], Guy and Medioni [GM92]. In some cases the saliency map is a 3D function of pose and orientation (e.g., [WJ95]). It has an advantage for ambiguous points, such as junctions and corners, but it requires larger memory and computational effort. However, Hough-like methods, like [GM92], are expected to suffer from the same high accumulated bias problems that appears with the Hough transform. We have not seen such saliency maps for large images that contain a lot of clutter.

Gutfinger and Sklansky have developed a classification method which combines supervised and unsupervised learning [GS91]. They demonstrate its applicability to separate the figure points from synthetic sets of points which include structured points and scattered noise points. Each point is associated with a 12-dimensional feature vector, which includes various statistics of the rest of the points. The classification takes place in this high-dimensional feature space, using an iterative algorithm. The evaluation of these feature vectors for all points requires great computational effort.

Figure ground discrimination is related to, but different from perceptual grouping. While the first process needs only to distinguish the data features that belong to some structure from those which do not, a grouping process provides a finer partitioning of the figure set into (disjoint or overlapping [Jac88, HMS96]) groups, each corresponding to a single object in the scene. In Figure 8.1(middle), for example, each smooth curve would form a group. As the figure ground discrimination is a simpler task than grouping, one could expect to solve it by simpler and faster algorithms. This simplification, however, is not apparent in previous work. Both problems are found to be difficult, and the methods proposed in previous work are often based on the same techniques (e.g., the use of saliency maps, global optimization methods, e.g., simulated annealing, and relaxation labeling. See [SB93] for a good review). The figure ground discrimination algorithm proposed here uses the same data structure as we used in [AL94b] for grouping, but is much simpler and faster.

Our starting point is the following observation. We argue that the figure ground discrimination problem is asymmetric. A data feature which belongs to the figure set must be a part of some structure. In order to verify this, we need to find this structure, either explicitly or implicitly. On the other hand, a data feature that belongs to the background set does not belong to any structure. We argue that it is easier to assert that a background data feature belongs to no structure at all than to find the structure to which a figure data feature does belong. Therefore, we propose
to look only for background data features, and to move them into the background set, until no more background data features can be found. The rest of the data features will form the figure set.

The proposed method can be classified as an asymmetric relaxation labeling process. Parent and Zucker [PZ89], and also Deng and Iyengar [DI96], have used relaxation labeling to improve edge detection results. Each pixel can take one of two labels — figure or background. After an initial labeling assignment is made (for example, by using an edge detector), the labels are updated iteratively, until convergence. The updating rule, applied separately to each pixel, is a function of the neighbors' configuration - they can either support its labeling or call for a change. One of the main questions in relaxation labeling is the convergence issue. Not only should convergence be guaranteed, but the number of iterations should be bounded as well. In some cases this might be exponential, at least from a theoretical point of view [DI96]. The algorithm is a limited relaxation labeling technique - it allows only one type of changes: a figure label can be changed to a background label, but not vice versa. This ensures not only the convergence of our algorithm, but also a linear number of iterations, in the worst case. It requires, however, that the initial labeling algorithm, that is, the edge detector, should preferably be tuned to miss less true edge points with the expense of producing more false alarms.

This approach is different from other relaxation labeling techniques in another crucial way. Usually, the neighboring system is selected according to the image grid - for example, a $3 \times 3$ or $5 \times 5$ neighborhood. It is assumed that the global structure should be inferred by the propagation of information inside the graph. However, in the absence of an appropriate propagating quality measure that accumulates along the path (e.g., the one proposed in [SU88]), there is no way to distinguish between a pixel that lies on a long boundary curve and a pixel that lies on a very short edge, only few pixels longer than the neighborhood size. This is apparent in the results reported in [DI96]. We use a different, non regular neighboring system, which is reflected in a much larger effective neighborhood in the image.

### 8.2 Grouping Cues and Their Graph Representation.

The presence of a local structure in the image can be characterized by local binary relationships between pairs (or larger subsets) of data features, even without finding and explicitly isolating the structure from its background. The mutual dependencies between data features are measured by perceptual grouping cues.

We apply here the framework developed in Chapter 3. Recalling the main notations, the grouping information available to the process is represented by the grouping likelihood graph. The grouping cues are considered to be random binary functions of
data feature pairs (e.g. pairs of edge points). For a pair of edgels, \( e = (u, v) \), the grouping cue \( c(e) \) provides a decision which may be either that the two edgels belong to the same group \( (c(e) = 1) \) or that they do not \( (c(e) = 0) \). Similarly, \( t(e) \) denotes the true association between the data features \( u \) and \( v \). That is, if both features indeed belong to the same group, then \( t(e) = 1 \), otherwise \( t(e) = 0 \). Using this notation, the cues’ reliability may be quantified by the two probabilities that the cues provide wrong decisions,

\[
\epsilon_{fa} = \text{Prob}\{c(e) = 1 \mid t(e) = 0\} \quad \epsilon_{miss} = \text{Prob}\{c(e) = 0 \mid t(e) = 1\}. \tag{8.1}
\]

As we have shown in Chapter 5, this model is easily extendible to more general cue functions which provide a non-binary output, such as the likelihood of the feature pair to be in the same group, and to cues which operate on three or more data features. In our experimentation, the grouping cue is a smoothness criterion; a function which decides whether or not two edgels belong to the same smooth curve. This particular grouping cue is described in Section 8.3.2. The reliability of grouping cues, quantified by \( \epsilon_{miss}, \epsilon_{fa} \), may be estimated analytically or experimentally.

We use the grouping likelihood graph (first introduced in Chapter 3) to represent the set of tested edgel pairs in the image, and the cue results. The underlying graph, \( G_u = (V, E_u) \), represents the set of edgel pairs to be tested by the cue. Each node represents an edgel, and each arc represents a tested pair. The measured graph, \( G_m = (V, E_m) \), represents the cue results. It has the same nodes set as \( G_u \), but only the subset of its arcs for which the cue result is \( c(e) = 1 \). For the included examples, \( G_u \) is constructed by connecting each edgel to its \( k \) nearest edgels (that is, it ignores the non-edge pixels). Intuitively, nodes that correspond to edgels which are lying on a smooth curve are expected to have many neighbors in \( G_m \), while nodes which correspond to clutter, or non-smooth edges, are expected to have very few neighbors in \( G_m \). Using this framework, the ground from figure discrimination is a node pruning process.

### 8.3 The Algorithm

This section is divided into three parts. The first describes the graph construction and the associated grouping cue, and the second deals with the iterative nodes pruning process. The third part is concerned with the overall complexity and the run time of this algorithm.

#### 8.3.1 A Fast Graph Construction

The underlying graph is constructed by connecting every edge point to its \( k \) nearest neighbors (typical values are between \( k = 10 \) to \( k = 30 \)). We use the algorithm
of Fridman, Bentley and Finkel, that uses a kd-tree to efficiently find the \( k \) nearest neighbors to a query point [FBF77]. Building the tree takes \( O(|V| \log |V|) \). Finding the \( k \) nearest neighbors to a query point takes \( O(k + \log |V|) \) (average time), or \( O(|V|(k + \log |V|)) \) for the entire graph. Note that the minimal degree in the graph is \( k \), but some nodes are likely to have higher degree, as all arcs are duplicated to create an undirected graph.

### 8.3.2 The Smoothness Grouping Cue

After building the underlying graph, the grouping cue is applied to all the edgels pairs which are connected by an arc in the graph. The grouping cue that was used in the following experimentation is a binary function that receives two edgels and returns 1 if they could lie on a smooth curve and 0 otherwise. There are many different ways to specify a smoothness criterion (e.g., [HH93, SU88, PZ89, GM92]).

A grouping cue which uses the CEP was used in other parts of this work. Here it is a combination of proximity, co-circularity and low curvature criteria. The main reason for the different cue is to simplify the cue calculation. Let \( v_1, v_2 \) denote the two edgels. Let \( d \) denote the distance between \( v_1 \) and \( v_2 \). Let \( \alpha_1, \alpha_2 \) denote the angles between the edgel direction (orthogonal to the gray level gradient) and the line connecting the two edgels (see Figure 8.2), then:

- **Proximity** - the distance \( d \) must not exceed some maximal distance. This maximal distance should depend on \( k \), because a larger \( k \) means that the effective neighborhood is larger, and thus includes a longer part of the possibly existing smooth curve. Here we use the criterion

  \[
  c_1 = (d < 0.6k),
  \]

  determined so that a point at the end of a smooth curve will have most of its \( k \) nearest neighbors inside a circle of that radius.

- **Co-circularity** - a smooth curve should not have fast curvature changes. Therefore, it should locally match with a circle, and the angles \( \alpha_1 \) and \( \alpha_2 \) are expected to be similar. Due to the high inaccuracy involved in measuring the gradient direction, we used the criterion

  \[
  c_2 = (|\alpha_1 - \alpha_2| < \pi/6),
  \]

- **Similar orientation** - Two co-circular edgels might not fit to any smooth curve if their orientation is very different (see Figure 8.2(b)). Therefore, the difference between the two angles was also limited:

  \[
  c_3 = (\beta < \pi/6).
  \]
• **Minimal radius** - two very close edgels can meet with all the above constraints and still form a high-curvature curve (see Figure 8.2(c)). This constraint limits the local curvature:

\[ c_4 = \left( \frac{d}{\cos\left(\frac{\alpha_1 + \pi - \alpha_2}{2}\right)} > r_{\text{min}} \right). \]

The value of \( r_{\text{min}} \) depends on the subjective definition of what is the maximum curvature of a smooth curve.

![Diagram](image_url)

Figure 8.2: The grouping cue notations are shown in (a). Figure (b) describes an edgel pair which does not lie on a smooth curve but still satisfies the first two constraints. It fails in the third criterion. Figure (c) describes a corner that is detected only by the fourth criterion.

The smoothness criterion is

\[ c(e) = c_1 \land c_2 \land c_3 \land c_4. \]

The smoothness criterion equals 1 if and only if all four constraints hold. Note that this criterion is invariant to similarity transformation (assuming that the pixel size follows the same scale change). This meets Lowe’s first requirement for good cues [Low85]: a good grouping cue must be viewpoint invariant and have good discrimination power (the detection condition). In our framework, the second demand means that the cue should be reliable. It should have small error probabilities - \( \epsilon_{\text{miss}}, \epsilon_{\text{fa}} \).

### 8.3.3 The Iterative Pruning Process and the Bootstrap Mechanism

Given the underlying graph, \( G_u \), and the measured graph, \( G_m \), the pruning process iteratively removes “background” nodes, classifying the nodes into figure and background sets. Each iteration includes two phases: a **node-pruning** phase and an
arc-filling phase. Let \( d_u(v) \) and \( d_m(v) \) denote the degree of a node \( v \) in \( G_u \) and in \( G_m \), respectively.

A node-pruning phase takes the underlying graph and removes nodes that are classified as “background”. A node is classified as “background” if \( d_m(v)/d_u(v) < \gamma \), where \( \gamma \) is a parameter of the algorithm. Otherwise it is classified as “unknown yet”. After this stage, it is likely that some (or even many) of the remaining, “unknown yet” nodes in \( G_u \), are left with less than \( k \) neighbors. If the process continues with a low number of neighbors, then the next classification will be unreliable. Therefore, the node-pruning phase is followed by an arc-filling phase.

In the arc-filling phase, new arcs are inserted into \( G_u \), such that at the end each node is connected to (at least) \( k \) neighbors (in our implementation we build a new graph for the remaining set of nodes, but it is effectively the same). The measured graph is also updated, by removing the same background nodes and applying the grouping cue to the new arcs. Hence the algorithm keep removing nodes and adding arcs (and cues), until it converge. The algorithm is summarized in Figure 8.3.

![Figure 8.3: The ground from figure algorithm.](image)

Before the algorithm starts, the expected value of \( d_m(v_b)/d_u(v_b) \) for a background edgel \( v_b \) is \( \epsilon_{fs} \). However, for a figure edgel \( v_f \) there is an effective miss ratio which is much higher than \( \epsilon_{miss} \), as many of its neighbors are background edgels. Hence, we cannot expect \( d_m(v_f)/d_u(v_f) \) to be as high as \( (1 - \epsilon_{miss}) \), and therefore the classification of \( v_f \) has a high uncertainty. At this early stage, the algorithm does not make any decision about figure points. As the iterations proceed, the expected degree of background edgels, \( d_m(v_b)/d_u(v_b) \), remains unchanged. However, each figure node that “loses” background neighbors is being connected to more figure points in its larger neighborhood, and becomes strongly connected, decreasing its chance to be deleted later on. This bootstrap effect significantly improves performance.
The parameter $\gamma$ controls the tradeoff between the number of missed figure edgels to the number of false alarms in the resulting figure ground set. If $\gamma$ is very small, very few nodes will be deleted, and the resulting figure set will include many false alarms. Similarly, if $\gamma$ is very high (close to 1), then many figure edgels might be deleted, and erroneously classified as background. In our experimentation we set the value of $\gamma$ between $\gamma = 0.2$ to $\gamma = 0.3$, depending on the noise level.

### 8.3.4 Complexity and Run Time.

The algorithm is guaranteed to stop after at most $|V|$ iterations, because at least one node is being deleted each time. In practice, it converges much faster. Moreover, in our implementation we have used an adaptive stopping rule that stops the process if the number of changes in the last iteration was less than 0.2% of the remaining nodes. Using this stopping rule, the algorithm stopped after $5 - 15$ iterations in all our experiments. From our experience, if one does proceed with the processing until no changes occur, the final result is indistinguishable.

Every iteration takes $O(|E|)$ for the node-pruning, and $O(|V| \log |V| + k))$ for the arc-filling (an average time), assuming that a new underlying graph is built at each iteration (this time can be reduced using dynamic data structures). Evaluating the grouping cues takes $O(|E|) = O(k|V|)$. Note that the overall complexity grows linearly with the neighborhood size, $k$ (unlike, for example, [DI96]).

In the following examples, the processing cpu time varied between 16sec for the Mickey-Mouse example (8,871 nodes, 6 iterations), up to 109sec for the craters image (21,000 nodes, 10 iterations). Note that in comparison to other methods, this is a relatively fast algorithm (while comparing run times, keep in mind the number of data features processed by the other methods). Furthermore, we believe that run time can be reduced by almost an order of magnitude if we avoid the re-creation of the entire kd-tree, the underlying graph and the measured graph at each iteration. We have not yet coded this improvement into our implementation.

### 8.4 Experimentation

The algorithm was tested on synthetic images, on computerized tomography (CT) images, and on natural images. Here we provide four detailed examples, reflecting the wide spectrum of tested images. A locally connected underlying graph ($k = 15$) was used for all examples. The minimal radius $r$ and the supporting ratio $\gamma$ were adjusted, to optimize the results (as discussed above in Section 8.3). Edges were extracted by the Khoros standard edge detector, and gradient direction was evaluated by convolving the image with $G_x$ and $G_y$ — the partial derivatives of a Gaussian ($\sigma = 1[\text{pixel}]$).
The first example (denoted “Mickey-Mouse” example) is a synthetic dot image, shown in Figures 8.4-8.5. This example is similar to those used in previous work (e.g., [HH93, SU88, GS91]), but it is much larger (including 8,871 edgels), and is very noisy (the figure set is only 10% of the edgels). The edgels data image (Figure 8.4(c)) includes 50% of the edgels (randomly selected) (Figure 8.4(b)), and nine times more additional clutter pixels (which are 15% of the total number of image pixels), with random orientations. The objects indeed seem to be lost in the noise, although they are more visible in the gradient direction (Figure 8.4(d)). The resulting figure and background sets are shown in Figure 8.5(a), and Figure 8.5(b), respectively. Figure 8.5(c) shows the missed figure points — points which are found in Figure 8.4(b) but not in Figure 8.5(a). The low number of missed points, and the low number of false alarms, is evidence of the high quality of the results available with the proposed method. Finally, Figure 8.5(d) shows the figure set superimposed on the background set, allowing a visual comparison with the original, unclassified data Figure 8.4(c).

The next example is a CT image (Figures 8.6-8.7). This image contains regions of different widths and structurally important edges, associated with a wide range of gradients. It is therefore unlikely that a single edge operator will find all salient edges and ignore the structurally unimportant ones. The ground from figure discrimination removes many of the edges which do not lie on smooth curves (compare Figure 8.7(a) with Figure 8.7(b)) and leaves, in the figure set, most of the boundaries of the main regions in the original image (Figure 8.7(c)). Therefore, higher tasks, such as search for a particular organ (e.g. the kidneys, or the blood vessels), is easier and as reliable when one starts from the figure set and not from the original edge map. A perceptual grouping process may also benefit from a Ground from Figure pre-processing. The grouping result, shown in Figure 8.7(d), was obtained by the algorithm proposed in [Al96b]. Note that the left and right kidneys, the bone, and a few other main parts, correspond to very large single groups.

The Lizard image (Figures 8.8-8.9) is typical to natural scenes. It contains a rich texture, which cause the edge detector to detect many edgels. The ground from figure discrimination successfully removes much of the texture-induced edgels (Figure 8.8(d)), while keeping most of the edgels corresponding to real object boundaries (Figure 8.8(c)). The result is shown in two other forms in Figure 8.9(a) and (b), and the grouping post-processing result is shown in (c).

As is usual in this type of tasks, neither the figure from ground nor the grouping are perfect. Note, for example, the wrong grouping of the lizard’s back with the right-bottom corner of the rock, which meet quite smoothly at the occlusion point. We did not try to make the figure sets prettier by closing the gaps using some edge completion technique. The results are the exact two disjoint subsets of the original edge image. We believe that some of the ground from figure misses may be corrected by edge completion methods (e.g., [Ull76, WJ95, ASZ96, HvdH93]) or by using our
a. Original synthetic image.  

b. Randomly selected figure edgels. 

c. The data: a noisy edgels image. 

d. Gradient direction of edge points \([-\pi, \pi]\).

Figure 8.4: Creating the synthetic “Mickey-Mouse” data. See Figure 8.1 for a magnification of the apple and its edge gradients.
a. Evaluated figure set (11%).

b. Evaluated background set (89%).

c. Missing figure points

\( (Figure \ 8.4(b) - Figure \ 8.5(a)) \)

d. Figure set (black) and background set (gray).

Figure 8.5: Results for the Synthetic "Micky-Mouse" image.
Figure 8.6: Results for the CT image.

a. A CT image.

b. Edgels image.

c. Gradient direction of edge points \([-\pi, \pi]\).

d. Resulting figure set (black) and background set (gray).
Figure 8.7: Results for the CT image (cont.). The ground from figure result is fed into a grouping process, which separates the figure set into smooth curve segments.
saliency map, shown in Figure 8.9(d). This map is a byproduct of the grouping likelihood graph (see Section 7.8 for details).

![a. The Lizard image.](image1)

![b. Edgels image.](image2)

![c. Evaluated figure set (40%).](image3)

![d. Evaluated background set (60%).](image4)

Figure 8.8: The Lizard image has a rich texture, which cause the detection of a lot of edgels. The ground from figure discrimination successfully removes most of the edgels corresponding with the texture, while keeping all those corresponding with the smooth boundaries.

The crater fractal example, shown in Figure 8.10(a), is a very complicate image, and a difficult case for figure ground discrimination. The “objects” (circles) appear in many scales, evoking a lot of edgels (the data image, shown in Figure 8.10(b), includes 22,100 edge points). There are occlusions, junctions, and shadows. The gradient-magnitude gradually decreasing along the circular edges, until they vanish at the nearly horizontal arcs. The shading effects cause the detection of weak vertical edges at the center of each circle.

Figure 8.10(c) and (d) show, respectively, the figure set and the original edge set, superimposed on the gray level image. Figure 8.11 shows two figure ground results, corresponding to different values of \( r_{min}, \gamma \). They demonstrate the scale effect on
a. Figure set (black) and background set (gray).

b. Figure set, superimposed on the (negative) original image.

c. Grouping of the figure set (one gray level per group).

d. Saliency map of the figure set.

Figure 8.9: Results for the Lizard image (cont.). The ground from figure result is fed into a grouping process, which partitions the figure set into disjoint sets, each corresponding to a smooth curve segment. This task becomes much easier after the ground from figure discrimination process.
the smoothness criterion: for higher $r_{\text{min}}$, the large circles remain in the figure set, while smaller circles are classified as background.

Figure 8.10: Results for the synthetic (fractal) craters image. This image is an extremely difficult case; there are a lot of edgels, many small details, occlusions, junctions, and different types of edges: very sharp edges and very smooth ones (of the shadows). (cont. in Figure 8.11).
Figure 8.11: Results for the craters image (cont.). (a) and (b) are for $r_{\text{min}} = 2.0$ and $\gamma = 0.25$. (c) and (d) are for $r_{\text{min}} = 6.5$ and $\gamma = 0.3$. The best values depend on the visual task.
8.5 Conclusions

A new figure from ground discrimination algorithm was presented and its performance was demonstrated experimentally over a set of images, showing good results. The tested images are taken from four different domains: a synthetic edge image, a CT image, a natural out-door image, and a fractal image.

The algorithm is based on the grouping likelihood graph - a data structure that represents the atoms of perceptual evidence: the results of the grouping cues. The proposed algorithm may be considered as a simplified and asymmetric relaxation labeling technique. It is asymmetric because it iteratively prunes background data features, until it stops and hypothesizes that the remaining feature is the figure set.

The algorithm implementation and performance was demonstrated in the domain of an edge image. However, we argue that it should also work in other domains, such as motion based ground from figure discrimination. Small regions which have unique motion parameters\(^1\), are not connected to any other region, and therefore will be classified as background. On the other hand, small regions which have the same motion parameters will support each other, and will remain in the figure set. This would be done in a similar fashion to the generic grouping algorithm, which uses the same graph representation, but changes the cue depending on the domains.

\(^1\)motion parameters corresponds to a motion model, e.g., an Affine motion
Chapter 9

Discussion

The goal of this work is to provide a theoretical framework for grouping processes; perceptual grouping information is extracted from the image using cue functions and is represented by the grouping likelihood graph. This new stochastic model is useful for various tasks in computer vision. It is used here for figure ground discrimination, for grouping, and for hypothesis verification in object recognition.

The proposed approach relies on established statistical techniques such as sequential testing and the maximum likelihood criterion. Statistical techniques are used in some previous grouping. This work is distinctive from past work in that it provides, for the first time, an analysis of the use of these principles. We propose a generic grouping algorithm that would apply to a wide variety of domains and yield predictable performance. The analysis relates the expected grouping quality to the cue reliability, the graph connectivity, and in some cases the computational effort invested.

We did not limit ourselves to theoretical studies. Three grouping applications, each of which is based on a different cue, are implemented as instances of our generic grouping algorithm, demonstrating its usefulness. Although we argued against judging the merits of vision algorithms by visually comparing their results on a few examples, we would like to indicate that our results are similar to those obtained by domain specific methods (e.g. [SU90, HH93] for smoothness based grouping, [AW94] for motion based grouping, and [SU88, GM92] for saliency maps).

Some interesting conclusions arise from our analysis and experimentation with grouping algorithms: it is apparent that higher connectivity, provided either by a complete underlying graph or by a high degree locally-connected graph, can compensate for unreliable cues and enhance grouping quality. Therefore, the selection of cues for a grouping algorithm, should not be based only on maximizing their reliability but also on their extent. The cue extent determines the connectivity of the valid underlying graph, or in other words, the amount of information which may be extracted by this cue.

The cue enhancement possibility, proposed in Chapter 5, should be considered for
every grouping task. We have shown that often, it is possible to obtain more reliable cues (with predicted reliability) with relatively little computational effort. This procedure incorporates multi-feature cues into the grouping process. One way to look at grouping cues is as quantifiers of high order statistical information (higher order than 1st order statistics). A straightforward example is the co-linearity cue described above, which examines the locations of three or more points, and provides a value which depends on their relative locations. The higher order statistical information provides an additional source of grouping information which is often more reliable.

One deficiency of the proposed approach is the existing gap between the theoretical performance analysis of the MLGC optimal grouping and the results of the practical algorithm, for which optimal grouping is not guaranteed. We have analyzed the MLGC criterion from two different aspects: the information aspect and the computational complexity aspect. One one hand, we have shown that if the model assumptions hold, then this grouping criterion can lead to excellent grouping results, depending on the reliability of the grouping cue and on the connectivity of the underlying graph. On the other hand, we have proved that this criterion is difficult to compute - in its general form it is an NP-hard problem. For practical use, however, we have developed a heuristic grouping algorithm, that tries to fill in the gap between theory to practice. Finding an approximation algorithm for the problem would close this gap, and is a subject for future research.

In the design of a grouping algorithm, one may either invest computational effort in enhancing the quality of a relatively small number of cues or use a larger number of unreliable cues and merge them by higher connectivity in the underlying graph. The framework proposed in this thesis makes this choice explicit by providing a cue enhancement procedure, independent from the maximum likelihood graph clustering method. Making the optimal choice is an interesting open question which we consider. Another research direction is to use our methodology in the context of another grouping notion, different from partitioning, by which the hypothesized groups are not necessarily disjoint.

In Chapter 1 we suggested a hierarchical classification of perceptual organization algorithms into saliency mapping, figure ground discrimination, and grouping. It has been shown, for certain classes of visual tasks, that hierarchical tasks are also computationally more complex. Object recognition, for example, becomes more complex when the projection model becomes more general, from orthogonal projection and up to perspective [Lin95, Jac96b]. Another example is some graph clustering methods, which can be sorted in a hierarchical order, and have an increasing complexity as well [Mat77]. In perceptual grouping, it is evident that there are figure ground discrimination methods (e.g., [GM92]) and grouping algorithms (e.g., [SU90]), that first evaluate a saliency map and then use it for their task. It would be speculative, however, to say that the proposed hierarchical order of perceptual organization tasks coincides with
their increasing complexity, as this has never been proved. However, in this thesis we provide a supporting example in which a general model for the representation of perceived organization information is used for all these tasks — creating a saliency map, figure ground discrimination and grouping. In this model, the complexity of the proposed corresponding algorithms increases in the same hierarchical order.

Finally, we believe that various perceptual grouping processes should be combined in order to make a complete system. Ground from figure would be the first filtering stage, followed by edge completion and perceptual grouping. All these stages can use the same data structure and the same grouping cues as their source of information, but may also use different cues at different stages. This is not a new idea, and we believe the proposed framework to be a convenient tool for combining all these tasks.

References


[Zis] A. Zisserman. *personal communication*.

