Conceptual Structures and Computational Methods for Indexing and Organization of Visual Information

Alejandro Jaimes

Submitted in partial fulfillment of the

Requirements for the degree

Of Doctor of Philosophy

In the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2003

ABSTRACT

Conceptual Structures and Computational Methods for Indexing and Organization of Visual Information

Alejandro Jaimes

We address the problem of automatic indexing and organization of visual information through user interaction at multiple levels. Our work focuses on the following three important areas: (1) understanding of visual content and the way users search and index it; (2) construction of flexible computational methods that learn how to automatically classify images and videos from user input at multiple levels; (3) integration of generic visual detectors in solving practical tasks in the specific domain of consumer photography.

In particular, we present the following: (1) novel conceptual structures for classifying visual attributes (the *Multi-Level Indexing Pyramid*); (2) a novel framework for learning structured visual detectors from user input (the *Visual Apprentice*); (3) a new study of human eye movements in observing images of different visual categories; (4) a new framework for the detection of non-identical duplicate consumer photographs in an interactive consumer image organization system; (5) a detailed study of duplicate consumer photographs.

In the *Visual Apprentice* (VA), first a user defines a model via a multiple-level definition hierarchy (a scene consists of objects, object-parts, etc.). Then, the user labels example images or videos based on the hierarchy (a handshake image contains two faces and a handshake) and visual features are extracted from each example. Finally, several machine learning algorithms are used to learn classifiers for different nodes of the hierarchy. The best classifiers and features are automatically selected to produce a *Visual Detector* (e.g., for a handshake), which is applied to new images or videos.

In the human eye tracking experiments we examine variations in the way people look at images within and across different visual categories and explore ways of integrating eye tracking analysis with the VA framework.

Finally, we present a novel framework for the detection of non-identical duplicate consumer images for systems that help users automatically organize their collections. Our approach is based on a multiple strategy that combines knowledge about the geometry of multiple views of the same scene, the extraction of low-level features, the detection of objects using the VA and domain knowledge.

Contents

List of Figures

List of Tables

Acknowledgements

First and foremost I would like to thank my advisor, Shih-Fu Chang, for giving me the opportunity to work with him throughout my Ph.D. His focus on research, enthusiasm, availability, and insight on my work have been pivotal in my development at Columbia. I am very grateful for his support and advice.

I would also like to thank the members of my thesis committee, Nevenka Dimitrova, Alexandros Eleftheriadis, Dan Ellis, and John R. Kender for agreeing to read this thesis in such a short period and for their valuable feedback on my work.

I am also grateful to Peter Allen and Steven Feiner, with whom I first had the opportunity to work with in a research environment at Columbia. Shree Nayar also deserves special mention because taking his excellent Computer Vision class first got me interested in this research area. It was, however, in Shih-Fu's Visual Information Systems course that I realized this was the work I wanted to do.

During my studies at Columbia I have been fortunate to collaborate with many people, inside and outside Columbia. I am very grateful to all of my friends and colleagues at Columbia: Ana B. Benitez, Shahram Ebadollahi, Winston Hsu, Raj Kumar, Seungyup Paek, Qibin Sun, Hari Sundaram, Tian Tsong, Louis Hualu Wang, Yong Wang, Lexing Xie, Donqahing Zhang, and Di Zhong. Kathy

McKeown worked with us on the Digital Libraries project. Kathy's feedback had a strong influence on my early work.

I am particularly grateful to Alexander C. Loui with whom much of the work in chapter 5 was carried out. I would also like to thank Kodak.

Ana B. Benitez's name appears in this thesis more times than any other name. Ana worked with me on some aspects of the work presented in chapters 3 and 5 and presented much of our work in MPEG-7. Thank you Ana for your constant support and friendship!

I am also very grateful to Corinne Jörgensen, who also worked with us on the experiments presented in chapter 3 and on taking our work to MPEG-7. Jeff Pelz of the Rochester Institute of Technology enthusiastically collaborated with us on the eye tracking work in chapter 4, and I am very thankful for his efforts. Tim Grabowski, and Jason Babcock, students of Jeff, made the eye tracking work possible with contributions from Diane Kucharczyk and Amy Silver. I am also grateful to Lawrence Bergman and Vittorio Castelli of the IBM TJ Watson Research center for their feedback on earlier work, much of which formed the base for chapters 2, 3, and 4. MPG-7 colleagues, in particular Toby Walker and Hawley Rising of Sony provided me with valuable feedback on the work of chapter 3.

I am particularly thankful to Rahul Swaminathan for his friendship and comments on chapter 5, and to Efstathios Hadjidemetriou for his feedback on the object recognition section of chapter 2. I would like to thank Enrico Bounano for helping me implement the STELLA interface of chapter 5.

I am also very thankful to the people I have worked with outside of Columbia during my summer internships. Without a doubt each experience enriched me personally and professionally. I would like to thank my colleagues and managers at AT&T Bell Laboratories (James Chang, Edward Sun and Trudy Mulcahy), Siemens Corporate Research (Shih-Ping Liu), and IBM (Howard Sachar, Ravi Rao, Fred Mintzer, Tomio Echigo, and John R. Smith).

Many of my friends read my thesis and provided feedback. I am very grateful to Blaine Bell, Adam Berenzweig, Raj Kumar, Ralf Melke, Tian Ng, Daby Sow, and Rahul Swaminathan for their input and encouragement.

I would like to thank my colleagues at IBM TJ Watson Research Center, Ching-yung Lin, Milind Naphade, Apostol (Paul) Natsev, and Belle Tseng for their support and feedback over the last few months. I am particularly thankful to John R. Smith, my manager at IBM, for his constant support and encouragement while completing this process and working at IBM. Thank you John!

My dear friends Evelina and Jean shared some of the most important non-academic moments of my stay at Columbia.

Lucy deserves special mention for her unconditional love and support during the early years.

Finally, I would like to thank my family. Johanna has been a constant inspiration. Her love, encouragement, and understanding made me and continue to make me very happy. My dad and my brother were and continue to be a source of support. There are no words to describe my mom's generosity, love and kindness throughout the years. I owe you all of my achievements.

To my family

1 INTRODUCTION

1.1 Motivation and Overview

In the last few years, there has been tremendous growth in the availability of multimedia data for *personal use*. This is partly due to better and less expensive technologies to facilitate personalized digital *content creation* (e.g., digital cameras), *acquisition* (e.g., scanners), and *access* (e.g., the world wide web). With the prospects of novel capture and display technologies there is no doubt that digital visual content in the future will become as ubiquitous as paper is today. Furthermore, advances in communications, affective, and wearable computing, assures us that personal *visual information*[1] will be used in unexpected and exciting ways, beyond anything we've been exposed to— personal collections of images and videos will truly be available everywhere, in many forms, at all times (Figure 1).

[1] Throughout this thesis we will use the term *Visual Information* to refer to 2-dimensional images and videos.

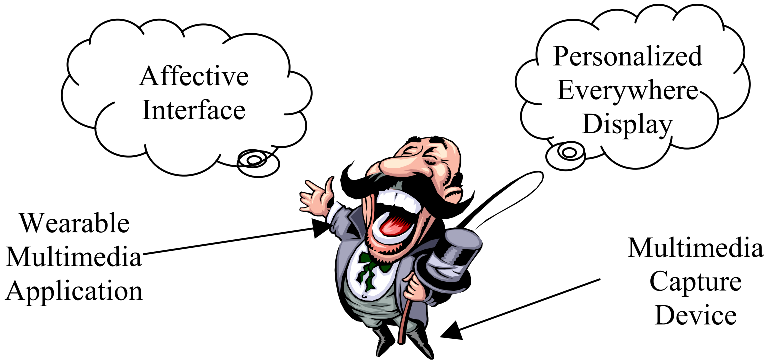

Affective Interface
Personalized Everywhere Display
Wearable Multimedia Application
Multimedia Capture Device

Figure 1. Alex in the future.

This view of the future immediately underlines the need to develop techniques to allow users to effectively *access* and *organize* their own personal digital image and video collections so that they can be used in or by these novel applications. This requires labeling, or indexing of the data at multiple levels. For example, the drawing in Figure 1 can be labeled as "drawing", "man", "man of the future", or "happy man" among others. Such labels can be created manually, but clearly this is not an option even for small personal collections (consider the number of family photographs you own). Fully automatic approaches are desirable in some cases, but are not always achievable with the current state of the art. Furthermore, these algorithms are often constructed by experts for specific applications and cannot accommodate individual user's needs. Therefore, our goal should be to develop flexible techniques that automatically index or label our visual information according to our interests, and that at the same time allow us to make the final decisions about how it is organized and used.

In order to achieve this goal, on one hand, it is imperative to have a deep understanding of visual content and the way users index it and search it. On the other hand we need to develop flexible approaches that learn from users and do not rely entirely on experts, but that index visual information automatically at multiple levels based on users' interests to accommodate users' subjectivity.

1.1.1 Problems Addressed

Based on these premises, in this thesis we address the problem of automatic indexing and organization of visual information through user interaction at multiple levels. Our work focuses on the following three important areas.

(1) Understanding of visual content and the way users search and index it.

(2) Construction of flexible computational methods that learn how to automatically classify[2] images and videos from user input at multiple levels.

(3) Integration of generic visual detectors in solving practical tasks in the specific domain of consumer photography.

More specifically, we present the following: (1) novel conceptual structures for classifying visual attributes (the *Multi-Level Indexing Pyramid*); (2) a novel framework for learning structured visual detectors from user input (the *Visual Apprentice*); (3) a new study of human eye movements in observing images of different semantic categories; (4) a new framework for the detection of non-

[2] Classification is explained in chapter 2. Our use of the word here implies labeling of the data.

identical duplicate consumer photographs in an interactive consumer image organization system; (5) a detailed study of duplicate consumer photographs.

The specific problems we address can be summarized as follows:

1. Understanding of visual content and the way users search and index it.
    a. Construct conceptual structures that can classify the visual attributes of an image into different levels.
    b. Determine if there are patterns in the way people view images of different semantic categories, and whether those patterns depend on the subject, the image, and/or the image categories.
    c. Examine a collection of consumer images that are labeled as duplicates by human subjects and propose a comprehensive model of distinctive classes of duplicates.
2. Construction of flexible computational methods that *learn* how to automatically classify images and videos from user input at multiple levels.
    a. Given an image or video develop algorithms that can automatically assign a semantic label to the image or video based on the objects that appear or on the entire scene depicted. We call these algorithms *Visual Detectors*.

b. Develop a computational framework that, given a set of training examples *S* individually labeled by a user, can generate *Visual Detectors* that label new images or videos according to the examples and object or scene model defined by the user. The *Visual Detectors* should be structured (e.g., objects contain parts) and should be generated automatically from user input without expert intervention. Many *Visual Detectors* are constructed by experts for very specific tasks such as face detection. The problem we address here is different because the construction of detectors requires no expert input.

3. Integration of generic visual detectors in solving practical tasks in the specific domain of consumer photography.

    a. Construct a framework for semi-automatic organization of personal photography collections. Given a set of images, automatically cluster those images so that a user can subjectively manipulate such clusters according to his personal interests.

    b. As needed step in the clustering process, construct a framework that, using *Visual Detectors*, can automatically determine if two images and are duplicates even if they are not identical.

c. Explore the structure of visual information by analyzing the results of eye tracking experiments using different semantic categories.

1.1.2 Outline of the chapter

In section 1.2 we discuss the evolution of photography and how that evolution drives the work presented in this thesis. In section 1.3 we discuss how the work in this thesis relates to the future of multimedia systems. In section 1.4 we briefly describe current visual information indexing systems and review related research. In section 1.5 we summarize the contributions of this thesis and in section 1.6 we explain why the work presented here is important. In section 1.7 we give an outline of the rest of the thesis.

1.2 The Evolution of Photography and Its Implications

1.2.1 Is it Art?

Man has always been fascinated by imagery. The first “images” we can think of where carved into rock during pre-historic times and most of us learn to draw before we are able to write. In modern times, before the invention of photography, practically all imagery, including paintings, sketches, and engravings, was considered art and was created mostly by skilled artists. Systematic reproduction of imagery, pioneered by the invention of lithography brought about a revolution in art and the kinds of images people could own and collect [6]. However, before the invention of photography in the 1860s most people could not create their own images and if they wanted portraits these had to be made by artists. After photography was invented, people lucky enough to afford a camera could now be creators of imagery (art?) via a not-so simple process. Portrait photographers proliferated and the masses, who could previously not afford having their portraits made, could now have their own photographs. In 1900 the creation of imagery was again revolutionized by the introduction of the Kodak brownie cameras— now everyone could easily become a creator and have a personal photography collection. This trend also extended to film with the introduction of 8 mm film cartridges in 1965. Again those fortunate enough to afford such equipment could easily create and view their own personal movies.

The introduction of the VCR in 1972 expanded the types of visual information that people could collect. From family snapshots and home movies to content created by third parties. Today, in industrialized nations, almost everyone has at least one camera and without a doubt an extensive photography collection. The proliferation of digital cameras has only increased the amount of imagery that ordinary people produce. Not only are ordinary people creating imagery, they are collecting it, processing it, organizing it, modifying it, and publishing it. Imagery can now be created by almost anyone, not just by highly skilled artists. Technologies such as those in *TiVO* [257] and the *TV-Anytime Forum* [263] pave the way for a future in which users will easily collect and reuse digital visual information, perhaps by combining it with their own.

As illustrated by Figure 2 the way people acquire and use visual content has changed tremendously. The process has gone from a passive one to a very active one. This trend, driven by technology, has several important implications that form the core of the research presented in this thesis.

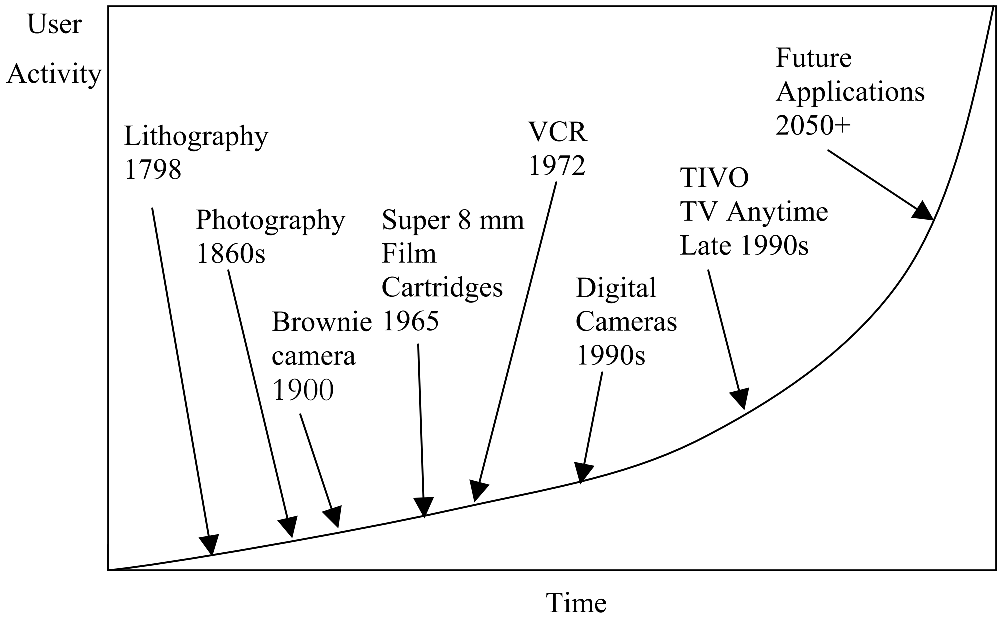


Figure 2. A non-mathematical, historical perspective on the trends in the way people use visual information.

1.2.2 Implications: Active User Role

The implications of the previous discussion are very simple: in the future users can be expected to take an even more active role in creating and using visual information for their own purposes. This completely changes the way people use content collected from other sources and the way they use their own content. The trend of Figure 2 suggests that visual information users of the future will be more sophisticated, more involved, and closer to the process of organization and packaging of their multimedia content. It is only the beginning of the new active user paradigm— for the future many novel applications can be envisioned.

The technical implications of this future trend can be summarized by the following two questions: (1) how can we gain a deeper understanding of visual information, its structure, and its users, to satisfy the needs of future novel applications? (2) how do we develop computational techniques that can adapt to the specific needs of users and consider the structure of visual information and that are flexible and scalable in the sense that the same framework can be applied to different problems without the need for the manual construction of new algorithms or expert input?

The work presented in this thesis directly addresses these two questions. The first one is addressed in chapters 3 and 4 and the second one is addressed in chapters 4 and 5.

1.2.3 Words of Caution: Not All Users Are Equal!

Even as new technologies develop users will be in a range similar to that depicted in Figure 2. Some will remain very passive and will want to get their information pre-packaged, while others will want to be fully involved in the creative process and organization of the collections. Perhaps this has always been the case, but now the balance has shifted and most people in the future will be clustered on the active side of Figure 2, regardless of how exactly multimedia is used in the future.

Our work addresses users in the entire range: the conceptual structures presented in chapter 3 apply to different types of visual information and are independent of the application; the techniques of chapter 4 apply to intermediate users or experts who wish to build systems that can be utilized by passive users; the eye tracking study of chapter 4 is of interest to experts as well as to anyone concerned with understanding how we look at images; the approaches of chapter 5 are geared towards future applications that focus on active users that subjectively organize their photography collections.

1.3 Trends and Needs

1.3.1 Better Understanding of Visual Information

In order to understand visual information it is crucial to recognize that imagery is evolutionary. This means that the photographs (and movies) that people create are the result of minor variations of other visual works they have seen before (see Figure 3). In the history of art this is not a new concept. The first photographs resembled painting and perspective projection was only understood as late as the Renaissance[3], after a long evolutionary process. Art is full of symbolisms and meaning is attached not only to objects, but to forms, colors, and textures. If we wish to develop future applications of visual information, don't we need to consider these different levels of meaning? How can we succeed if we do not understand and do not study all of the levels of meaning that can be encoded in visual information?

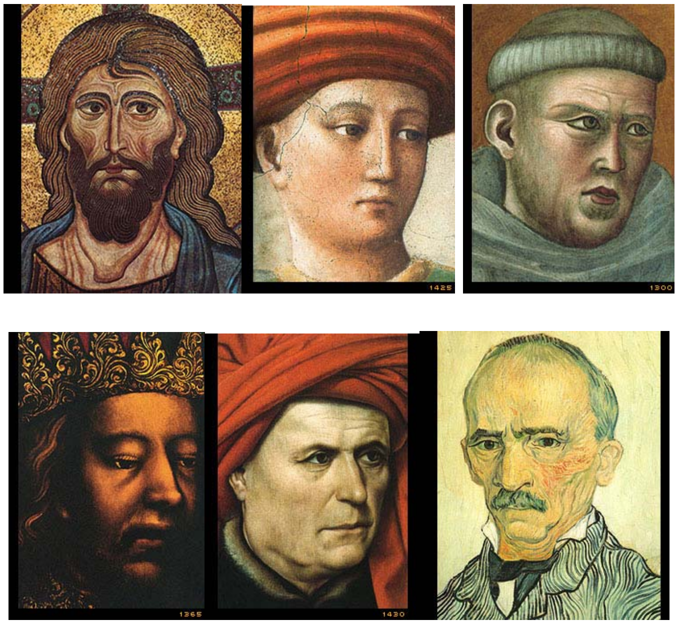


Figure 3. An example of paintings from different time periods.

Man has been studying images and communicating through images for hundreds of years. Nevertheless, it is naïve to assume that we are close to fully decoding the meaning of the messages conveyed in all images. However, in order to build the multimedia applications of the future we need to construct conceptual structures

[3] According to David Hockney's theory artists as early as the 1420s used special optical projection devices [69], and may have understood principles of perspective projection at the time.

that accommodate the different levels of attributes inherent in visual information so that we can develop computational approaches to handle them. This means not only understanding the history of visual information, but also understanding how we look at images.

Computationally, this evolution has even stronger implications. Since all of the images that exist are the product of an evolutionary process, they inherit structure from their ancestors. The photographs most consumers make, for example, follow composition guidelines with roots in art. In domains such as news and sports, it is clear that broadcasters follow patterns set by their predecessors. The way cameras are placed follows certain standards, which change over time. As a result the images that we see, within certain domains, have a recurrent and consistent structure. Consider Baseball video (Figure 4), or practically any sport.

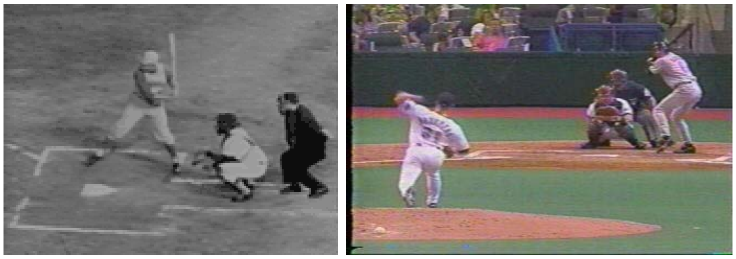


Figure 4. An example of consistent structure due to the evolutionary process. Early (left) and modern (right) batting scene views in Baseball TV broadcast.

As we will argue in chapter 4, structure is important in the application of flexible frameworks that learn.

We examine different levels of visual descriptors in chapter 3, and study the way that people look at images in chapter 4. In chapter 5 we analyze in detail non-identical duplicate consumer images, which is useful in understanding visual similarity judgments.

1.3.2 Personal Scalable Solutions Using Machine Lerning

It is absolutely clear that the world is full of structure. In photographs and video, objects, scenes, and events repeat over and over. Objects and scenes, in turn, have structures that characterize them— objects have parts and parts have sub-parts. Many current automatic approaches to index visual information rely on specific algorithms constructed by experts. The goal of such algorithms (*Visual Detectors*) is to automatically label, or index visual content. While these algorithms can often be robust, it is clear that it is not possible to construct a large number of detectors by hand. Therefore, for future applications it is desirable to build systems that allow the construction of programs that learn, from user input, how to automatically label data without the need of experts. Not only should such systems be scalable but they should also take into consideration users' interests and subjectivity. This requires recognizing the structure inherent in visual information and exploiting such structure in computational frameworks. Without a doubt machine learning will form the basis of successful, scalable applications in the future. Those applications will change, and the way such algorithms learn will change. Perhaps learning will take place without explicit input from users.

The fundamental paradigm, however, will remain. How can we construct frameworks that learn visual detectors and make use of their inherent structure?

We address this question in chapter 4, in which we present a novel framework to construct structured visual detectors from user input at multiple levels. Such detectors exploit implicit and explicit world structure.

1.3.3 Organize Your Own Images

Again referring to the trend of Figure 2 we argue that personal digital multimedia collections will drive the most important future applications of multimedia. People will not stop collecting and making photographs, and the advent of new technologies will open the doors to exciting applications that use our own collections. But how can we drive such technologies?

Regardless of what future applications bring us, it is clear that human subjectivity will continue to be a factor. Just as the trend of Figure 2 suggests, more people will be active, meaning that they will be more involved in producing and using their content. Most of these active users will want to have a final say on how their images are organized and how they are used.

While in some cases it may be desirable to have algorithms that fully automatically organize our content, we need to recognize that current limitations in the state of the art make it impossible.

In chapter 5 we focus on semi-automatic techniques for organizing personal image collections. This is clearly a needed first step for the future use of such collections.

1.4 Current and Future Indexing Systems

Visual information systems consist of four basic components: the *data*, the *indexing structures*, the *interface* and the *users*. The data, in our case images and videos, can be infinitely complex and can clearly be described in an infinite number of ways. The way data is indexed depends on factors such as context, domain, and application, among others. The way visual information is searched has changed in the last few years and there are now new paradigms for searching and browsing. Basic concepts and techniques for *Visual Information Retrieval* (*VIR*)[4] are described in detail in chapter 2. The overall goals, however, remain the same: given a personal image or video database, how can users effectively find what they are looking for? How can they subjectively organize their collection or browse through it?

1.4.1 Related work

The continuing increase in the amount of visual information available in digital form has created a strong interest in the development of new techniques that

[4] We will also use the terms Content Based Retrieval.

allow efficient and accurate access of that information. One crucial aspect is the creation of indexes (e.g., labels, or classifications) that facilitate such access. In traditional approaches, textual annotations are used for indexing— a cataloguer manually assigns a set of key words or expressions to describe an image. Users can then perform text-based *queries*, or *browse* through manually assigned categories. In contrast to text-based approaches, recent techniques in Content-Based Retrieval (CBR)[5] [102][92][223][273] have focused on automatic indexing of images based on their visual content. The data in the images or video is indexed, and users can perform *queries by example* (e.g., images that look like this one) or *user sketch* (e.g., images that look like this sketch). Traditional query-by-example and query-by-sketch techniques are usually based on low-level features (color, texture, etc.), thus concentrating on the form of the visual information (*syntax:* color, texture, etc.) rather than on the meaning (*semantics:* objects, events, etc.). Users, however, are generally interested in semantics. Moreover, even when syntax is a necessary part of the query, the user frequently wishes it to be subordinate to semantics— the query "show me all the images that contain green apples" is more likely than the query "show me all the green areas that are apples". Although useful, low-level features are often unsuitable at the semantic level: meaningful queries by sketch are difficult to formulate, and example-based queries often do not express semantic level distinctions.

[5] Throughout the thesis we will use the term *Visual Information Retrieval (VIR)* instead.

Consequently, most recent efforts in VIR attempt to automatically assign semantic labels to images or videos. Proposed techniques range from classification mechanisms, to approaches that structure and describe data. Automatic classification of images, for example, can be done at the object (e.g., the image contains a horse, or a naked body [113]), or scene level (indoor and outdoor [254], mountain scene [264], etc.). Classification of video can also be performed at the scene or object levels. Several other approaches attempt to automatically structure data (image collections or videos). Additionally, there are major efforts to create structures and standards for the description of multimedia content. MPEG-7 [91][192] for example, standardizes a framework for describing audio-visual content.

In addition to issues regarding the complexity of visual information, there are issues related to users and their subjectivity when searching/browsing. It is well understood that different users search for information in different ways and that their search criteria change over time [185][212]. Related work in this and other areas is described in subsequent chapters.

1.5 Summary of Contributions and Impacts

The contributions of this thesis can be summarized as follows. In this thesis we performed the following tasks:

1. Develop a conceptual framework for classifying visual attributes into ten levels. The framework makes distinctions between *syntax* and

semantics and between *general concepts* and *visual concepts*. The *Multi-level Indexing Pyramid* was tested extensively in the context of *MPEG-7* [239]. Many of the concepts and components in the pyramid are included in the standard [190][191][176]. Other researchers [37] have independently tested the ability of the pyramid to classify image descriptions and found that it is able to fully classify all of the attributes generated in image description experiments.

2. Present a new framework for learning structured visual detectors from user input at multiple levels. The *Visual Apprentice* has been used in constructing detectors for different applications (e.g., baseball, news, and consumer photography) in the context of several projects [206][149][139]. It has also been used as an intermediate tool for manually constructing highly effective detectors for indexing baseball video [275].

3. Present a novel study that compares the eye tracking patterns of human observers within and across different semantic image categories. This study is important because it is the first one to compare eye movements within and across different image categories. Finding eye tracking patterns can be useful in understanding the way people look at images and in constructing automatic classification algorithms.

4. Explore automatic clustering of consumer images within a system that helps users organize their own photographs. We propose the following:

    a. A new clustering algorithm based on a simple variation of Ward's minimum squared error algorithm.

    b. New features for clustering images based on composition.

    c. A novel computational framework for automatically detecting non-identical duplicate consumer images. The framework uses multiple strategies based on knowledge of the geometry of different images of the same scene, and the integration of low level features, visual detectors, and domain knowledge to measure similarity. This is a *new* problem, which has not been addressed in the literature before, and the framework specifically developed to address it is unique.

5. Present a new study in which subjects labeled non-identical duplicate and non-duplicate pairs in an extensive consumer image database.

6. Develop a new extensive classification of non-identical duplicate consumer photographs. Understanding of such a classification helps the development of effective algorithms to detect non-identical duplicate images. This is the first comprehensive classification of duplicate consumer photographs.

1.6 Why the work in this thesis is important

We have argued that future users of visual information will play a more active role in using the images and video they collect. The systems and approaches to search, display, and use information will, without a doubt change. New capture devices will be developed which will facilitate automatic understanding of multimedia data. Knowledge repositories will be an inherent part of multimedia systems and personalized visual information will be everywhere, in different, unforeseen forms.

These exciting future changes, however, will require advances in the fundamental problems of visual information indexing. These fundamental problems will have to be addressed before we can build the exciting systems of the future. Furthermore, these problems will not change because they are at the core of future applications that use visual information. We summarize these problems:

1. Understanding the data and the users.
2. Constructing flexible systems that learn.
3. Investigating approaches to help users organize their own visual information collections.

The problems addressed by this thesis are therefore fundamental to the development of new approaches to enable future applications. Contributions in understanding visual information will surely have an impact on future applications. Even if we do not know what those applications are we will still need to have a deep understanding of the data. How to construct systems that learn is without a doubt a major challenge addressed by this work. Recognizing subjectivity and the need to develop frameworks that help users organize their collections is also crucial.

1.7 Outline of Thesis

In chapter 2 we present some basic concepts and an overview of current work in addressing the problems just described. In chapter 3 we present the *Multi-level Indexing Pyramid,* in which visual attributes are classified into ten levels. In the *Visual Apprentice* (VA) in chapter 4, first a user defines a model via a multiple-level definition hierarchy (a scene consists of objects, object-parts, etc.). Then, the user labels example images or videos based on the hierarchy (a handshake image contains two faces and a handshake) and visual features are extracted from each example. Finally, several machine learning algorithms are used to learn classifiers for different nodes of the hierarchy. The best classifiers and features are automatically selected to produce a *Visual Detector* (e.g., for a handshake), which is applied to new images or videos.

In the human eye tracking experiments of chapter 4 we examine variations in the way people look at images within and across different visual categories (e.g., landscape, people shaking hands, etc.) and explore ways of integrating eye tracking analysis with the VA framework.

Finally, in chapter 5 we present a novel framework for the detection of non-identical duplicate consumer images for systems that help users automatically organize their collections. Our approach is based on a multiple strategy that combines knowledge about the geometry of multiple views of the same scene, the extraction of low-level features, the detection of objects using the *VA* and domain knowledge.

2 BASIC CONCEPTS AND LITERATURE REVIEW

2.1 INTRODUCTION

In this chapter we give an overview of basic concepts and existing approaches related to the problem of indexing and organization of visual information through user interaction at multiple levels.

While we do not present an exhaustive literature review, we do focus on the three *areas* discussed in chapter 1.

(1) Understanding of visual content and the way users search and index it.

(2) Construction of flexible computational methods that learn how to automatically classify images and videos from user input at multiple levels.

(3) Integration of generic visual detectors in solving practical tasks in a specific domain.

This chapter, therefore, is divided into three parts, each of which deals with issues that are relevant to each of these areas.

The *first* area is very broad and one can easily find hundreds of topics of interest in different fields dating back hundreds of years. For the purposes of the problem we are addressing, however, at least two paramount issues can be identified: (1) what are the basic concepts that, based on our understanding of visual content and users, can help us build computational approaches to automatically index and organize visual information? (2) what do we know about the way in which users search and index visual content and how can we use that knowledge to build better computational frameworks? Clearly, seeking the answers to these questions will be of great importance in addressing the problem of automatic indexing and organization of visual information. To address the first question, in the first part of the chapter, we introduce some basic concepts such as *percept*, *concept*, *syntax*, and *semantics*. These definitions are of paramount importance as they will be used throughout the thesis and help us understand different aspects of visual information. Since some of them are directly related to the pyramid, they were proposed to *MPEG-7* [76]. Several of these terms are included in the standard [190]. The second question is addressed in detail in chapters 3 and 4.

The *second* area, construction of flexible computational methods, certainly requires attention to at least two aspects: (1) what are the basic ways in which users can interact with current visual information retrieval systems and at which levels does that interaction take place? (2) what is the state of the art in flexible computational approaches that are based on user input and how are the current approaches used? What are their strengths and limitations? In the second part of the chapter, we discuss the basic components of VIR systems. In particular, we focus on query interface modalities and interfaces, discussing in detail some of the current approaches and systems. This section is important because in order to build effective computational frameworks that learn from user input we need to understand the different ways in which users can interact with such systems. Interfaces will change in the future, and perhaps the flexible systems we envision will not require the type of user input of current systems. As our discussions suggest, however, some of the limitations are tightly linked to the different levels of meaning inherent in visual information (see also chapter 3). The analysis in this chapter, therefore, serves to identify those issues even though this is done in the view of current systems. The discussion on similarity, for example, is strongly related to the problem addressed in chapter 5. Naturally, we also review in some detail different types of flexible computational approaches, including those that use relevance feedback and learning, which are directly related to the framework of chapter 4.

The *third* area, integration of generic visual detectors in specific domains, presents many formidable challenges, some of which are described in this chapter. Not only is it necessary to have a deep understanding of the domain, it is also necessary to understand the advantages and limitations of different approaches to construct the visual detectors. Since we study our application domain (consumer photography) in detail in chapter 5, in this chapter we consider the following issues: what is the state of the art in the construction of visual detectors and how are they applied? What are the strengths and limitations of applying such detectors to specific domains and how do flexible approaches address the limitations or benefit from those strengths?

In the third part of the chapter we address these questions. First, we give a somewhat detailed overview of the main approaches to construct visual detectors for VIR systems. Then, we discuss the object recognition problem and some model and appearance-based methods related to the work of chapter 4. The distinction between approaches developed for VIR and "traditional" approaches in object recognition is important, particularly in the context of applying generic visual detectors in specific domains. The differences are discussed in detail and at the end of the chapter and we discuss the challenges that arise when object recognition techniques are used in practical applications. This discussion is particularly relevant in the context of the work presented in chapters 4 and 5.

2.1.1 Outline

In section 2.2, we present basic concepts related to understanding of visual content and users. In section 2.3 we address issues related to the construction of flexible computational methods that learn. Finally, in section 2.4, we address integration of generic visual detectors in specific domains.

2.2 UNDERSTANDING OF VISUAL CONTENT AND USERS

In this section, we define some basic concepts that involve users and visual content. Related work and issues about understanding of visual information and users are dealt with in detail in chapters 3 and 4.

2.2.1 Percept vs. Concept

Images are multi-dimensional representations of information, but at the most basic level they simply cause a response to light (tonal-light or absence of light) [12]. At the most complex level images represent abstract ideas that largely depend on individual knowledge, experience, and even particular mood. We can make distinctions between *percept* and *concept* [1]. The *percept* refers to what our senses perceive, which in the visual system is light. These patterns of light produce the perception of different elements, such as a specific texture and a particular color. No interpretation process takes place when we refer to the *percept*: no knowledge is required. A *concept*[6], on the other hand, refers to a

[6] Definition from Merriam-Webster dictionary.

representation, an abstract or generic idea generalized from particular instances. As such, it implies the use of background knowledge and an inherent interpretation of what is perceived. Concepts, which can be described in different ways, can be very abstract and subjective in the sense that they depend on an individual's knowledge and interpretation.

2.2.2 Syntax vs. Semantics

While a *percept* refers to what we perceive through our senses (e.g., visual elements), *syntax* refers to the way in which those visual elements are arranged. When referring to syntax, no consideration is given to the meaning of the elements, or to the meaning of their arrangements. *Semantics*, on the other hand, deals with the meaning of those elements and of their arrangements. As will be shown in chapter 3, *syntax* can refer to several perceptual levels— from simple global color and texture to local geometric forms, such as lines and circles. *Semantics* can also be treated at different levels.

2.2.3 General vs. Visual Concept

Here we wish to emphasize that general concepts and visual concepts are different. A visual concept includes only visual attributes, while a general concept can include any kind of attribute. We use a ball as an example. One possible general concept can represent a ball as a round mass. Concepts for similar objects may vary between individuals. For example, a volleyball player's concept of a ball may be different from a basketball player's concept of a ball because, as described earlier, a concept implies background knowledge and interpretation. In Figure 5, we see that the attributes used for the general and visual concepts of a ball can differ. Each box represents a universe of attributes[7], and each circle, the set of attributes observers A and B choose to describe a ball. Attributes outside the circles are not chosen by the observers to describe this particular concept. Observer A is a volleyball player, and when asked to describe a ball, she chooses soft, yellow, round, leather, and light-weight attributes. Observer B is a baseball player, and when asked to give describe a ball, she chooses hard, heavy, white, round, and leather attributes. Note that, naturally, there is also a correlation between some general and visual attributes (e.g., big) and visual attributes are a subset of general attributes.

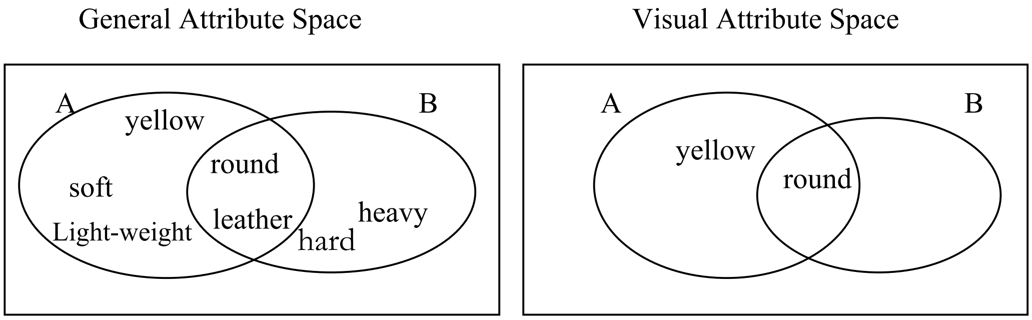


Figure 5. We divide attributes into those that are general (a) and those that are visual (b). Attributes in each set (A, B) are used by different individuals to describe the same object in this example (a ball).

[7] In this section, we use the word attribute to refer to a characteristic or quality of an object (e.g., blue, big, heavy). We do not make a distinction between attribute name and attribute type (e.g., color: blue).

These basic definitions are useful since they point out very important issues in indexing visual information: different users have different *concepts* (of even simple objects), and even simple objects can be seen at different conceptual levels. Specifically, there is an important distinction between *general concept* (i.e., helps answer the question: what is it?) and *visual concept* (i.e., helps answer the question: what does it look like?). We apply these ideas to the construction of our conceptual indexing structures in chapter 3, and to the development of techniques for automatically learning visual detectors in chapter 4.

2.3 FLEXIBLE COMPUTATIONAL METHODS THAT LEARN

We present an overview of the main search approaches in current VIR systems. We emphasize differences between the query formulation paradigm[8] (e.g., query-by-example, or similarity), and the indexing technique (e.g., placing items in semantic categories or indexing using low-level features). Then we describe interactive techniques which are directly related to the work of chapter 4.

[8] We will use the terms query formulation, and interface interchangeably.

2.3.1 Basic Components of Visual Information Retrieval

Practically all current systems to retrieve visual information can be characterized by the *interface* and the *indexing mechanism*. The *Multi-level Indexing Pyramid* presented in chapter 3 is particularly useful in our analysis. The pyramid, depicted in Figure 16 contains ten levels. The first four levels, *(1) type or technique,(2) global distribution, (3) local structure,* and *(4) global composition*, are used to classify *syntactic* visual attributes. The remaining levels, *(5) generic object, (6) generic scene, (7) specific object, (8) specific scene, (9) abstract object,* and *(10) abstract scene,* are used to classify *semantic* descriptions of visual information. In the discussions below we will refer to different levels of the structure.

2.3.1.1 The Interface

When users perform a search, they interact with a system to express what they are looking for (Figure 6). This is done by providing the system with a representation for a query. The system must then interpret the query and, using the database's indexing information, return relevant images to the user. In some cases, the query representation is fairly straightforward because the interface allows the user and the system to "speak the same language." In other words, the interface lets the users express exactly what they want to express, and the index contains enough information to represent a user's interests. For example, a user is looking for any image that contains a horizontal line and performs a query by drawing a sketch of a horizontal line. The system can then return images that have a horizontal line. For this query to be successful, it is necessary for the user to be able to express his request unambiguously using the interface, and for the index to have enough information to satisfy that query. In this particular case, the representation used for the query (i.e., the straight line) matches the query itself very well since the user is actually searching for a straight line. When the user is looking for semantics, however, it might be difficult for the system to interpret the representation used to specify the query. The line in the example may represent the surface of the ocean, or a landscape.

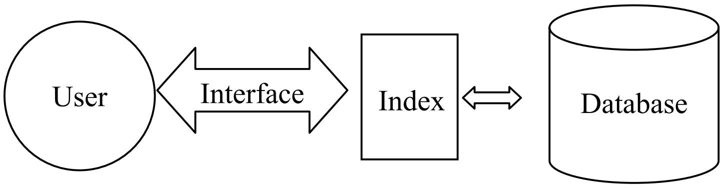


Figure 6. A *Visual Information Retrieval* system consists of 3 basic components: the *database* (e.g., images), the *index* (e.g., color histograms, or textual labels), and the *interface* (e.g., query-by-example).

The interface allows the user and the system to communicate. Ideally this communication will take place using a common *language*[9]. Two important aspects of the language used to perform the query are its *expressive power* and its *ease of use*. By expressive power we mean *what* can be expressed using the language. The second aspect refers to *how* difficult it is for the users to formulate the desired query using the language. Of course, such language should be common to system

[9] Note that the strict definition of language requires the existence of symbols, syntax, and semantics.

and user, otherwise, misunderstandings will occur— in other words, the index must represent information that is not very different from what the user expresses using the query. In Figure 7 we classify the different interface modalities available in most existing VIR systems[10], and show how they relate to the conceptual structure of chapter 3.

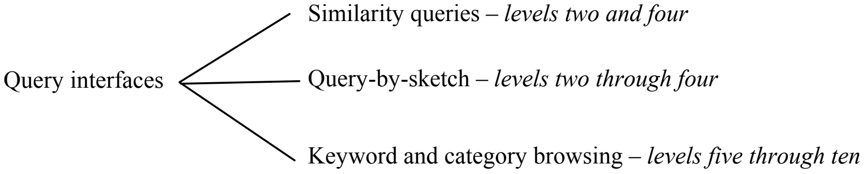

Query interfaces
Similarity queries – *levels two and four*
Query-by-sketch – *levels two through four*
Keyword and category browsing – *levels five through ten*

Figure 7. Types of query interfaces and relation to the levels of the conceptual structure in Figure 16.

As the previous example suggests, it is important when developing approaches for indexing and organizing visual information to be aware of the way in which indexing structures represent information (i.e., structures in chapter 3). Similarly, the query interface modalities play an important role. Each of these interface modalities (similarity, sketch, and keyword) will be further explained in Section 2.3.2.

2.3.1.2 Indexing

The interface is the mechanism through which the user and the system communicate. The index, on the other hand, is data that provides the access point

[10] We use the terms *similarity query* and *query-by-example* interchangeably.

to the information in the database. Different modalities to index visual content provide access to the images at different levels. As depicted in Figure 8, indexing techniques in most current Visual Information Retrieval can be roughly divided into two groups, depending on the types of features they use: (1) local and global syntax features; and (2) local and global semantic features.

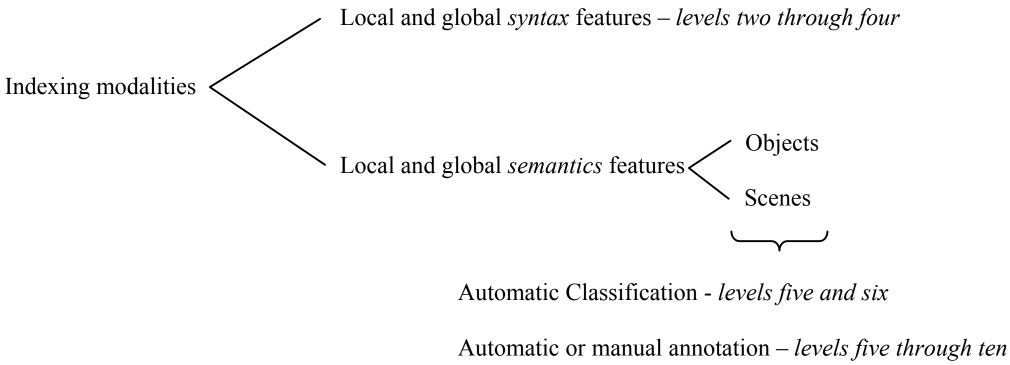

Indexing modalities
Local and global *syntax* features – *levels two through four*
Local and global *semantics* features
Objects
Scenes
Automatic Classification - *levels five and six*
Automatic or manual annotation – *levels five through ten*

Figure 8. VIR indexing modalities and their relation to the structure in Figure 16.

In the first modality, indexing is usually performed automatically, using color, texture, and other features, at levels two through four of the pyramid of chapter 3 (Figure 16). In the second one, images are classified according to the objects they contain, or according to the scene they represent (e.g., indoor/outdoor). Currently manual annotation is mostly used at levels five through ten of Figure 16, but it can be used at any level, while automatic indexing at semantic levels (classification) currently occurs only at levels five and six.

Low-level features can be very powerful, but they are often not suitable for indexing at the semantic levels. For example, Figure 9 shows two images having very different color histograms, but very similar semantic content.

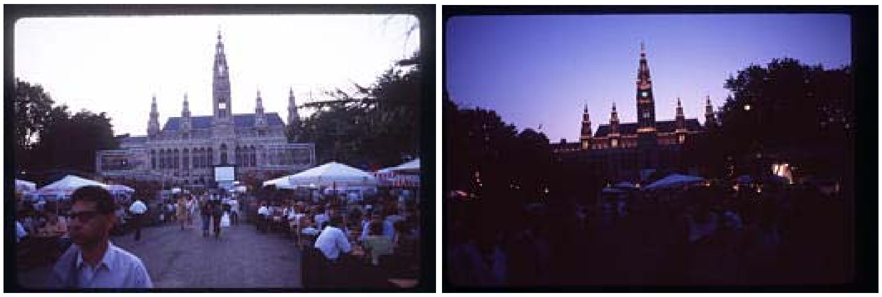


Figure 9. Two images having very distinct color histograms, but identical subject matter (photographs A. Jaimes).

In chapter 3 (and [48]) it was shown that users employ emotive attributes (e.g., sad or happy) to describe and access images. Some attempts have been made to automatically index images at emotive levels (e.g., [98]), but most focus on recognition of objects and scenes.

Manual annotation continues to play a very important role in many practical applications. In the future, we envision systems that facilitate manual annotation of visual information, and allow users to exploit that information when searching. We discuss this further in chapter 6.

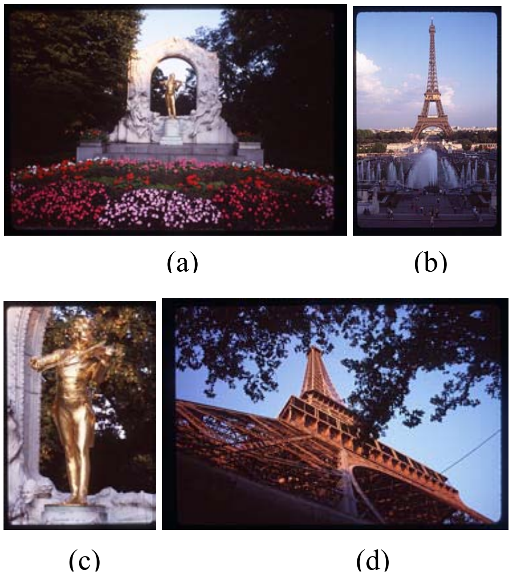
2.3.2 Query Interfaces

We identify three different interface modalities (Figure 7): *query by example, query-by-sketch*, and *keyword/category browsing*. We also discuss interactive techniques: *relevance feedback,* and *learning*.

2.3.2.1 Query by Example and Issues of Similarity

In query-by-example (similarity query) interfaces, the user selects one or more images (or portions of them as in the SPIRE system [81]) and queries the system for images that are similar. Typically, low-level features (e.g., color, texture, etc.) are used to represent the example images and to perform the query. While this approach is very useful for certain types of databases (e.g., textile fabrics, wall paper, etc.), it suffers from major disadvantages when users are interested in semantic content. This is because low-level features used by current systems do not capture the semantics of the query image, and because the concept of similarity is particularly complex with visual information. While this type of interface is *easy to use*, the *expressive power* of such queries is limited. In certain cases, however, it is possible to use query-by-example interfaces to search for semantics. In the SPIRE system [81], for instance, higher level objects can be constructed from portions of images. Similarly, in QBIC [200], semantically meaningful regions are manually selected when the database is populated. For example, if the image is from an on-line catalog, and represents a person wearing a sweater, only the region corresponding to the sweater would be indexed, and the rest of the image ignored. When a user performs a query-by-example using the image or a portion of the image, semantic information (e.g., the region from the sweater) is used to search for meaningful results. Of interest is also the VEVA query interface language [105], which is a visual language based on a set-theoretic formal model of functional databases.

As shown by the *Multi-level Indexing Pyramid* of chapter 3, semantics can be considered at many different levels. Images (a) and (b) in Figure 10, for example, could be considered similar because they show a complete object at some distance. Images (c) and (d) in the same figure are similar because they are close-ups of objects. At the same time, we could say that images (a) and (c) are very similar because they contain the same object. They are dissimilar, however, in terms of colors and composition (syntactic attributes).


(a) (b)

(c) (d)

Figure 10. An example of images that could be considered similar/dissimilar depending on the basis used to measure similarity (photographs by A. Jaimes).

Another important issue with similarity is that it often depends on context. For example, an observer given 3 circles and 2 squares is asked to group the most similar items. If she uses shape, the 3 circles would be in one group and the squares in a different group. If size is used, however, 2 small squares and a small circle would be grouped together. Note that "small" depends on how objects are compared, in other words, on the metric used to measure similarity. Another alternative would be to group the elements based on overall similarity (where shape and size are equally weighted). This is shown in Figure 11.

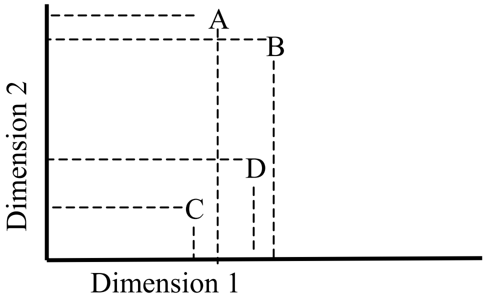


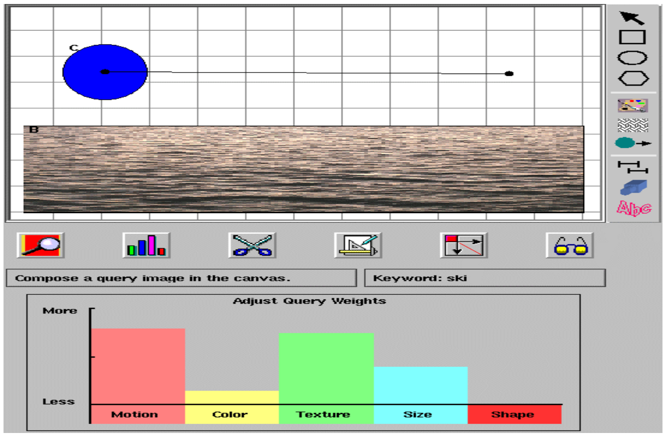
Figure 11. An example from [28] of how four elements can be grouped depending on the feature used to determine similarity. If we consider overall similarity (measured, in the example, by the Euclidean distance between representative points), reasonable groups would be AB and CD. If we consider dimension 1 the groups would be AC and BD. Considering dimension 2 the groups would be AB and CD.

It is apparent from the figure that similarity of the objects depends on the context. If A and B were the only elements, there could be little doubt about placing them in the same group.

Context, semantic information, and the features used to measure similarity play a crucial role in VIR systems and have a strong impact in the problems we address in this thesis. As we will see in chapter 5, many of the difficulties encountered in building computational frameworks to organize visual information can be attributed to these issues.

2.3.2.2 Query by sketch

In query-by-sketch systems, users typically draw a sketch corresponding to the image or video they are interested in (Figure 12). In addition to drawing a sketch, the user is usually required to determine which features are important for each query (e.g., color, texture).



Compose a query image in the canvas.

Keyword: ski

Figure 12. A query-by-sketch example from *VideoQ* [89]. In this example, the user has drawn a sketch to represent a water skier. The user has given the water a particular texture, and has placed the highest importance on motion and texture features (followed by size, color, and shape, which is unimportant in this query).

Regardless of whether the user is able to draw a "good" example to perform the query, the success of the system depends on how the query is *interpreted*. In Figure 13, a naïve user performs a sketch query hoping to retrieve an image of a person standing in front of a building. The photograph is *conceptually* similar to the sketch, but very different in terms of the drawing itself. Current query-by-sketch approaches [89][136] focus on the form of the sketch (i.e., syntax), and not on what the sketch may represent (i.e., semantics). Their *expressive power* is limited to syntax— the *language* used by the system expresses only syntax.

In spite of that limitation, users that understand such systems well and have enough practice/abilities in drawing their queries can use query-by-sketch systems effectively. Although unsuitable at the semantic level, this approach supports queries that retrieve images based on local structure (i.e., level 3 of Figure 16). In the example of Figure 13, a more accurate sketch could be effective in retrieving the desired image, but users must know exactly what they are looking for and have exceptional visual memory to remember important details (e.g., size, location, 3D perspective, etc.).

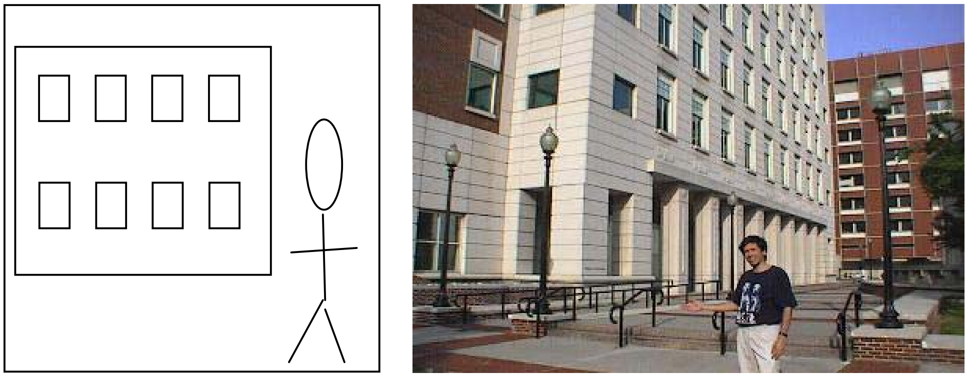


Figure 13. A sketch used to represent a concept (a person in front of a building), and the image the user may actually want to retrieve (photograph A.B. Benitez).

2.3.2.3 Keyword and Category Browsing

As discussed earlier, text provides an excellent way to communicate with the system at levels five through ten of the structure of Figure 16. This is particularly true at the abstract levels. It is well known, however, that there are serious limitations to the use of text alone.

Some information is easily accessible using text. A query for a “car” is easily formulated by typing in that keyword or browsing in that category. In some applications, (e.g., medical systems), some of the information can only be accessed using text: it would be difficult to find a particular patient’s x-ray otherwise. Text annotations can also carry important image content information that would be difficult to find using a different mechanism (e.g., a medical term to explain an anomaly in an x-ray image).

2.3.3 Interactive Techniques

Several techniques have been developed to reduce the limitations of sketch and similarity interfaces. The main goal of such techniques is to increase the expressive power and ease of use of those interfaces. In this section, we discuss approaches in which the results obtained with query interfaces (query by example and query by sketch) are utilized in a feedback loop by the user. The framework we present in chapter 4 requires fairly straightforward and simple input. Nevertheless, it is possible to use the same concepts of that chapter with more complex types of input, such as the ones we describe here. Relevance feedback, for instance, could be easily integrated, and the interface types just described could be used.

Relevance Feedback

In *Relevance Feedback* [224][219], a user performs an initial query and the system returns the best matches. The user can then label as positive matches those images that are most similar to the ones in which she is interested, and as negative matches the ones most dissimilar. The system then performs a new search utilizing the user's feedback. The process repeats as many times as needed as the user continues the search.

Many different relevance feedback approaches have been proposed [244][99][182][95][166][224]. In particular, feedback can be *categorical* (i.e., the user indicates what items are in the same category as the target), or *relative* (i.e., item A is more relevant than item B) [99]. For example, in [224], the user marks each returned image with one of the following labels*: highly relevant, relevant, no-opinion, non-relevant*, or *highly non-relevant.* In [99], however, the user simply selects the images that are closer to the target concept than the unselected images.

The main advantage of relevance feedback is that the search is guided by the user. A drawback of these techniques, however, is that they assume that the database is rich enough to return "good matches" that can guide the search process. This may not always be the case. In addition, the problems (and benefits) associated with different types of interfaces outlined in section 2.3.2 should still be considered (e.g., ways to express similarity, etc.).

Learning

An alternative to performing relevance feedback at the time of the query is to learn what the user wants allowing her to assign class labels to the objects/image classes in which she is interested. In other words, the user may label some image regions with a name such as "sky" and others with a name such as "not sky." The system can then use that information to learn the concept in which the user is interested (e.g., sky), and construct a classifier— a program that, given an input (e.g., a region), makes a decision to determine the class of that input (e.g., sky, not sky). The classifier learned can then be used to automatically label content (e.g., indoor, outdoor image). Once the content is labeled, it can be accessed using the keyword/category browsing approach discussed in section 2.3.2.3.

The learning of classifiers is closely related to indexing and classification (section 2.3.1.2), to object recognition (section 2.4.2), and to learning visual detectors (chapter 4). The process of learning from the user, however, often requires some form of user interaction, which is related to the other techniques discussed in this chapter. We discuss this further in the next section.

2.4 INTEGRATION OF VISUAL DETECTORS IN SPECIFIC DOMAINS

In the previous section we discussed different query and indexing approaches, including some interactive frameworks. In this section we focus specifically on the construction of *Visual Detectors* (i.e., programs that automatically label objects or scenes in images or videos). These discussions are relevant to the framework of chapter 4 for building visual detectors from user input, and to the problem addressed in chapter 5.

We make a distinction between scene classification approaches and work in object recognition.

2.4.1 Scene Classification

Classification seeks to place images into specific semantic categories (i.e., a classifier is a function that, given an input, assigns it to one of k classes). Typically, this is performed at the scene level (e.g., indoor, outdoor, etc.), and the object level (e.g., objects in the image are labeled, or the image is labeled with the names of objects it contains).

In many classification approaches, a training set is used to build the classifier. For example, a training set for an indoor/outdoor scene classifier would consist of a set of indoor images and a set of outdoor images. Each of the approaches described below uses training sets in different ways.

Content-based classification using visual features can be scene-based (e.g., using global low-level features [254][265]; region configuration-based [243][171][86]) or object-based (e.g., detection of faces in [114], naked people and horses in [113], objects defined by the user in [145]).

In scene-based classification, the image as a whole is given a semantic label. Examples include indoor vs. outdoor [254], city vs. landscape [117][265][264], beach, sunsets [243][171], etc.

In the particular approach presented in [254], the image is first divided into a 4x4 grid of 16 blocks. Low-level features are then computed for each block and for the whole image. Features for color (Otha color histogram [204]), and texture (Multiresolution Simultaneous Autoregressive Model [177] and DCT coefficients) are computed for each image and for each block. Using a multi-stage classification approach, the blocks are classified independently and the results are used to classify the images. More specifically, for each image, each of the 16 blocks is classified using a nearest-neighbor classifier. The classification results of each block for each image are concatenated into a feature vector. In the second classification stage, a majority vote classifier is applied to the vectors, yielding a final result for the image.

The division of the image into blocks is useful because it tends to capture spatial characteristics of images belonging to the same class. For example, in outdoor images the sky appears at the top, while in indoor images sections of wall are common in the upper part of the images. Features, such as color and texture, also capture characteristics of images in each class. Indoor images tend to have textures that are formed mostly by horizontal and vertical edges. The colors and texture of elements that appear in outdoor images are quite different from those in indoor images.

Other approaches similar to the one in [254] have been developed to classify images. In [264], for example, images are divided into 10x10 sub-blocks for classification. Features, such as edge direction histograms, are used to discriminate between city and landscape images.

In [243] images are classified using *Composite Region Templates (CRT)*. The basic idea in *CRTs* is that images that belong to the same semantic class show a similar configuration of regions. For example, in a beach scene, the top region is blue (sky), with a yellow region (sand) beneath it. Initially, a string is generated which represents the configuration of regions in an image in a vertical scan (e.g., Blue followed by yellow, etc.). Instead of directly comparing the strings, region configurations called *Composite Region Templates* are created. A *CRT* is a relative ordering of M symbols from the string (e.g., “Blue Yellow” for M=2). For example, suppose we have two strings that represent two images and their respective regions: $S_a=s_0s_1s_2s_3$ and $S_b=s_0s_1s_3$. The *CRT* $T=s_0s_3$ occurs once in each image. Note that if regions s_0 and s_3 are important for that semantic class (e.g., sky and sand in a beach class), they will be accounted for in S_a and S_b, regardless of other regions in the image (like s_1 and s_2) that would be taken into account if the strings were compared directly. The system classifies images by first extracting regions and generating a *CRT*. A decision is then made based on the frequencies of the image's CRT with respect to the training set (e.g., the probability that the image is a beach scene given that there is a blue patch followed by a yellow patch). The use of *CRTs* is similar to the work presented in [171] in which region configurations are also used to perform scene classification.

Approaches to perform scene-level classification can be very effective, and thus demonstrate that semantic classification can be achieved using low-level features. Building such classifiers usually requires little user input (e.g., only labeling of the images at the scene level). The accuracy of these techniques, however, is limited by how well the selected features can be used to discriminate between the classes of interest. Additionally, the size and nature of the training (if a training set is required) and test sets directly affect the accuracy.

2.4.1.1 Visual Information and Text

It is also possible to combine textual and visual information to build classifiers. In [243], text classification is performed first, followed by classification using visual features. In [247], textual information augments the result of visual classification (face detection). Lastly, in [206], the benefits of combining visual and textual features were explored for indoor vs. outdoor classification.

In addition to improving performance, text can help in indexing images at higher semantic levels (i.e., levels 5 through 10 in Figure 16). One of the common problems, however, is that the use of both modalities is not always possible because text is not always available. When it is available, it can often be unreliable (e.g., text accompanying web documents in [244]). Additionally, it may come from different sources, and have strong variations in style and quality.

In spite of some of the technical difficulties associated with the use of features from different modalities (e.g., visual and textual), their use is of extreme importance for indexing visual content in terms of semantics (see the news application in [66]). Research in this direction is likely to increase and continue to contribute to better retrieval systems.

2.4.2 The Object Recognition Problem

Object Recognition is perhaps the largest area within Computer Vision. However, it is closely related to the framework of chapter 5 and so in this section we give a brief overview of some techniques for two-dimensional object recognition. A complete review and classification of all the approaches would be a very difficult task. The goal here is to give the reader a general understanding of some of the main techniques. There are many ways in which different techniques can be grouped, and the divisions below are presented to aid in the explanations rather than to classify different approaches. Extensive reviews can be found throughout the computer vision literature (e.g., [222][82]).

First, we give a brief overview of object recognition. Then, we outline important knowledge representation techniques, and discuss the differences between object recognition and VIR.

The goal of object recognition is to determine *which* objects are present in a scene, and *where* they are located [178].

There are many approaches to the object recognition problem in the computer vision literature. In general, however, recognition involves three basic components (Figure 14): (1) the image; (2) the features extracted from the image (e.g., geometric features, etc.); and (3) a decision process based on those features. The features can be pixel intensity values, or complex geometric representations, among others. The decision process depends not only on the features extracted from the image, but also on the type of knowledge the system uses in making a decision. There are many variations in how a decision is made, what knowledge is included, and how such knowledge is represented internally.

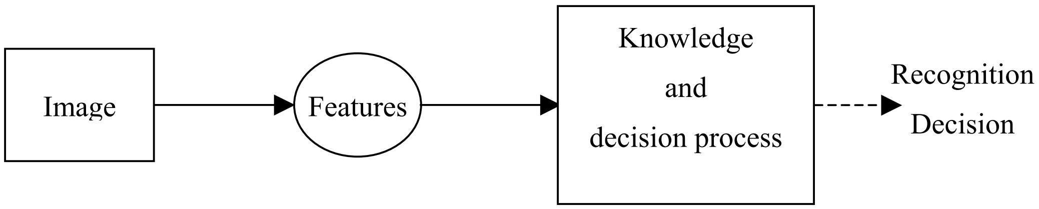


Figure 14. Basic components of an object recognition system.

One possible strategy for recognition is to have a detailed model (e.g., with detailed geometric constraints) for each object (or type of object) to be recognized and to try to match image features to each model. Traditionally, this approach is referred to as *model-based* recognition, described next.

2.4.2.1 Model-based Recognition

In the framework of chapter 4 the user explicitly constructs a model and labels training examples. A *model* is an internal representation for each object that is to be recognized. The recognition process consists of a search for correspondences between components of a model and observations in an image. For example, to recognize a square using the simple model of a square in Figure 15, the object recognition process would entail finding two corners, as determined by the model in the model database.

Given a set of models in a model database, the search strategy used to find a match between a model and the observed features can be classified into either *model-driven* or *data-driven* [119]. In the model-driven approach, a model is selected from the database, and the image is searched in order to find components that match that model. In the data-driven approach, the system picks a component from the image (e.g., one of the lines) and tries to find the model that best matches that component. In both cases, recognition is complete when a correspondence is found between all of the components of a model and the observations in the image.

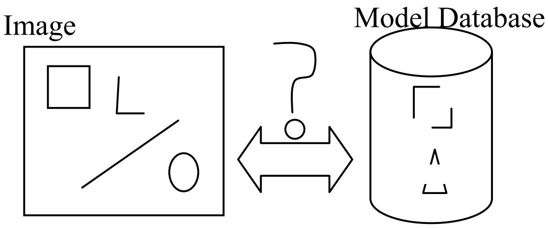


Figure 15. An image and a database of models.

Regardless of whether the recognition strategy is model or data driven (or a combination of both), the recognition process can be formulated as having three stages [118]: *selection*, *indexing*, and *matching*. For example, in the work presented in [118], object recognition is performed by selecting a subset of the image data; indexing (i.e., searching) the object models from a model database; and matching of the models and the image data. Many systems have followed this three-stage process.

As the number of models and the complexity of each model increases, the model indexing process becomes more complex, requiring the use of special structures and techniques for efficient recognition. Efficient indexing of the models is important in supporting scalability of model-based approaches. One technique, the data-driven indexed hypotheses technique [119], provides a data structure for the model-related data so that during recognition, models can be efficiently added to a database, and the search for a model component that best matches a given image feature can be efficiently performed.

In general, many techniques use *hypothesize-and-test* algorithms to find correspondence. A hypothesis is formed first (e.g., there is a square in the image). Then, tests are performed to determine whether that hypothesis is correct or not (i.e., find a correspondence between the model and the elements in the image). As the example suggests, the features used to represent the image, the model used, and the decision process can be treated independently, although they are closely related.

Model Representation

The discussion in section 2.4.2 emphasized that recognition implies the use of knowledge during the process (unknown objects cannot be recognized). In this section, we give a brief overview of some of the major knowledge representation techniques [111] used in object recognition [109]. We argue that even though some of these techniques have not been widely used in object recognition/VIR, their use and importance is likely to increase due to the significance of knowledge representation in future VIR systems. This is specially likely in systems that integrate different types of data (audio, text, video, etc.).

Over the last few years the number of ways in which models can be represented has grown significantly. Some of the earlier techniques discussed here have been applied to recognizing 2D as well as 3D objects, under different types of constraints and for different applications.

Template Matching

One of the simplest ways to model an object is to have a template of a particular view of the object. During recognition, the template is simply swept over the image in an attempt to maximize cross-correlation (e.g., find the best match between the template and the image features). Template matching tries to find the best embedding of a template sub-image to an observed image, over all translations and rotations. In some cases, multiple two-dimensional views of the same object are stored as templates, and the recognition process tries to find the best pixel-level matches between the template and the image.

In the work presented in [120], template matching was used to index news video (e.g., recognize news anchor scenes).

Template matching is simple, but can be computationally expensive, and it is very restrictive in the sense that the templates must closely match the observed objects.

Production Systems

In this paradigm, a model is represented by a set of production ("if-then") rules[11], which are usually constructed manually. During the recognition process, an if-then rule is executed only when the "if" part of the rule is matched. Using this mechanism, the designer of the system can control the way in which the rules are applied, and thus, the search for correspondence between the models and the image components.

Production systems were very successful in the medical domain (text only), where diseases could be diagnosed based on a set of rules obtained from an expert. In the MYCIN system [111], for example, it was found that the production system could actually outperform experts in diagnostic tasks.

The same principle has been applied, less successfully, to recognizing objects in 2D images, in aerial image interpretation and in other tasks. For example, in one of the earliest rule-based image analysis systems [204], semantic labels are assigned to regions in color images of outdoors scenes. The initial image is segmented and a symbolic top-down analysis of the regions is performed using production rules— each region is labeled according to the set of rules. In that system, rules included the following relations: the sky touches the upper edge of

[11] Rules in an expert systems are called production rules because new information is produced when the rules fire. The term production itself is used in psychology to describe the relationship between situations and actions, and more commonly referred to as a rule [111]

the picture; the road touches the lower edge of the picture, etc. The system is capable of recognizing four objects and four sub-objects.

In another early rule-based system [180], a similar approach is used to recognize objects in aerial images. In that system, the number of rules is close to 500. This exemplifies one of the drawbacks of this technique: although rules in these systems are easily understood by humans, as the number of rules grows, the maintenance and understanding of the rules becomes more complex. Generation of the rules is often a difficult task as well. The same authors present a set of tools for knowledge acquisition in that domain in [181]. In spite of some disadvantages, rule-based systems, cane be powerful since they allow modularity, easy expansion, and natural expression (i.e., human readable rules).

Semantic Networks

Representing topological and inclusion relationships with rules tends to be very cumbersome. In contrast, semantic networks allow for richer representations of knowledge (an important advantage of semantic networks). A semantic network is a method of knowledge representation using a graph made up of nodes and arcs where nodes represent objects (and their parts) and arcs represent the relationship between the objects [111]. Examples of relationships include part/subpart, specialization, and adjacency. Little work has been done in object recognition using semantic networks, but their use is likely to gain importance in VIR systems.

Blackboard Systems

The goal of blackboard systems [83] is to integrate knowledge of different "experts" (e.g., a sky recognizer, a grass recognizer) into a single framework. Each expert is called a knowledge source (*KS*), which is basically a procedural module, and the experts act independently but interact and exchange information by using a blackboard. A blackboard usually consists of an area in memory which the different experts can write to and read from. A separate entity, called the scheduler (usually a set of if-then rules), decides when to use each knowledge source. Examples of systems that use this paradigm include [122] and [71].

The main difference between production and blackboard systems is that production systems handle simple if-then declarative statements, whereas in blackboard systems knowledge sources may be complex software modules. The drawbacks, however, are the same: manually constructed rules are difficult to generate and maintain.

An alternative that allows the combination of different experts is belief networks [226]. A belief network represents the joint probability distribution for a set of variables. This is represented by a directed graph in which nodes represent variables and arcs represent conditional dependency. Each variable (node) is associated with a conditional probability table that specifies the conditional distribution of the variable given its immediate parents in the graph. Belief networks can be used to represent causal dependence and to integrate the inputs

[205] of different classifiers (e.g., the value of a variable could be determined by a classifier). In the work presented in [205], different object detectors (i.e., experts) are combined to classify images. The probability that an image is an outdoor image, for example, is computed from the probability scores produced by different classifiers (for example, a classifier that detects sky, a classifier that detects vegetation, etc.). In this case, the "rules" correspond to a network of nodes, and the directed arcs determine the order of evaluation. One of the advantages is that the network can in principle be learned automatically from the training set provided by the user, in other words, the network structure does not have to be manually constructed. In practice, however, this is not the case: finding an optimal network structure given a data set is an NP-complete problem, and suboptimal network structures often have very poor performance (in the example above, the learning algorithm could produce a network where the outdoor/indoor variable is a precursor of both sky and vegetation nodes). The structure, then, is usually constructed manually, but the conditional probability tables associated with each node are computed automatically.

Transform Methods

As discussed earlier, we can characterize the object recognition problem as a classification problem. The majority of model-based systems have relied heavily on shape-based descriptions to represent the geometry of the objects. The goal in many of those systems is to recognize objects from any viewing angle, both in two and three-dimensional scenarios.

A different set of techniques, however, has focused on *appearance matching*. Under this paradigm, object models are constructed as collections of images of the object in various positions, orientations, and lighting conditions. Features are extracted from the images in one domain (e.g., spatial) and in some cases transformed to other domains (e.g., frequency). A very successful technique consists of representing images using an eigen (function, image) decomposition, and performing classification using the decomposition coefficients. In many of these techniques, though, either the existence of a single object on a uniform, known background, or good automatic segmentation results are assumed (e.g., [262][194]).

2.4.3 Object Recognition in Specific Domains

Object recognition in specific domains can be formulated the same way as the traditional object recognition problem: determining *which* objects appear in an image and *where* they appear.

Several approaches have been devised to perform object-level classification[12]. In the Body-plans approach presented in [113], specialized filters (e.g., skin for humans, hide for horses) are first applied to the image to detect naked people or horses. Once the filters detect the presence of skin/hide, the extraction of cylinder-like primitives occurs, and their configurations are matched against a Body-plan. A Body-plan is a sequence of groups constructed to mirror the layout of body segments in people and animals. For example, a horse has four legs, a trunk, a neck, a head, etc. In order to detect a person or horse, the system tries to construct a sequence of groups according to the Body-plan. In the case of a horse, it would try to collect body, neck, and leg segments. Then, it would try to construct body-neck or body-leg pairs, and so on. Note that each model representation includes a combination of constraints on color and texture, and constraints on geometric properties, such as the structure of individual parts and the relationships between parts. Each segment (e.g., body, neck) is found by a classifier (which, in this case, is a routine that decides if a segment is present or not). The naked-people Body-plan is built manually and, although a learning component is used in the horse classifier, its topology is given in advance. Learning is achieved by constructing an augmented feature vector that all

[12] In some cases, we will use the word *detection*. *Classification* assigns an object to a category, whereas *detection* determines the presence of an object.

classifiers (for the segments) use for training. Once individual classifiers are learned, their decisions are used to obtain sub-classifiers.

One disadvantage of this approach is that various components of the models are built manually. The filters for skin and hide, for example, are very specific to the naked people/horses application, and applying the same techniques to other classes would require building new filters. Another drawback of this approach is that it is based on cylinder-like primitives – detecting them is often a difficult task due to occlusion, changes in lighting, etc. (see section 2.4.3). The Body-plans technique, however, demonstrates that models for object recognition can be effectively used in Visual Information Retrieval applications. It also stresses the importance of including real-world constraints on the models (e.g., body-leg pairs), and the possibility of combining manually constructed components with automatic ones. The framework we present in chapter 4 addresses some of the disadvantages of the Body-plans technique.

Other approaches (e.g., detection of faces in [269]) have been successfully applied to VIR. The WebSeer system [114], for example, uses the face detection technique developed in [220] to index images collected from the World Wide Web. In Section 2.4.3 we explore the area of object recognition and discuss how it relates to VIR.

2.4.4 Differences Between Object Recognition and VIR

Although research in object recognition goes back more than 30 years, most of the work in this area has focused on the recognition of simple isolated objects in highly constrained environments, where factors such as viewpoint, occlusion, scale, lighting, noise, sensor quality (e.g., cameras, etc.), are carefully controlled. In Visual Information Retrieval, however, few of theses constraints hold; many of the traditional object recognition techniques have limited applicability for VIR.

While most of the work on object recognition has focused on use of knowledge-based systems for finding correspondence (e.g., model-based recognition), the majority of VIR techniques have focused on similarity: images are searched or classified according to how similar they are to a set of examples. Although this is rapidly changing, little domain knowledge is usually included in VIR systems.

In object recognition, the goal has often been to *precisely* determine which objects appear in a scene and where they are located. In an industrial inspection system or a robotics system, for example, there is a computable cost for errors in recognizing an object and its exact location, and requirements are often posed in terms of system accuracy. In contrast, in most VIR applications, the accuracy requirements vary widely, and performance is measured differently (in terms of precision/recall).

Another major difference is the involvement of the user: in traditional object recognition, the goal is to build systems that work without user input. In VIR, humans are the final users of the systems developed. This has many implications, not only in terms of performance, but also in terms of the subjectivity involved.

2.4.5 The New Challenges

In spite of the recent advances in object recognition, and the successful application of techniques in certain domains, many of the recognition systems developed to date have serious limitations if they are to be applied to the general problem of Visual Information Retrieval. Of course, the success of the techniques depends largely on the contents of the database, the user, and the purpose. A database of simple objects with controlled lighting conditions and uncluttered background, for example, can benefit greatly from many of the traditional object recognition approaches.

Traditional object recognition techniques, however, face new challenges when applied in the context of VIR. We outline some of the major difficulties, which cannot usually be controlled in VIR applications.

- Illumination: the appearance of objects can vary significantly under different lighting conditions.
- Extraneous features: shadows and specular reflections make feature

extraction more difficult.

- Occlusion: parts of objects are often covered by other objects, complicating the recognition process.
- Viewpoint: large variations of viewpoint occur.
- Scale: objects and textures appear in many different sizes.
- Noise: color variations and other imperfections.
- Clutter: presence of numerous unmodeled objects in the scene.
- Background: foreground-background separation is non-trivial.
- Types of objects: rigid, deformable, and flexible objects. Existence of non-traditional objects, such as the sky, water, and trees.
- Types of models: geometric models typically used in object recognition do not necessarily coincide with "models" used by humans. For example, a car jack is usually thought of in terms of its function, not its geometric characteristics. Novel model representations may be necessary.
- Ability to adapt: there is a large number of objects in realistic scenes, therefore it is desirable to have systems that can learn to detect new objects (instead of having an expert construct a detector for each object), and easily adapt to variations in realistic scenes (lighting, etc.).

In order for object recognition approaches to be applicable to general VIR (face detection is a good example), they must more closely resemble general vision systems [82]. They should be able to adapt to different conditions (e.g., lighting, occlusion, etc.), make more use of world knowledge, and be suitable for application under fewer constraints. Those three aspects-*adaptability*, *knowledge*, and *constraints* represent the major limitations of current systems. For example, most of the systems described in [82], "know" a limited set of objects: 4 objects and sub-objects in outdoor scenes [204], area classes (vegetation, buildings, etc.) [196], airplanes [84], lamps, desks, etc. These systems use limited sets of objects, top-down interpretation, and rely heavily on prediction. Although the inclusion of specific knowledge is beneficial, future systems for VIR must be scalable in the number of objects they can recognize, and must be easily adaptable to the many variations encountered in domains with few constraints.

2.5 SUMMARY

In this chapter we gave an overview of basic concepts and existing approaches related to the three *areas* discussed in chapter 1: understanding of visual content and users, construction of flexible computational methods that learn, and integration of generic visual detectors in solving practical tasks in specific domains.

In particular, we presented basic visual information concepts, discussed the main components of VIR systems, and gave an overview of object recognition highlighting the advantages and limitations of different techniques. We made distinctions between the interface and indexing components of VIR systems, discussed the differences between various interface modalities, and outlined the benefits of different indexing mechanisms.

Our overview of some of the major object recognition strategies was presented with an emphasis on its relation to VIR. We identified the differences between traditional object recognition and object recognition for VIR, and outlined the challenges that object recognition techniques face if they are to be applied to the general VIR problem.

The topics covered in this chapter are very important in addressing the problem of indexing and organization of visual information through user interaction at multiple levels. The different types of indexing schemes are clearly related to the way that information can be accessed, as are the different query interfaces. Understanding the benefits and limitations of these current approaches is fundamental to the work presented in the rest of the thesis and to the development of the visual information indexing systems of the future.

3 THE MULTI-LEVEL INDEXING PYRAMID

3.1 INTRODUCTION

In this chapter we present a novel conceptual framework for the classification of visual attributes.

The *Multi-Level Indexing Pyramid* we introduce classifies visual attributes into ten levels stressing the differences between *syntax* and *semantics*. Our discussions show that different techniques, discussed in chapter 2, are useful for different purposes, and that the level of indexing depends on the application. For example, it makes no sense to look for pictures of Bill Clinton using texture. Similarly, there are many instances in medical imaging in which textual search is not nearly as powerful as query-by-example.

We present various experiments[13], which address the Pyramid's ability to achieve the following tasks: (1) classification of terms describing image attributes generated in a formal and an informal description task, (2) classification of terms that result from a structured approach to indexing, (3) guidance in the indexing process. Several descriptions, generated by naïve users and indexers were used in experiments that included two image collections: a random web sample, and a set of news images. To test descriptions generated in a structured setting, an *Image Indexing Template* [50] was also used. The results of the experiments suggest that the Pyramid is conceptually robust (i.e., can classify a full range of attributes) and that it can be used to organize visual content for retrieval, to guide the indexing process, and to classify descriptions obtained manually and automatically.

The pyramid has several important applications supported by our experiments. It can be used to classify visual descriptors generated manually or automatically, to guide the manual annotation process or the creation of automatic feature extraction algorithms, and to improve retrieval in visual information systems by eliminating the ambiguity between attributes that could refer to the image at different levels (e.g., color blue vs. emotion blue).

[13] The experiments presented in this chapter were performed jointly with Corinne Jorgensen of the State University of New York at Buffalo and Ana B. Benitez from Columbia University.

3.1.1 Related Work

Discourses on imagery have arisen in many different academic fields. Studies in *art* have focused on interpretation and perception [1][9], aesthetics and formal analysis [4], visual communication [12], levels of meaning [16], etc. Studies in *cognitive psychology* have dealt with issues such as perception [26][27][34], visual similarity [36], mental categories (i.e., concepts) [23], distinctions between perceptual and conceptual category structure [24][25][33], internal category structure (i.e., levels of categorization) [29][30][32][35][28], etc. In the field of *information sciences*, work has been performed on analyzing the subject of an image [46][59][61][62], and on general issues related to manually assigning indexing terms to images [42][56][58][63]. Work has also been done on analyzing the types of attributes that can be used to describe images [47][48]. Two sets of proposed attributes have been widely disseminated: The *Dublin Core* [39] and the *VRA Core* [64]. The Art Information Task Force has also proposed the *Categories for the Description of Works of Art* [43]. Work in that field has also been done in classification [49][56] and query analysis [40][41]. In addition, there have been efforts to construct thesauri (often called ontologies or taxonomies) [57] and other tools to facilitate manual indexing tasks. *TGM I/II* [57], for example, consists of a list of terms and relationships specifically constructed for describing visual information, and the template in [37] consists of a list of description categories so that people manually indexing images can “fill in” the template in a structured, consistent way. Other related work includes [55] and [50], among others.

There have also been many efforts related to the organization of multimedia data. Some of that work includes [78][127][131][173][248], and [215]. The MPEG-7 standard [176], for example, standardizes a set of descriptors for multimedia information, some of which focuses exclusively on visual content [192].

The *Multi-level Indexing Pyramid* was proposed to MPEG-7 [76][77][143]. Many of the concepts and components in the pyramid are included in the standard (e.g., *syntax, semantics, abstract concepts*, etc.).

The authors of [187][188] have performed studies to investigate the semantic categories that guide the human perception of image similarity. The authors have suggested several levels of semantics (abstract, semantic templates, semantic indicators, low-level primitives and their perceptual groupings).

3.1.2 Outline

In section 3.2, we discuss conceptual structures to classify visual attributes into different levels. In section 3.3 we describe how the pyramid can be applied. In section 3.4 we present experiments to test the multi-level indexing pyramid. We discuss our results in section 3.5 and summarize in section 3.6.

3.2 INDEXING VISUAL INFORMATION AT MULTIPLE LEVELS

In this section, we focus on the problem of multiple levels of description when indexing visual information. We present a conceptual framework, which draws on concepts from the literature in diverse fields, such as cognitive psychology, library sciences, art, and Visual Information Retrieval. We make distinctions between *visual* and *non-visual* information and provide the appropriate structures. We describe a ten-level *visual* structure, which provides a systematic way of indexing visual information based on *syntax* (e.g., color, texture, etc.) and *semantics* (e.g., objects, events, etc.), and includes distinctions between *general concepts* and *visual concepts*. We define different types of relations (e.g., *syntactic, semantic*) at different levels of the *visual* structure, and also use a *semantic information table* to summarize important aspects of an image[14].

One crucial aspect of building a VIR system is understanding the data that will be included in the system so that it can be appropriately indexed. Appropriate indexing of visual information is a challenging task because there is a large amount of information present in a single image (e.g., *what* to index?), and different levels of description are possible (e.g., *how* to index?). Consider, for

[14] For simplicity, throughout the chapter we use the term image to refer to visual information in general. In section 3.3 we discuss some of the issues that may arise if the structure is applied to video.

example, a portrait of a man wearing a suit. It would be possible to label the image with the terms "suit" or "man". The term "man", in turn, could carry information at multiple levels: *conceptual* (e.g., definition of man in the dictionary), *physical* (size, weight) and *visual* (hair color, clothing), among others. A semantic label, then, implies explicit (e.g., the person in the image is a man, not a woman), and implicit or undefined information (e.g., from that term alone it is not possible to know what the man is wearing).

The research on visual information that has been carried out in different fields shows that indexing such information can be particularly complex for several reasons. First, visual content carries information at many different levels (e.g., *syntactic*: the colors in the image; *semantic*: the objects in the image). Second, descriptions of visual content can be highly subjective, varying both across *indexers* and *users*, and for a single user over time. Such descriptions depend on other factors that include, for example, the indexer's knowledge (e.g., art historian), purpose of the database (e.g., education), database content (e.g., fine art images; commercial images), and the task of the user (find a specific image or a "meaningful" image).

The structure presented in this chapter places state-of-the art content-based retrieval techniques in perspective, relating them to real user-needs and research in other fields. Using structures such as these is beneficial not only in terms of understanding the users and their interests, but also in characterizing the Visual

Information Retrieval problem according to the levels of description used to access visual information.

Visual And Non-Visual Content

The first step in creating conceptual indexing structures is to make a distinction between *visual* and *non-visual* content. The *visual content* of an image corresponds to what is directly perceived when the image is observed (the lines, shapes, colors, objects, etc). The *non-visual content* corresponds to information that is closely related to the image, but that is not present in the image. In a painting, for example, the price, current owner, etc. belong to the *non-visual* category. Next we present a novel indexing structure for the *visual content* of the image and for completeness follow with a structure for *non-visual* content. Non-visual content constitutes the metadata that accompanies visual information.

3.2.1 Visual content

Our visual structure contains ten levels, all of which are obtained from the image alone: the first four refer to *syntax*, and the remaining six refer to *semantics*. In addition, levels one to four are directly related to *percept*, and levels five through ten to *visual concept* (see chapter 2). While some of these divisions may not be strict, they should be considered because they have a direct impact on understanding *what* users are searching for and *how* they try to find it. They also emphasize the limitations of different indexing techniques (manual and automatic) in terms of the knowledge required. An overview of the structure is given in Figure 16.

A pyramid is an immaterial structure built on a broad supporting base and narrowing gradually to an apex[15]. We choose this structure because visually it provides an intuitive representation of the knowledge required to perform classification at different levels. In our *Multi-level Indexing Pyramid* more knowledge is necessary to index at the lower levels than at the higher levels. For example, more knowledge is required to recognize a specific scene (e.g., Central Park in New York City) than to recognize a generic scene (e.g., park). There may be exceptions, however. An average observer may not be able to determine the technique or materials (e.g., oil, watercolor) that were used to produce a painting, but an expert would. In general the structure implies that computational analysis at the lower levels of the pyramid requires more complex knowledge. Face recognition (*specific object*) versus face detection (*generic object*) constitutes just one example.

[15] Definition from Merriam-Webster dictionary.

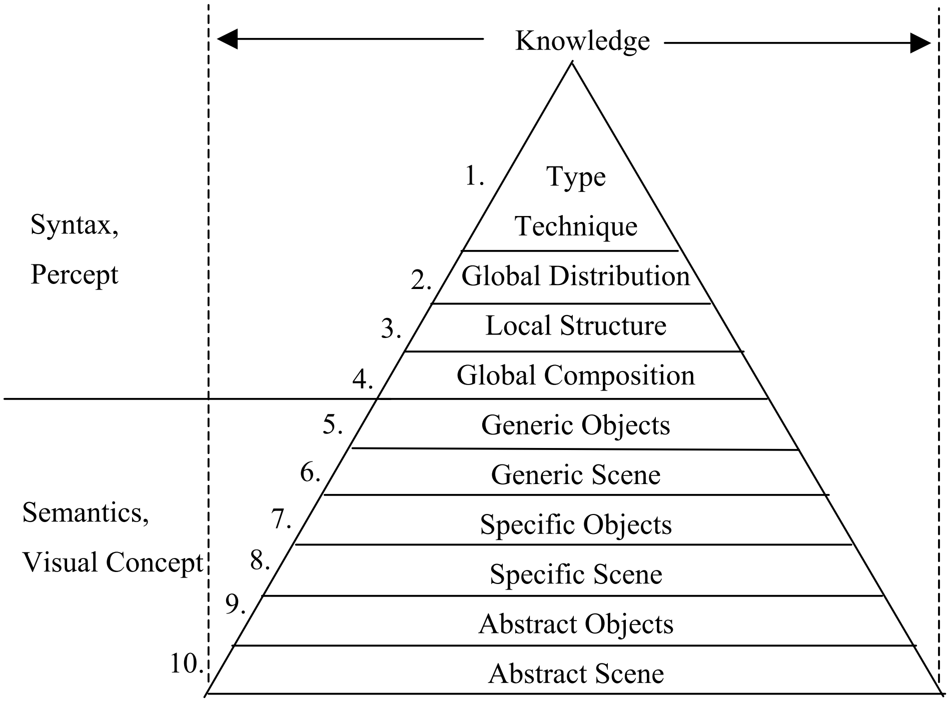


Figure 16. The indexing structure is represented by a pyramid.

The way we order the levels follows this rationale and therefore, it makes more sense to place the higher semantic levels at the bottom. An alternative way to think of the structure is as an iceberg: the simplest syntactic attributes at the top represent only "the tip of the iceberg", the earliest, most obvious, or most superficial manifestation of some phenomenon[16].

Each level, which is independent of the other levels, can be used to classify an image's attributes, effectively providing access points to the image at different

[16] Definition from Merriam-Webster dictionary.

levels. For example, an image can be searched based on similarity at the type/technique level, or on the generic objects it contains. It is important to keep in mind that the levels chosen for a particular application will depend on the data being indexed and on the way it is used.

The pyramid does not depend on any particular database model, visual information type, user type, or purpose; therefore, not every level of the structure would be necessary in every case. Different projects demonstrate that while many researchers are addressing various levels of the "puzzle," there has not heretofore been a conceptual structure which can unite these diverse efforts and demonstrate the relationships among the pieces of this puzzle.

We demonstrate the applicability of the structure to a wide range of attributes, and show that the pyramid can also facilitate searching by disambiguating among attributes that could appear at several different levels of the structure.

Three main factors entered into the construction of the proposed model: (1) range of descriptions; (2) related research in various fields; and (3) generality. In considering the range of descriptions, the focus was only on *visual content* (i.e., any descriptors stimulated by the visual content of the image or video in question; the price of a painting would <u>not</u> be part of *visual content*). Since such content can be described in terms of *syntax* or *semantics* the structure contains a division that groups descriptors based on those two categories. This division is of paramount importance, particularly when we observe research in different fields. Most of the work on *Visual Information Retrieval*, for example, supports syntactic-level indexing, while work in art places strong emphasis on composition (i.e., relationships between elements) both at the syntactic (i.e.., how colors, lines, and patterns are laid out) and semantic levels (i.e., the meaning of objects and their interactions). Most of the work in information science, on the other hand, focuses on semantics. The structure was developed based on research and existing systems in different fields.

3.2.1.1 Type/Technique

At the most basic level, we are interested in the general visual characteristics of the image or the video sequence. Descriptions of the type of image or video sequence or the technique used to produce it are very general, but are very frequently used and are therefore of great importance. Images, for example, may be placed in categories, such as painting, black and white photograph, color photograph, and drawing. Although *Type/Technique* attributes are not strictly syntactic, they do directly impact the visual elements. For simplicity, therefore, it is appropriate to group this level with the other levels that refer to syntax. Related classification schemes at this level have been done conceptually in [127][173], and automatically in WebSEEk [244].

In the case of digital photographs, the two main categories could be color and grayscale, with additional categories/descriptions, such as number of colors, compression scheme, resolution, etc. We note that some of these may have some overlap with the non-visual indexing aspects described in Section 3.2.3. Figure 17a (page 85) shows an interesting example.

3.2.1.2 Global Distribution

The *type/technique* in the previous level gives general information about the visual characteristics of the image or the video sequence, but gives little information about the visual content. An image (or video sequence) is characterized at the *global distribution* level using low-level perceptual features, such as spectral sensitivity (color), and frequency sensitivity (texture). Individual components of the content are not processed at this level: the low-level perceptual features are extracted from the image as a whole. *Global distribution* features may include global color (measured, for instance, as dominant color, average color or color histogram), global texture (using one of the descriptors such as coarseness, directionality, contrast), global shape (e.g., aspect ratio), and, for video data, global motion (e.g., speed, acceleration, and trajectory), camera motion, global deformation (e.g., growing speed), and temporal/spatial dimensions (e.g., spatial area and temporal dimension), among others. The example in Figure 17b shows images of two buttons that have similar texture/color. Notice that the global distribution attributes of the example could be quite useful for retrieving visually similar images, but they would probably be useless for retrieving images containing specific objects.

Although some global measures are less intuitive than others (e.g., it is difficult for a human to imagine what the color histogram of an image looks like), these global low-level features have been successfully used in various Visual Information Retrieval systems to perform query by example (QBIC [200], WebSEEk [244], Virage [72]) and to organize the contents of a database for browsing (see discussions in [92]). An interesting comparison of human and machine assessments of image similarity based on global features at this level can be found in [179].

3.2.1.3 Local Structure

Local Structure is concerned with image components. At the most basic level, those components result from low-level processing and include elements such as *dots, lines, tone, color*, and *texture*, extracted from the images. In the Visual Literacy literature [12], some of these are referred to as the "basic elements" of visual communication and are regarded as the *basic syntax symbols.* Other examples of local structure attributes are temporal/spatial position, local color, local motion, local deformation, and local shape/2D geometry. Figure 17c (page 85) shows images in which attributes of this type may be of importance. In the x-ray image of a tooth, the shape and location of a cavity are of great importance. Likewise, identification of local components in microscopic images can be more important than the image as a whole.

Such elements have also been used in Visual Information Retrieval systems, mainly in query by user sketch interfaces, such as those in [136][89][217] and [246]. The concern here is not with objects, but rather with the basic elements that represent them and with combinations of such elements: four lines, for example, form a square. In that sense, we can include here some "basic shapes" such as circle, ellipse and polygon.

3.2.1.4 Global Composition

Local Structure is characterized by basic elements. *Global Composition* refers to the arrangement, or spatial layout of these basic elements. Traditional analysis in art describes composition concepts, such as *balance*, *symmetry*, *center of interest* (e.g., center of attention or focus), leading line, viewing angle, etc. [1]. At this level, however, there is no notion of specific objects; only *basic elements* (i.e. dot, line, etc.) or groups of basic elements are considered. In that sense, the view of an image is simplified to an entity that contains only basic syntax symbols: an image is represented by a structured set of lines, circles, squares, etc. Figure 17d presents images with similar composition (both images have objects in the center, and the leading line is diagonal). The composition of the images in Figure 17f is also similar, where the leading line is horizontal. A composition similarity-search was implemented in Virage [72].

3.2.1.5 Generic Objects

In the first four levels, the emphasis is on the perceptual (percept-related) aspects of the image, thus, no knowledge of actual objects is required (i.e. recognition is not necessary) to perform indexing, and automatic techniques rely only on low-level processing. While this is an advantage for automatic indexing and classification, studies have demonstrated that humans mainly use higher-level attributes to describe, classify and search for images [47][48][49]. Objects are of particular interest, and they can also be placed in categories at different levels: an apple can be classified as a fruit, as an apple, or as a Macintosh apple.

When referring to *Generic Objects,* we are interested in what Rosch [32] calls the *basic level categories*: the level at which there are attributes common to all or most members of a category— the highest level of abstraction at which clusters of features are assigned to categories. In other words, it is the level at which there are attributes common to all or most members of a category. In the study of art, this level corresponds to *pre-Iconography* [16], and in information sciences [61] is called the *generic of* level. Only general everyday knowledge is necessary to recognize the objects at this level. When viewed as a Generic Object, a Macintosh apple would be classified as an apple: that is the level at which the object's attributes are common to all or most members of the category. In the experiments reported in [32], for example, the authors found that a large number of features were listed as common to most elements of a basic category such as "apple", whereas few if any features were listed as common to elements in a superordinate category one level higher such as "fruit". The general guideline is that only everyday knowledge is required at this level. Techniques to automatically detect objects at this level include a significant amount of work in object recognition (see Section 2.4.2), and techniques in VIR (e.g., [113][145]).

A possible difference between our definition and those in [16][61] is that our *generic objects* include objects that may traditionally not be considered objects (e.g., the sky or the ocean). Examples of generic objects "car," and "woman" are shown in Figure 17e. Figure 17g shows a "building," but note that in that figure the name of the building is used, so that particular attribute is a specific object attribute.

3.2.1.6 Generic Scene

Just as an image can be indexed according to the individual objects that appear in it, it can also be indexed as a whole based on the set of all of the objects it contains (i.e., what the image is of). Examples of scene classes include city, landscape, indoor, outdoor, still life, portrait, etc. Some work in automatic scene classification has been performed by [254][264][206] and studies in basic scene categories include [35][29].

The guideline for this level is that only general knowledge is required. It is not necessary to know a specific street or building name in order to determine that it is a city scene, nor is it necessary to know the name of an individual to know that it is a portrait. Figure 17f shows two images whose attributes correspond to generic scene. Other generic scene attributes for the same pair of images could be "mountain scene", "beach", etc.

3.2.1.7 Specific Objects

Specific Objects can be identified and named. Shatford refers to this level as *specific of* [61]. Specific knowledge of the objects in the image is required, and such knowledge is usually objective since it relies on known facts. Examples include individual persons (e.g., Bill Clinton in Figure 21), and objects (e.g., "Alex" and "Chrysler building" in Figure 17g). In [247], automatic indexing at this level is performed: names in captions are mapped to faces detected automatically.

When observing the difference between the generic and specific levels, it is important to note that there are issues of indexing consistency that should be taken into account. For example, assume we have an image of a Siamese cat named Fluffy. Indexer A may decide to label this object as "generic object cat", "specific object Siamese cat". Indexer B may decide to label this object as "generic object cat", "specific object Fluffy". Indexer C, on the other hand, may decide to label the object as "generic object Siamese cat", "specific object Fluffy". All three approaches are correct since the *relationship* between the *generic* and *specific* labels is maintained: the specific description is more specific than the generic one in all three cases. The level of specificity chosen, then, depends on the application. In many indexing systems in information sciences, issues of indexer consistency are addressed by the use of templates (e.g., indexers fill in the fields of a template specially designed for the application or type of data) and vocabularies (e.g., indexers can only use certain words). In spite of these mechanisms, however, indexer consistency is always an issue that should be addressed. In Section 3.4 we discuss some experiments comparing the use of the pyramid with the use of a template developed specifically for indexing images.

3.2.1.8 Specific Scene

This level is analogous to *Generic Scene*, but specific knowledge about the scene is used. A picture that clearly shows the Eiffel Tower, for example, can be classified as a scene of Paris (see Figure 17h). The other image in the same figure shows a similar example for Washington D.C.

3.2.1.9 Abstract Objects

At this level, specialized or interpretative knowledge about what the objects *represent* is applied. This is related to *Iconology* in art [16], or the *about* level presented in [61] (i.e., what the object is about). This is the most difficult indexing level since it is completely subjective, and assessments among different users vary greatly. The work in ref. [49] showed that users sometimes describe images in affective (e.g. emotion) or abstract (e.g. atmosphere, theme) terms. Examples at the abstract scene level include sadness, happiness, power, heaven, and paradise. For example, a woman in a picture may represent anger to one observer, but pensiveness to another observer. Other examples of abstract object descriptions appear in Figure 17i as "arts" and "law".

3.2.1.10Abstract Scene

The *Abstract Scene* level refers to what the image as a whole *represents* (i.e., what the image is about). It was shown in [48], for example, that users sometimes describe images in affective (e.g., emotional) or abstract (e.g., atmospheric, thematic) terms. Examples at the abstract scene level include sadness, happiness, power, heaven, and paradise. Examples of abstract scene descriptions for the images in Figure 17j are "Agreement", and "Industry".

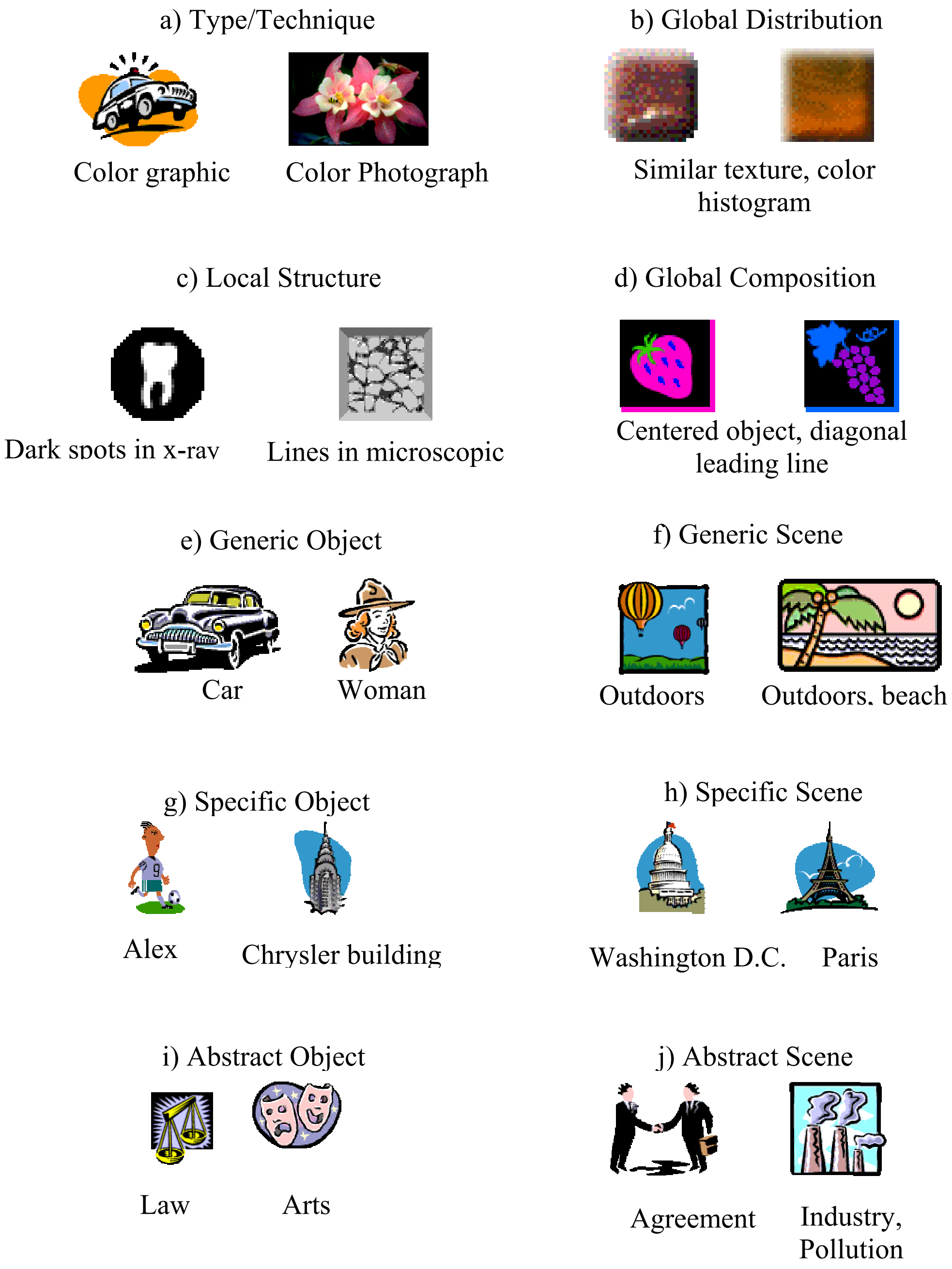


Figure 17. Example images for each level of the visual structure presented. Each image, or part of an image can be described at any of the levels of the pyramid.

3.2.2 Visual Content Relationships

As shown in Figure 18, the structure presented can be used to classify elements[17] within each level according to two types of relationships: *syntactic* (i.e., related to perception) and *semantic* (i.e., related to meaning). *Syntactic* relationships can occur between elements at any of the levels of the pyramid shown in Figure 16, but *semantic* relationships occur only between elements of levels 5 through 10. Semantic relationships between syntactic components could be determined (e.g., a specific combination of colors can be described as warm [98]), but we do not include them in our analysis.

Syntactic relationships include *spatial* (e.g., next to), *temporal* (e.g., before), and/or *visual* (e.g., darker than) relationships. Elements at the *semantic* levels (e.g., objects) of the pyramid can have both *syntactic* and *semantic* relationships (e.g., two people are next to each other, and they are friends). Additionally, each relationship can be described at different levels (*generic, specific,* and *abstract*). Figure 19, shows an example of how relationships at different levels can be used to describe an image— relationships between objects can be expressed at the syntactic and semantic levels.

[17] We use the word element here to refer to any image component (e.g., dot, line, object, etc.), depending on the level of analysis used.

We note that relationships that take place between the *syntactic* levels of the *visual structure* can only occur in 2D space[18] since no knowledge of the objects exist (i.e., relationships in 3D space cannot be determined). At the *local structure* level, for example, only the basic elements of visual literacy are considered, so relationships at that level are only described between such elements. Relationships between elements of levels 5 through 10, however, can be described in terms of 2D or 3D.

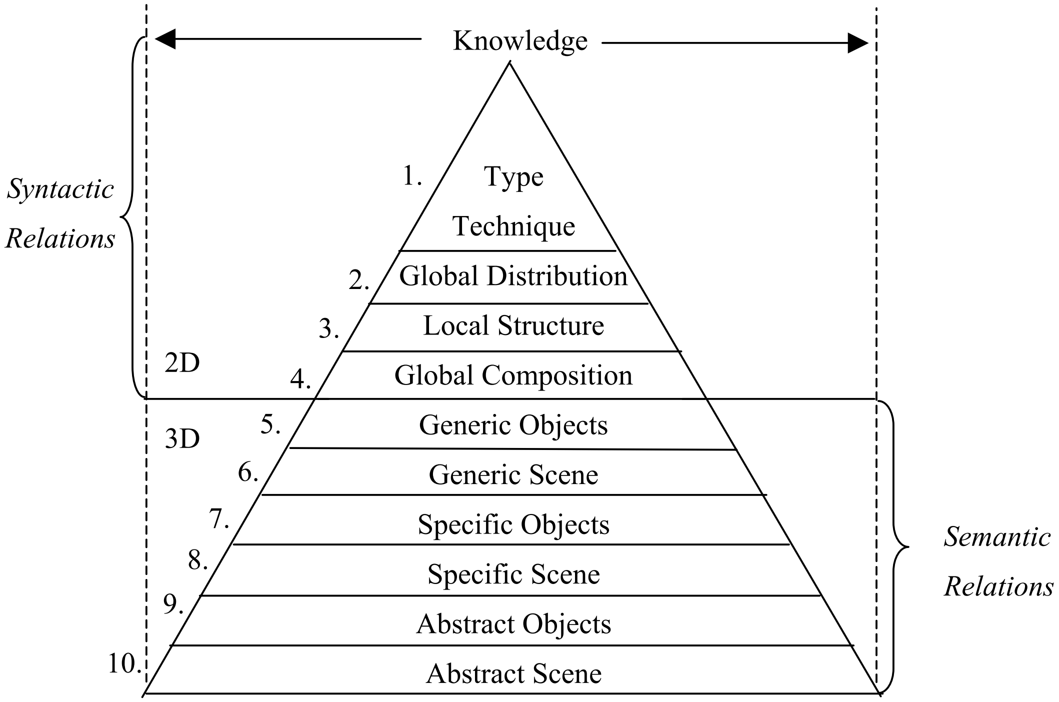


Figure 18. Relationships are based on the *visual structure*

[18] In some cases, we could have depth information associated with each pixel, without having knowledge of the objects. Here we make the assumption that depth information is not available.

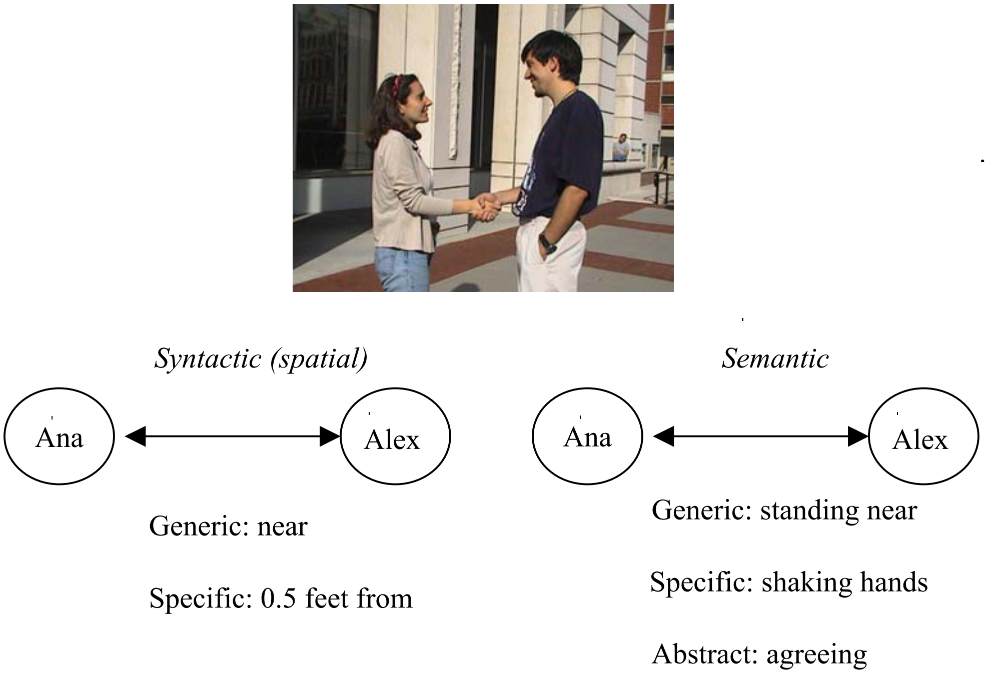


Figure 19. *Syntactic* (*spatial*) (left) and semantic (right) relationships that could describe this image.

Following the work of [128], we divide *spatial relationships* into the following classes: (1) *topological* (i.e., how the boundaries of elements relate) and (2) *orientational* (i.e., where the elements are placed relative to each other). Topological relationships include near, far, touching, etc., and orientation relationships include diagonal to, in front of, etc.

Temporal relationships connect elements with respect to time (e.g., in video these include before, after, between, etc.), while *visual relationships* refer only to visual features (e.g., bluer, darker, etc.). A more detailed explanation of relationships is provided in [76]. These relationships are included in the MPEG-7 standard [176].

3.2.3 Non-visual content

The focus of our work is on visual content. For completeness, however, we briefly describe some aspects of non-visual content.

Non-visual information (Figure 20) refers to information that is not directly depicted in the image, but is associated with it in some way. While it is possible for non-visual information to consist of sound, text, hyperlinked text, etc., our goal here is to present a simple structure that gives general guidelines for indexing.

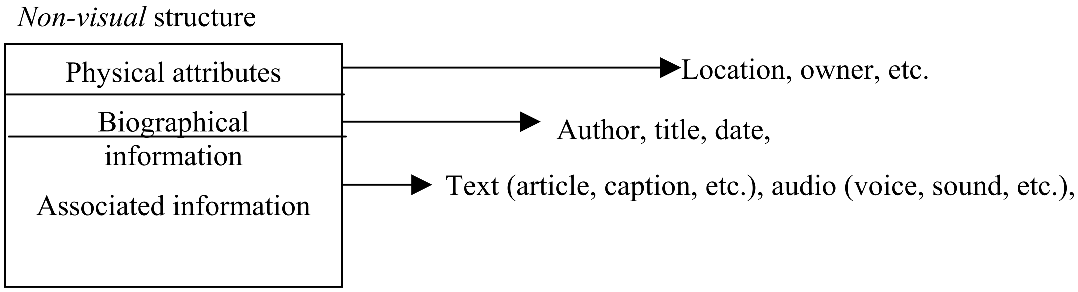


Figure 20. Non-visual information.

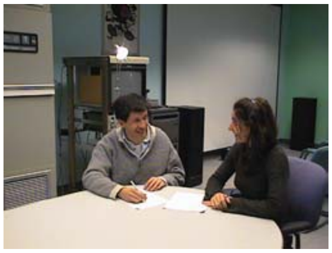
Often, there is *biographical information* including author, date, title, material, etc. This information may relate to the image as a whole, or any of the items contained within the image.

Associated Information is directly related to the image in some way and may include a caption, article, a sound recording, etc. *Physical attributes* simply refer to those that have to do with the image as a physical object. These include location of the image, location of the original source, storage (e.g., size, compression), etc. As an alternative to our division of non-visual attributes, similar attributes in [61] were divided into biographical and relationship attributes.

3.2.3.1 Semantic Information Table

Following the work of [61], we define a *Semantic Information Table* to gather high-level information about the image (Figure 21). The table can be used for individual objects, groups of objects, the entire image, or parts of the image. In most cases both *visual* (i.e., levels 5 through 10 in Figure 16) and *non-visual* information (i.e., associated information in Figure 20) contribute to populating the table. Although the way in which visual/non-visual content contributes to populate the fields in the *semantic table* may vary depending on the specific image, the table does help answer questions such as: "What is the subject (person/object, etc.)? What is the subject doing? Where is the subject? When? How? Why?"

While the table provides a compact representation for some information related to the image, it does not replace the indexing structures presented in previous sections. It does point out, however, that the pyramid can be applied to metadata as well. In the experiments of section 3.4, for example, we use the pyramid to classify attributes that are generated manually.



	Specific	Generic	Abstract
Who	A. Jaimes, Ana	Man, woman	Happy
What action	Discussing MPEG-7 Research	Meeting	
What object	Monitor VR3230	Video Monitor	Hi-tech

Figure 21. Visual and non-visual information can be used to semantically characterize an image or its parts.

3.3 APPLYING THE PYRAMID

The pyramid has several important applications. It can be used to classify visual descriptors generated manually or automatically, to guide the manual annotation process or the creation of automatic feature extraction algorithms, and to improve retrieval in visual information systems by eliminating the ambiguity between attributes that could refer to the image at different levels (e.g., color blue vs. emotion blue).

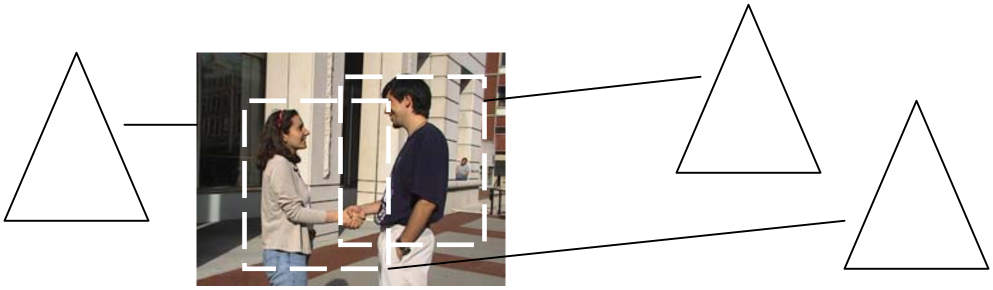


Figure 22. Recursive application of the pyramid.

The *Multi-level Indexing Pyramid* can be applied to any type of image. In fact, it can be applied recursively as illustrated in Figure 22. Any portion of the image, of arbitrary shape, can be described by any number of attributes, which can be classified using the pyramid.

Specific object attributes (placement, texture, size, etc.) are syntactic. The pyramid's specific object level is semantic. The specific objects (e.g., Bill Clinton) have syntactic attributes, but levels 5-10 deal only with semantics, so to determine the position/other of a specific object like Bill Clinton, a local structure (syntactic) attribute would be used. In other words, the specific object attributes in the template are mapped to the syntactic local structure level, not the semantic specific object.

The pyramid can also be used to classify visual attributes in video. It can be applied to the entire video or to arbitrary sections of the video. The main issue in the application of the pyramid to video is the persistence of the attributes. Attribute values will change over time possibly changing the level of description they belong to. Consider an attribute that describes an object in an image that contains several objects. If the camera zooms in so that the object being described now occupies the entire frame, the attribute may in effect be describing the scene and no longer just an object in the scene. In such case, the attribute for the object has become a scene level attribute instead of an object level attribute. It would make more sense, therefore, to have two separate descriptions for the video shot, one *before* the zoom and one *after* the zoom. In other words, the same classification of attributes can be applied to several frames in a video as long as the attribute values do not change significantly. If they do the video or shot can be divided and the pyramid can be used to classify the attributes of the individual parts.

3.4 EXPERIMENTS

Recall that in chapter 2 we defined an index as the data that provides the access point to the information in a database. In the experiments that follow, those indices are created manually; and indexing refers to manually assigning textual attributes to images (i.e., textual descriptions).

3.4.1 Research Questions

We performed several experiments using the multi-level indexing pyramid to determine if the pyramid is complete, that is, if it can unambiguously classify a full range of visual content descriptions for an image in at least one level of its structure, and if it can improve results in an image retrieval system. The image descriptions were primarily produced by two groups of participants: *naïve users* (i.e., no prior training in indexing of visual information) and *indexers* (i.e., trained in library science methods in image indexing). We addressed the following questions.

I. How well can the Pyramid classify terms describing images generated by *naïve* participants, both in a *spontaneous*[19] informal description and a *structured* formal description for *retrieval*?

II. How well can the Pyramid classify the attributes generated, by participants experienced in indexing visual information, using a structured approach?

III. How well can the Pyramid guide the process of generating and assigning a full range of indexing attributes to images?

IV. How well can the Pyramid classify varying conceptual levels of information at the semantic level such as *specific, generic*, and *abstract*?

V. Can the Pyramid be used in a retrieval system to improve performance?

For the first four questions completeness and non-ambiguity were used as goodness criteria. We performed a qualitative analysis of the data that resulted from several manual indexing experiments to determine if all of the descriptions could be unambiguously classified into at least one level of the pyramid.

[19] Spontaneous descriptions, unlike structured ones, are generated without a controlled vocabulary, template, or any other guidance.

For the fifth question we implemented an image retrieval system and quantitatively compared the results of using free text queries and text queries that specified the desired level in the pyramid.

In the sections that follow first we give a brief overview of the experiments. Then we describe an *Image Indexing Template* [50], which was used in the experiments. The *Image Indexing Template* is useful in analyzing the completeness of the pyramid because it was developed independently by another researcher based on empirical data from image indexing experiments.

After describing the template, we explain the images, the descriptions, and each of the experiments in detail.

3.4.2 Overview

In the *first group* of experiments, in section 3.4.6.1 to section 3.4.6.3, we obtained spontaneous and structured image descriptions by naïve and experienced indexers. We classified the descriptions into levels of the pyramid, and in separate experiments used the pyramid to guide the indexing process.

In the *second group* of experiments, in section 3.4.6.4, we mapped the *Image Indexing Template* to the pyramid.

In the *third group* of experiments in section 3.4.6.6 we built a keyword retrieval system using the pyramid.

3.4.3 The Image Indexing Template

An *Indexing Template* was developed independently over several years of research using a data-driven approach from an information sciences perspective by Corinne Jörgensen [47][50][53]. The template was developed directly from image descriptions generated in several image indexing experiments with the purpose of providing a structured approach to indexing. In other words, the template was developed so that individuals could use it to manually describe visual content. The pyramid, on the other hand, was developed conceptually based on research in several fields related to visual information, mainly with the purpose of classifying visual attributes for different purposes, including the construction of better visual information organization systems.

Since the template has been tested in image description tasks, it is natural to use it experimentally to address the research questions for the pyramid that we have posed. Some of the fields in the template, and their brief description, are shown in Table 1. A person using the template would describe an image to fill in the corresponding fields (e..g, describe the personal reaction to the image).

Table 1. Image Indexing Template from [47][50]

1. LITERAL OBJECT (perceptual).

This class contains items which are classified as being literal (visually perceived) objects.

2. PEOPLE (perceptual).

The presence of a human form.

3. PEOPLE QUALITIES (interpretive).

Interpretive qualities such as the nature of the relationship among people depicted in an image, their mental or emotional state, or their occupation.

4. ART HISTORICAL INFORMATION (interpretive).

Information which is related to the production context of the representation, such as Artist, Medium, Style, and Type.

5. COLOR (perceptual).

Includes both specific named colors and terms relating to various aspects of color value, hue, and tint.

6. LOCATION (perceptual).

Includes attributes relating to both general and specific locations of picture components.

7. VISUAL ELEMENTS (perceptual).

Includes those percepts such as Orientation, Shape, Visual Component (line, details, lighting) or Texture.

8. DESCRIPTION (perceptual).

Includes descriptive adjectives and words referring to size or quantity.

9. ABSTRACT CONCEPTS (interpretive).

Abstract, thematic, and symbolic image descriptors.

10. CONTENT/STORY (interpretive).

Attributes relating to a specific instance being depicted, such as Activity, Event, and Setting.

11. PERSONAL REACTION

Personal reactions to the image.

12. EXTERNAL RELATIONSHIP

Comparison of attributes within a picture or among pictures or reference to an external entity.

3.4.4 Images

Two groups of images were used for the experiments (Table 2). The first set, *set A*, consisted of a random set of 700 images taken from the World Wide Web by an automated process. Since images were produced by different institutions and individuals for a wide range of purposes, they included photographs, cartoons, illustrations, graphics, and animations, among others. The images in this first set did not have any text associated with them. This was problematic for some of the experiments because generating semantic attributes for the images was often not possible (e.g., how do you assign an abstract level attribute to a graphic of a button?). Therefore, we sought a second set of images that would be useful for creating semantic descriptions, particularly at the specific and abstract levels. Set B consisted of twelve color photographs of current news events in different parts of the world and was obtained randomly from an Internet news newsgroup [96]. Each image in set B was accompanied by a short descriptive caption.

Table 2. Summary of image collections used in the experiments.

Set	Description
Web images (set A)	700 images collected randomly from the Internet. Images in this set were not accompanied by any text. Images included photographs, cartoons, illustrations, graphics, animations, etc.
News images (set B)	12 color photographs collected randomly from an Internet news newsgroup. Each image was accompanied by a caption. Example caption: *US special envoy to the former Yugoslavia Robert Gelbard (R) talks to the press as the leader of Kosovo Albanians, Ibrahim Rugova (L w/ glasses), listens following their talks in Pristina, Yugoslavia March 10. Gelbard came to Pristina March 10 to seek peace between Serbian police and ethnic Albanians.*

3.4.5 Generation of Image Descriptions

An image description is a set of terms or sentences that describe the visual contents of an image.

The *naïve users* were forty one beginning *Master of Library Science* students[20] from a variety of backgrounds who had no previous experience with image indexing, retrieval nor with more general indexing and classification procedures. The *indexers* were twenty-two students that had received training in the theory and practice of indexing images in Corinne Jörgensen's "Indexing and Surrogation" class.

Additional descriptions were generated by the author and his colleagues Corinne Jörgensen and Ana B. Benitez (Table 3).

Table 3. Summary of experiment participants.

Participants	Description
41 Naïve participants	No training in indexing of images.
22 Indexers	Trained in Library Science, specifically on image indexing.
3 Researchers	One of the researchers is an expert on image indexing. The other two do not have a background in Library Sciences.

[20] Students in Corinne Jörgensen's class at the Department of Library and Information Studies at the State University of New York at Buffalo.

Naïve Participants

Using the methodology reported in [48], forty-one naïve users viewed projected images and produced both spontaneous (informal), and retrieval-oriented (formal) image descriptions. Each image was projected for two minutes in a classroom setting. For the spontaneous descriptions the subjects were asked to write the words or phrases that came to their mind as they viewed the images. The informal descriptions were lists of words or short phrases describing the images. For the formal descriptions the setup was the same but the participants were asked to describe each image as if they wanted to retrieve it. The formal descriptions, therefore, were more complete sentences or long phrases *describing an image to be retrieved*. Four images from set A were randomly selected and each image was described by the same individual using both methods. The descriptions generated by six of the forty-one individuals were then randomly selected for the experiments. Since each individual described four images using two methods, we obtained a total of 48 image descriptions. This resulted in approximately 500 terms, 242 from spontaneous descriptions and 241 from retrieval-oriented descriptions, with an average of 10.4 terms per image (Table 4).

Table 4. Summary of descriptions generated by *naïve* indexers for the experiments.

Original Participants	Indexing methods	Images Indexed	Images Selected
41 Naïve indexers.	Spontaneous and retrieval oriented.	Each individual indexed 4 randomly selected images from Set A using both methods.	Descriptions of 6 individuals were selected randomly, for a total of 48 image descriptions.

Indexers

The *indexers* (twenty-two students that had received training in the theory and practice of indexing images) produced structured indexing records using the *Indexing Template* described above for the two sets of images (Sets A and B). The indexers used a controlled vocabulary which they developed for the project using manually selected terms from existing thesauri (*AAT* [44] and *TGM I* [54]). The terms were selected based on their appropriateness for describing *visual* content and indexers were able to add free-text terms when no appropriate thesaurus term was present. After a class discussion these terms were added to the thesaurus. Images were indexed online through a web-browser interface with multiple frames displaying the image to be indexed, the *Indexing Template*, and the controlled vocabulary. Indexers were presented with a random selection from the image collection and were not allowed to "choose" which images they indexed; they indexed a total of approximately 550 images.

Table 5. Summary of descriptions generated by *indexers*.

Original Participants	Indexing Methods	Images Indexed
22 persons trained in Library Sciences, particularly in image indexing.	Structured indexing using the Indexing Template	Sets A and B; approx. 550 images.

Indexers were instructed to index the "visually relevant" and "visually striking" content of the images, rather than attempting to fill each slot of the *Indexing Template*. Indexers spent an average of ten to twenty minutes indexing each image. Inter-indexer consistency was developed through discussion of the indexing process and comparison of practice indexing records in class, and was assisted by the controlled vocabulary.

Researchers

The author and his colleagues Corinne Jörgensen and Ana B. Benitez generated spontaneous descriptions for the twelve images of set B. In addition, they generated descriptions for the images of set B using the Pyramid as a guide for indexing (Table 6).

Table 6. Summary of descriptions generated by the researchers.

Participants	Description
3 researchers.	12 images from set B using the pyramid to guide the indexing and generating spontaneous descriptions.

3.4.6 Experimental Results

As was mentioned earlier, three groups of experiments were performed. The first group consisted of several experiments that used the image descriptions generated by the three groups of participants (*naïve, indexers, researchers*). These experiments, described in detail below, are summarized in Table 7.

Table 7. Summary of the experiments.

	Naïve	**Indexers**	**Researchers**
Spontaneous descriptions	Experiment IA	None	Experiment IIA
Spontaneous for retrieval	Experiment IB	None	None
Structured descriptions	None	Experiment III	None
Using pyramid	None	None	Experiment IV
Using captions only	None	None	Experiment IIB

The results for the second group of experiments, in which the Indexing Template was mapped to the pyramid are presented in section 3.4.6.5. The results for the third group of experiments, in which we build a retrieval system using the pyramid are presented in section 3.4.6.6.

The data sets for the first two groups of experiments are somewhat small, so those experiments can be considered preliminary and exploratory. However, the results as aggregated across the experiments suggest that further work in developing a conceptual approach to image indexing (such as that instantiated in the Pyramid) would be beneficial. The dataset for the last experiment is large enough to demonstrate the benefits of using the pyramid structure in a retrieval system.

3.4.6.1 Classifying Spontaneous Descriptions

Experiment I addressed research question I by determining how well the Pyramid classifies terms from naïve participants' descriptions. The author and his colleagues Corinne Jörgensen and Ana B. Benitez classified the six participants' 242 terms from spontaneous descriptions and 241 terms from retrieval-oriented descriptions for the same four images from image set A into levels of the Pyramid.

As column IA in Table 8 indicates, attributes were generated for all levels of the pyramid except for the lowest syntactic levels for *spontaneous* descriptions (Global Distribution and Global Composition). This is in agreement with previous analysis of spontaneous descriptions demonstrating that lower-level descriptors are used less frequently in spontaneous description tasks [51].

When naïve participants were asked to describe images more formally in a *retrieval* context (column IB in Table 8), we see that attributes occur at the lower syntactic levels as well as with descriptions generated by *indexers*. It should be noted that the terms from the spontaneous and retrieval-oriented descriptions were not necessarily the same; different terms at different levels of the Pyramid were used in the spontaneous and structured descriptions. Comparative analysis of the two describing tasks data is interesting but not directly relevant to the questions presented here, which focus on whether a *range* of attributes for each task is accommodated at all ten levels of the Pyramid. The results indicate that the Pyramid is able to classify a variety of attributes as described by naïve users in both a spontaneous describing task and in a more formal, retrieval-oriented task.

Table 8. Mapping of image attributes to the pyramid levels as generated by different methods. The X in the table indicates that attributes were generated at the corresponding pyramid level for the corresponding experiment. Blank boxes indicate that no attributes were generated at that level.

Pyramid Level/Experiment	**IA Spontaneous**	**IB Retrieval**	**IIA Researcher**	**IIB Caption**	**III Struc-tured**
1. Type/Technique	X	X			X
2. Global Distribution		X	X		X
3. Local Structure	X	X	X		X
4. Global Composition		X	X		X
5. Generic Objects	X	X	X	X	X
6. Specific Objects	X	X	X	X	X
7. Abstract Objects	X	X	X	X	X
8. Generic Scene	X	X	X	X	X
9. Specific Scene	X	X	X	X	X
10. Abstract Scene	X	X	X	X	X

The news images in set B were used in experiment II. Corinne Jörgensen and Ana B. Benitez *spontaneously* described a set of five randomly selected images each and the descriptions were classified into the pyramid levels by the author, who did not describe those images (column IIA in Table 8). The Pyramid and the Template were not used. Although Corinne has extensive experience in image indexing and Ana does not, no major differences in the overall range and types of attributes generated between the two were found. Both described objects, events, people, as well as emotions and themes. Note that terms were present in all levels except that of Type/Technique. Example descriptions are shown in Table 9.

Table 9. Sample spontaneous descriptions of news images by one participant experienced in image indexing (1) and one participant not experienced in the task (2).

Researcher 1	Researcher 2
Airport Greek policeman Guns Outdoor Duty Terrorism Protection Death	Interview Outdoors Three men Reporters Grim expressions Microphones thrust in face

In an additional experiment, we mapped the words from the captions of each of the images in set B to the pyramid. Not surprisingly, the captions had attributes at the semantic levels (column IIB in Table 8), but none at the syntactic levels. An example caption is presented in Table 2.

Information appeared on all of the semantic levels of the Pyramid across both the authors' descriptions and the captions; however, with the caption information there were more terms which belong in the specific levels (again, a not unexpected result). Specific level information depends on the observer's prior familiarity with the particular object, person, or place depicted. Interestingly, the mapping results for the captions are more closely related to the "spontaneous" descriptions than to descriptions in a retrieval context.

Overall, the experiments with spontaneous and caption descriptions show support for the ten-level Pyramid. We were able to uniquely classify all of the attributes generated into at least one level of the pyramid.

3.4.6.2 Classifying Structured Descriptions

Experiment III addressed how well the Pyramid can classify the attributes that result from a structured approach to indexing. Structured indexing implies the use of indexing tools such as a metadata structure and a controlled vocabulary, as well as training in the process of indexing. For this experiment, structured image descriptions generated using the *Indexing Template* were used. Although the indexers were instructed to index only “salient” aspects of each image, a large number of attributes were generated in each case. There seemed to be an overall tendency to “over-index” the image using the template. Additionally, students' indexing was being graded, prompting them to be very thorough. This, however, produced in-depth indexing for each image. Thirty-three randomly-selected indexing records for 33 unique images generated by the 22 student indexers for images from Set A were mapped to the Pyramid by the author and his colleagues (approximately 1,050 terms). Each of the researchers performed mapping for a different set of images.

Overall, the attributes from the Indexing Template were mapped at all levels of the Pyramid, and each indexer used attributes from several levels of the Pyramid (Table 10). However, only *one* term occurs at the Specific Scene level across all of the indexers. This is to be expected since there was no descriptive text attached to these images, without which it is often not possible to derive those types of attributes. In the case of the naïve users' descriptions, however, “accuracy” was not such a concern and they did in fact supply specific terms with no concrete knowledge of the correctness of these terms.

It is interesting to compare the mapping to the pyramid of the indexers' descriptions with the mapping of the spontaneous descriptions of the naïve users (Table 8 columns IA and III). Note that the mapping for naïve participants in the retrieval task (column IA) occupies the same levels of the pyramid as mapping for the indexers in the structured indexing task (column III). This consistency in the descriptor levels in the pyramid in the two tasks suggests that when respondents were asked to describe the images as if they wanted to find them (retrieval mode), their descriptions become more "formal" or structured, as was shown previously in [50]. This also suggests that the needs of image searchers may be more closely suited by a structured method (e.g., the Pyramid being tested) to classify image descriptions. Other interesting dimensions include the consistency in the levels of description between different types of images (photographs, graphics, etc.).

This preliminary analysis demonstrates good correspondence between the index terms and the levels of the Pyramid. This suggests that the Pyramid's conceptual structure can be used to classify a wide variety of attributes produced as a result of a structured indexing process such as that using the Indexing Template. While mapping of the descriptors to the levels of the Pyramid was straightforward in most cases, some further guidelines would be beneficial for performing such mappings. The Pyramid is designed both to describe an entire image *and* to be used recursively to describe specific areas of an image. In the mapping process, the capability of the Pyramid to be used recursively resolved some of the issues encountered during the mapping. For example, an individual object in an image could be described using several attributes (e.g., color); with respect to the *object*, the *color* is a global distribution measure, but with respect to the *image* it is an attribute at the local structure level. These results suggest that the Pyramid itself may be a candidate for guiding the image indexing process.

Table 10. Sample image indexing record terms from the Indexing Template mapped to the Pyramid levels.

IMAGE TERM	PYRAMID LEVEL
painting	1. Type/Technique
oil	1. Type/Technique
cracked	2. Global Distribution
Red, white	3. Local Structure
background	3. Local Structure
rectangle	3. Local Structure
center	3. Local Structure
eye level	2. Global Composition
flag	5. Generic Object
historical landscape	6. Generic Scene
patriotism	9. Abstract Object
Pride	10. Abstract Scene

3.4.6.3 Generation of Indexing Based On The Pyramid

Experiment IV tested how well the Pyramid guides the process of generating and assigning a full range of indexing attributes to images. In this experiment, two image sets were indexed using the Pyramid.

For the first part of the experiment, we (the author and Ana B. Benitez) independently indexed a total of 31 images of Set A. The indexing of this set of images was performed on randomly selected, *unseen* images, not used by the same author in previous mapping work or seen in any other context. This indexing of web images produced 287 terms. In contrast to the indexing performed by the student indexers, no controlled vocabulary was used.

Sample image descriptions for this work using the Pyramid as a guideline are shown in Table 11. The major conclusion from this experiment is that descriptions for each level of the Pyramid were easily generated. Although the examples do not contain *Specific Object* and *Specific Scene* descriptions, some of these were populated based on the researchers' general knowledge about the content depicted in the images. It should be noted that the descriptions generated here are shorter than the descriptions generated by student indexers using the Indexing Template (on average 9.3 terms per image for the authors versus 10.4 for the student indexers), perhaps as a result of a lack of a controlled vocabulary. However, the goal here was not to demonstrate the completeness of indexing done using the Pyramid but to demonstrate that the levels of the Pyramid can suggest adequate conceptual guidance for the process of indexing and that all Pyramid levels are relevant to the visual content of an image. Indeed, the results suggest that this is the case.

Table 11. Sample image descriptions generated using the Pyramid as a guideline.

Pyramid Level	**Image 1 terms**	**Image 2 terms**
1. Type/Technique	color photograph	color photograph
2. Global Distribution	white, brown	clear
3. Local Structure	curves	curves, lines
4. Global Composition	centered, eye level, close-up	leading line
5. Generic Object	person, man, head, neck, shirt	ducks, lake, mountain, bridge, vegetation
6. Generic Scene	portrait, indoors	outdoor, daytime, landscape
7. Specific Object	None	None
8. Specific Scene	None	None
9. Abstract Object	efficiency	family
10. Abstract Scene	None	vacation dream

The second part of Experiment IV followed the procedures for the web image indexing using the Pyramid. Two researchers (one of whom also participated in the web image indexing) indexed five images each from Set B (news images), using the Pyramid again to guide the indexing (135 terms or 13.5 terms per image). Results of the mapping were identical to the mapping of spontaneous descriptions for the previous Set A, with information lacking only in the Specific Object and Specific Scene Levels. When captions are used, these levels are populated as well (e.g., Mirage 2000 Jet Fighter; Aden, Yemen).

Using the pyramid, the participants generated descriptions relevant to more levels of visual content description than those that were generated by naïve indexers in the spontaneous tasks without the pyramid (i.e., global distribution and global composition descriptions were entered). This is similar to results reported in [50], which increases the images' utility and their access points for retrieval. Therefore, the Pyramid is capable of generating attributes in the same areas covered by the image Indexing Template.

3.4.6.4 Levels of Indexing

The fourth research question concerns how well the Pyramid structure can classify varying levels of information at the semantic level. The results from the news image indexing using the Pyramid are most instructive for this question because the web images that we collected did not include any textual information useful for assigning semantic level attributes. In contrast, the news images were accompanied by captions (see examples in Table 2) that provided a lot of information that could be used to assign semantic labels to the images.

The generic and specific levels of data are handled well by this conceptual structure, although we did find important open issues. One of the significant questions was the level at which a description should be placed. The choice between generic and specific, for example, was sometimes difficult since annotations at these levels may depend on the specific application or context (e.g., generic object: cat; specific object: Siamese cat, or generic object: Siamese cat; specific object: Felix). The distinction between object and scene, at the abstract level, also proved to be challenging in some cases, since the same abstract term could apply to a single object or the entire scene (e.g., flag or scene representing patriotism). Although it is perfectly valid to have the same term at two different levels, we found that indexers were sometimes unsure about the assignment. As for the type of terms to use, the Indexing Template seemed to provide a more comfortable fit in some cases since it elicits terms in more detailed categories (e.g., abstract theme, symbol, and emotion at the abstract levels). For example, the Pyramid does not make a distinction between object and event, which is made by the template. Lastly, we found that syntactic level descriptions (e.g., global distribution) were easier to generate for some images than others (e.g., the global color blue in a texture image from the web vs. the global color of a news image). In many applications, however, indexing at the syntactic levels can be performed using automatic techniques.

Abstract qualities were slightly more problematic for indexers to distinguish and in some cases these were not felt to be particularly intuitive. The Pyramid structure defines an Abstract Object as a localized entity with abstract meaning (e.g. flag = patriotism); if the abstraction cannot be localized it becomes an Abstract Scene (e.g. democracy).

3.4.6.5 Mapping from the Template

We manually mapped attribute types from the Indexing Template to the Pyramid. Some types mapped easily on a one-to-one basis (e.g. Visual Elements attributes to Local Structure). In other cases though, attributes from several *different* facets in the Indexing Template populate the same Pyramid levels. This was primarily a matter of resolving different levels of analysis. For instance, the Indexing Template distinguishes between "Living Things" and "Objects," while the Pyramid includes both of these as "Objects." The Pyramid describes the generic and specific naming levels explicitly, while the Indexing Template left this component to standard online bibliographic system implementation using "descriptors" (general keywords) and some attributes at the specific level are included in the template as identifiers (proper nouns).

The Indexing Template seemed to provide a more comfortable "fit" for some of these more abstract terms. For instance, term such as "democracy," "patriotism," and "respect," are perhaps more easily characterized by the more finely distinguished theme, symbol, and emotion of the Indexing Template than abstract object or scene. It may be that at the "abstract" level the object or scene distinction is less useful than a finer-grained analysis, or perhaps than a unitaryapproach (a *single* "abstract" level). Additionally, the Pyramid does not make a distinction between object and event, which is a natural part of descriptive language; this distinction is usefully made by the template. Both of these are open issues and bear further consideration and testing.

Mapping the template to the pyramid demonstrated that the pyramid is able to classify the wide variety of attributes that appear in empirical research.

Table 12. Image indexing template attributes mapped to pyramid levels.

INDEXING TEMPLATE (GROUP & ATTRIBUTE TYPE)	**INDEXING TEMPLATE (ATTRIBUTE)**	**EXAMPLE**	**PYRAMID LEVEL**
EXTERNAL INFORMATION			
	> Image ID		NA
	> Creator/Author		NA
	> Title		NA
	> Publisher		NA
	> Date		NA
	> Image Type	color, X-ray, graphics, etc.	Type/Technique
	> Access Conditions		NA
	> Technical Specifications	resolution, file size, etc.	NA
INFERRED INFORMATION			

> "Environment"	>> When - time in image	Middle Ages, summer	Generic Scene
	>> Where - General	city, rural, indoor, office	Generic Scene
	>> Where - Specific	Paris, Chrysler Building	Specific Scene
> Subject/Topic		overall subject/theme: nature	Abstract Object/Scene
> Medium		oil, watercolor, digital image	Type/Technique
> Symbolism		Garden of Eden, afterlife	Abstract Object/Scene
> "Why"	>> Emotions/Mental States	sadness, laughter	Abstract Object/Scene
	>> Relationships	brothers, romance	Abstract Object/Scene
> "Miscellaneous"	>> Point of view/Perspective	bird's-eye, close-up	Global Composition
	>> Style	abstract, realism, etc.	Abstract Scene
	>> Genre	landscape, portrait	Generic Scene
	>> Atmosphere/overall mood	gloomy, mysterious	Abstract Object/Scene
VISUAL ELEMENTS			
> Color	>> Color	Red, blue	Global Dist/Local Strc. Structure
	>> Color Quality	dark, bright	Global Dist/Local Strc. Structure

	>> Placement	center, overall, foreground	Local Structure
> Shape	(>> Placement)	Square, elongated, curved	Global Dist/Local Strc. Structure
> Texture	(>> Placement)	smooth, shiny, fuzzy	Global Distribution, Local Structure
LITERAL OBJECTS			
> Category - General		What group; tool	Generic/Specific Objects
> Type - Specific	(>> Placement)	What it is - hammer	Generic/Specific Objects
	>> Shape		
	>> Texture		
	>> Size		
	>> Number		
	>> Color		
LIVING THINGS			
	> Type	human or what animal	Generic Object
	(>> Placement)		
	>> Size	large, very small	
	>> Gender	male, female, undetermined	Specific Objects
	>> Age		Specific Objects
	>> Number		
	>> Pose	seated, standing, lying down	Generic/Specific Scene

	>> Name	Ghandi	Specific Object/Scene
	>> Physical Action/Event	running, talking, football	Generic/Specific Scene
	>> Status	occupation, social status	Abstract Object/Scene
COLLATERAL INFORMATION	> Caption		
	> Related Text		
	> Voice Annotations		

3.4.6.6 The Pyramid in a Retrieval System

We built a retrieval system to determine if access to images improves when the pyramid structure is used. In the following sections we discuss these experiments.

Images

Three sets of images were used for the experiment. The first set of images is the same set A of section 3.4.4 (703 images collected randomly from the internet). The second set of 98 images was a sample drawn from the MPEG-7 content set (CDs 6-8) [193]. Finally, the third set of images (49 images) was a sample drawn from the MPEG-7 Melbourne Photo Database. Different sets of images were used in order to achieve a full range of image types and content.

Indexing

The images were indexed using the indexing template of section 3.4.3. The indexing was done with the same web-based tool described in section 3.4.5 by student indexers at the State University of New York at Buffalo, and by the researchers (Ana B. Benitez, Corinne Jörgensen, and the author). A limited Controlled Vocabulary was used to improve indexing consistency but terms could be added which were not in the Controlled Vocabulary. In order to improve consistency in the indexing terms, one of the researchers reviewed the terms added by the participants and made minor corrections (e.g., if two similar terms were added by the participants, only one was used).

Indexing terms from the template were mapped to each level of the pyramid. A new indexing attribute was added *to the template* to accommodate Genre indexing terms at a finer indexing level. Terms that had been placed in Genre included overall types such as still life and portrait, and general genre classification of the image, such as "science fiction" or "action and adventure." Within the template, these latter terms were moved to a separate indexing group called Category. The pyramid structure, however, was not changed in any way and indexing terms in the Category group of the template were mapped to the pyramid in the same way that other terms were mapped. Most of the terms in the template's Category group mapped to the abstract scene level of the pyramid.

The images from the Melbourne Photo Database were indexed using the pyramid as the starting point, mainly to complement the set that was indexed using the template.

Searching

In total 850 images were indexed. In order to use the multi-level indexing pyramid for retrieval, we constructed a search engine to allow searching using terms at any level of the pyramid: *Type/Technique, Global Distribution, Local Structure, Global Composition, Generic Object, Specific Object, Abstract Object, Generic Scene, Specific Scene*, and *Abstract Scene*.

A search with a term in the Local Structure field, for example, would only search for images that contain that term at the same pyramid level (i.e., Local Structure).

In order to perform the comparisons with keyword search, we also implemented a second search engine which did not use the pyramid structure. The interface of this search engine, then, contains only a keyword search field. In this scenario, a search for a term will search all the terms for each image, without considering the levels of the pyramid.

Evaluation

In both search engines implemented, we used the same database of images, with the same terms. One of the advantages that became very evident is that if an image has the same term in more than one level, ambiguity will occur if the pyramid structure is not used. For example, an image with a small blue circle in the center may have the attribute "blue" at the local structure level. The same image, may have an overall blue color, so it may have a "blue" attribute at the Global Distribution level. A second image may have an overall blue, but not a blue circle (or local blue element). A person searching for images that contain a blue area, might want to use the "Local Structure: blue" query. This query will specifically target an image with a local blue. Using the second search interface, with a single field, will produce all images that have blue, whether it is local or global. This can significantly increase the number of errors: many images that should not be retrieved will be returned by the system.

To see whether this ambiguity is common, we computed the number of times that terms repeat for the same image at different levels (e.g., the blue circle/blue image case). The set of 850 images contained a total of 18,975 terms. 3,216, or 17% of those terms occur more than once, at different levels in the same images. In total, approximately 700 images contained terms that repeated at different levels. In other words, in searching 700 out of 850 images, there is ambiguity if the pyramid is not used, because the same term is used to describe visual attributes at different levels. This ambiguity in 83% of the images is quite significant and underlines the importance of structures such as the one presented.

For the purposes of understanding visual information and users, it is important to analyze the reasons for these ambiguities. The same term is used to describe the same image at different levels in the following scenarios:

- A term that can be used to describe an object is used at the abstract level. For example, a portrait of a basketball player might be labeled as “basketball” even though a basketball is not shown in the picture.
- The term blue could be abstract or syntactic. This shows why those two levels are important. If the distinction is not provided, a query for "blue" will be ambiguous (i.e., the color blue or the feeling?). Using the pyramid level, the meaning would be unambiguous.

Some examples of terms that repeat at different levels for the same image are shown in the Table 13. There are syntactic as well as semantic terms. In addition, note that terms at different levels of the pyramid repeat, meaning that each level proposed helps disambiguate between such terms.

Table 13. Example of terms that repeat at different levels of description of the same image.

No. appearances in a description	Keywords
10	Foreground
9	center, left, upper, foreground, right
8	center, left, upper, foreground, lower
7	center, right, left, one, bottom , GREY
6	bright, center, background, left, upper, right, bust, one, small, grey
5	curved, large, right, left, metallic, rectangle, center, grainy, floating, foreground, smooth, lower, citizen, smooth, interior, front, dull, one, floating, grey
4	bright, blue, flowery, right, center, red, left, circle, flat, foreground, ground, background, black, white, building, sports, faint, tennis, clear, male, bright, smooth, dull, signs, river, glasses, irregular, large, adult, human, standing, clothing, grainy, black, white, green , grey, overall, top, upper
3	adult, smiling, young, pink, bluish, heart, aircraft, airplane, silver, ark, male, blurred, irregular, map, ship, space, bright, pale, roost, basketball, sports, miniature, dog, view, glossy, slot, tree, leafy, theater, clothing, straight, space, sports, christmas, many, accessories, wooden, house, tableware, scuba, tram
2	female, standing, friendship, chair, cartoon, fantasy, furnishing, furry, clouds, confusion, airplane, hostility, university, lasalle, lake, landscape, light, vegas, carnival, lights, student, teacher, writing, brown, seated, aircraft, class, equipment, olive, sandy, starbust, farming, watery, death, fantasy, dimensional, age, hell, trees, islands, child, house, pebbled, etc.

In the retrieval experiment that follows we are not measuring the accuracy of the indexing terms, but rather the retrieval performance improvements obtained when queries include a keyword and the level of the pyramid they refer to. In other words, we make the assumption that all indexing terms have been correctly assigned. In Table 14, we compare "free text" query results not using the pyramid with “pyramid queries” in which the pyramid is used to specify a level of retrieval. Precision measures the percentage of retrieved images that are relevant and recall measures the percentage of relevant images that are retrieved. The recall for free text and pyramid queries is 100% in both cases. All of the relevant images are retrieved using both approaches since we assumed that the terms were correctly assigned. The precision, therefore, is 100% for the pyramid queries, but varies with the free text queries as indicated in the table above. This occurs because images that are not relevant are returned when the pyramid level is not used. As shown in the table, precision can be as low as 20% if the pyramid is not used; the pyramid disambiguates those terms that may be used to describe visual content at different levels.

The database we used contained a significant amount of variation, and the experiments show the possible improvements if the pyramid is used. It is again important to note that not all the levels are necessary in every application or for every image.

Table 14. Comparison of free text and pyramid queries.

Indexing Term	**Free Text Matches**	**Pyramid Matches**	**Pyramid Level**	**Free Text Errors**	**Free Text Precision**
Geometric	11	7	Loc Struct	4	63%
Friendship	26	22	Abs Scene	4	85%
Coarse	6	3	Glob Distrib	3	50%
Cartoon	29	19	Type/Tech	10	65%
Clouds	4	3	Gen Obj	1	75%
Shiny	36	34	Loc Struct	2	94%
University	10	5	Gen Scene	5	50%
Lake	18	1	Gen Scene	17	94%
Landscape	61	58	Gen Scene	3	95%
University	10	5	Gen Scene	5	50%
Landscape	61	58	Gen Scene	3	95%
River	7	6	Gen Obj	1	85%
Sculpture	16	6	Type/Tech	10	62%
Gold	26	4	Glob Distrib	22	84%
Pale	57	12	Glob Distrib	45	78%
Moon	5	1	Spec Scene	4	20%
Happiness	91	79	Spec Scene	12	86%
Desire	7	4	Abs Obj	3	57%

An interesting option, which we did not test, is to combine terms from different levels of the pyramid to perform the queries. For example, a term for *Local Structure* could be used in conjunction with a term for *Generic Object*.

3.5 DISCUSSION

We presented a *Multi-Level Indexing Pyramid* and results on qualitative analysis of data from several experiments. While the research demonstrated that the distribution of attributes among the levels of the Pyramid varies depending upon who generated them (*indexers, researchers, naïve participants*) and upon the task (*describing, indexing, retrieval*), we found no instances where an attribute could not be classified into a level of the Pyramid. In addition, the Pyramid provides guidance to indexers by making explicit *specific, generic*, and *abstract* levels of description. Especially useful is the recursive nature of the Pyramid, which permits associations among objects and attributes (e.g., can be applied to a scene, object, section of an image, etc.).

The limitations of the experiments are the small number of images used and the use of student indexers. However, the limited number of images still produced a large number of terms, which were mapped in the experiments and were more than adequate to demonstrate that the Pyramid levels can be used to classify a wide range of terms. The student indexers produced high-quality indexing records, and the data generated by the non-student participants did not differ from data gathered by other methods. Although we would not expect a change in the outcome of the analysis if we used data from professional indexers, it would certainly be useful to perform more experiments using records generated by groups of experts.

Our analysis of the results shows that all of the levels of the pyramid were used in at least one of the tasks, and that the pyramid was able to unambiguously classify all of the descriptions that were generated. Given the wide range in the types of images utilized, and the variety of the tasks and participants we can argue that the Pyramid is a robust conceptualization of visual image content. The results of these experiments support the use of the Pyramid both as a method of organizing visual content information for retrieval and as a method for stimulating additional attributes, which could be added to existing records. Additionally, the Pyramid's ten-level structure could be used to represent attributes generated by both manual and automatic techniques.

Some of other key issues in applying the pyramid include whether the distinction between the different levels is clear and if the definition of the levels is independent of the user, domain, and task, among others. Although we did not perform experiments specifically to answer these questions, we did analyze data from a fairly wide variety for several distinct tasks. Our analysis suggests that in some cases the distinctions between the levels can be difficult (e.g., difference between abstract and generic levels), particularly if the person using the pyramid is unfamiliar with those concepts. However, we argue that with a clear understanding of the concepts that define the pyramid and the ways in which it can be applied, those difficulties can be overcome. Furthermore, in our experiments we did not find anything to suggest that the definition of the levels depends in any way on external factors such as the user, domain, or task. This might be partly due to the fact that the structure was developed considering the work in different fields, but without relying on any specific model, domain, or task.

A different, but related and very important issue has to do with indexer consistency. Even if the levels of the pyramid are clearly defined for several indexers, their interpretation might be subjective and vary over time, or errors might be made during the indexing process. As with the *Image Indexing Template*, one common practice way to alleviate the problem is to use controlled term vocabularies for the indexing and or search process.

3.5.1 Extensions to the Framework and Future Work

Future research includes testing the Pyramid more widely, using additional material and more experienced indexers to generate descriptors, as well as exploring indexer training using the Pyramid. Other important work should focus on determining whether some combination of the pyramid with another structure such as the *Indexing Template* would be useful, and the circumstances under which each may be a more appropriate choice to guide indexing.

The goal of these experiments was not to test one against the other but rather to substantiate that the range of attributes addressed by the Pyramid is adequate. The experimental work pointed to some differences, as discussed earlier, between the Pyramid and the Indexing Template. Differences that could fruitfully be explored between the two structures concern the range of attributes produced by each, the differences among the attribute types generated and the levels populated, and the number of attributes produced by each, as well as the communities that would find these structures most useful.

One very interesting question to investigate is whether the Pyramid can serve as an entry point in providing access to images within a specific domain. While we tested the range of attributes classifiable within a generalized domain (the web images), levels of the Pyramid may be populated differently across different domains. For example, in a collection of satellite photographs syntactic level attributes, even if assigned manually, could be of great importance. In contrast, in a collection of news images the syntactic levels of the pyramid may not be necessary and most or all of the descriptions will be at the semantic levels.

As image collections become even more diverse and accessible, refinements to target specific types of images, video, and audio will become even more important. The current research has produced data that would aid in exploring these questions as well.

Additional experiments worth pursuing include determining if using the pyramid causes a time reduction in indexing, and if it can be used to facilitate browsing. Testing the pyramid, in a retrieval framework, against other structures (not only keyword search) would be worthwhile.

Finally, the pyramid can be extended to other aspects of multimedia. In [142] we proposed the application of the pyramid to audio without finding any important limitations. No experiments have been performed with audio descriptions, however. The application of the pyramid to video was described in section 3.3, but no experiments have been performed to identify the open issues.

3.6 SUMMARY

In this chapter we presented a novel ten level pyramid structure for indexing visual information at multiple levels. The structure is suitable for making *syntactic* and *semantic* as well as *perceptual* and *conceptual* distinctions. For completeness, we presented the *semantic information table*, which represents *semantic* information extracted from *visual* and *non-visual* data. The multi-level pyramid allows classification of visual attributes and relationships at multiple levels.

Several experiments were performed to evaluate the application of the pyramid presented in this chapter. First, descriptions were generated manually, using the pyramid and using an image-indexing template developed independently over several years of research into image indexing [50]. The descriptions were generated by individuals trained in indexing (i.e., with an information sciences background) and also by individuals without any prior indexing experience (i.e., naïve users). Experiments were performed to answer several questions: (1) How well can the Pyramid classify terms describing image attributes generated by naïve users for retrieval? (2) How well can the Pyramid classify the attributes that result from a structured approach to indexing? (3) How well can the Pyramid guide the process of generating and assigning a full range of indexing attributes to images? (4) How well can the Pyramid classify varying semantic levels of information (*specific, generic*, and *abstract*)?

The experiments presented suggest that the Pyramid is conceptually complete (i.e., can be used to classify a full range of attributes), that it can be used to organize visual content for retrieval, to guide the indexing process, and to classify descriptions generated manually. In a related set of experiments, it was shown that using the pyramid for retrieval tasks was more effective than using keyword search alone. This improvement occurs because a search in which the level of the pyramid is specified helps eliminate ambiguous results. For example, a query with the word "blue" could refer to the color (a syntactic level attribute), or to the emotion (a semantic level attribute). A simple "blue" query can easily return the wrong results if the user is interested in the semantic blue, and not in the syntactic blue. Were examined 29 keywords from descriptions generated by different individuals for 850 images, and found that using the pyramid resulted in improvements in precision between 5% and 80%.

4 LEARNING STRUCTURED VISUAL DETECTORS

4.1 INTRODUCTION

In this chapter, we present a new framework for the *dynamic* construction of structured visual object and scene detectors from user input at multiple levels. We also present a study of human observer's eye tracking patterns when observing images of different semantic categories.

In the *Visual Apprentice*, a user defines visual object or scene models via a multiple-level *definition hierarchy*: a *scene* consists of *objects,* which consist of *object-parts*, which consist of *perceptual-areas*, which consist of *regions*. The user trains the system by providing example images or videos and labeling components according to the *hierarchy* she defines (e.g., image of two people shaking hands contains two faces and a handshake). As the user trains the system, visual features (e.g., color, texture, motion, etc.) are extracted from each example provided, for each node of the hierarchy (defined by the user). Various machine learning algorithms are then applied to the training data, at each node, to learn classifiers. The best classifiers and features are then automatically selected for each node (using cross-validation on the training data). The process yields a *Visual Object or scene Detector* (e.g., for a handshake), which consists of an hierarchy of classifiers as it was defined by the user. The *Visual Detector* classifies new images or videos by first automatically segmenting them, and applying the classifiers according to the hierarchy: *regions* are classified first, followed by the classification of *perceptual-areas*, *object-parts*, and *objects*. We discuss how the concept of *Recurrent Visual Semantics* can be used to identify domains in which learning techniques such as the one presented can be applied. We then present experimental results using several hierarchies for classifying images and video shots (e.g., Baseball video, images that contain handshakes, skies, etc.). These results, which show good performance, demonstrate the feasibility, and usefulness of dynamic approaches for constructing structured visual object or scene detectors from user input at multiple levels.

In addition to the Visual Apprentice, we present a study in which we analyze human eye tracking patterns when observing images from 5 different semantic categories (*two people shaking hands, crowd, landscape, centered object*, and *miscellaneous*). The results of our study are important because finding similar eye tracking patterns for images within the same category allows the direct linking of eye tracking and automatic classifiers. For instance, instead of asking users to manually label examples in the Visual Apprentice, it could be possible to passively train the system by directly using eye tracking patterns for some image categories.

4.1.1 Why Learning At Multiple Levels?

It is desirable to construct systems that can automatically examine visual content, and label it based on the semantic information it contains. Many of the current systems perform this task by classifying images or video and assigning them semantic labels. Typically such classifiers are built (by experts) to perform specific tasks (e.g., indoor vs. outdoor image classification). The classifiers index image or video data, and users can then utilize the corresponding labels for searching and browsing. Different users, however, search for information in different ways, and their search criterion may change over time. Therefore, many of the current automatic classification approaches suffer from two disadvantages: (1) they do not accommodate *subjectivity* (i.e., the expert decides which classifiers to construct), (2) they do not allow the construction of structured models from user input at multiple-levels.

Manual construction of image or video classifiers can produce systems that are accurate and work well in specific domains. If the number of objects or scenes to classify is large, however, such approach becomes impractical. Furthermore, class definitions depend on the experts that build the systems, and any modification to the class definitions must be performed by the experts themselves. In addition, users may have interests that are different from those of the experts building such systems. The definition of a "handshake image class," for example, may vary among different individuals: for one user the class may include images that show the hands of two individuals, but nothing else. For another user, it may include only images in which people are pictured, from a certain distance, shaking hands. While specialized algorithms can be very useful in some domains (e.g., face recognition), we argue that successful Visual Information Retrieval systems should be *dynamic*, to allow construction of classifiers that cater to different users' needs. Algorithms should be as general as possible so that they can be applied in several domains, and they must exhibit enough flexibility to allow users to determine the classes in which they are interested. In addition, they should allow the definition of complex multiple-level models that can accurately represent (and capture) real world structures. One way to enhance the capability of such systems is to construct flexible frameworks that use machine learning techniques [186][83][108][65].

4.1.2 Related Work

Research in VIR has grown tremendously in the last few years (for recent reviews and references see [80][241][223][273][102][93][92]). Many of the systems (e.g., QBIC [200], VisualSEEk [246], VideoQ [89], Virage [72], Spire [80], etc.) have used query-by-example ("show me images like this one"), and query-by-sketch ("show me images that look like this sketch"). Some systems have enhanced capabilities for query formulation (e.g., in Spire, users can perform queries using examples from different images; in Semantic Visual Templates [93] the system tries to help the user formulate queries, using relevance feedback). Others have focused on classification, using visual features only, textual features only (e.g., WebSEEk [244]), or combinations of different types of features (e.g., textual and visual in news images; [206] visual and audio in [197]). Distinctions between different approaches can be made in many different ways. In chapter 2, for example, distinctions were made based on the level of description used, interface, type of features, etc.

Approaches that perform classification of visual content based on visual features, like the *VA*, can be divided into those that perform automatic classification at the scene level (indoor vs. outdoor [254][206], city vs. landscape [264]), and at the object level (faces [114], and naked people and horses [113]).

Scene level classifiers determine the class of the input image as a whole [264][254][206]. In many of the approaches the image is divided into blocks and the final classification decision is using feature extracted from all of the blocks. In the work of [254], for example, the image is divided into a 4x4 grid of 16 blocks. Features are computed for each of the blocks and each block is classified independently (indoor/outdoor). The final decision is made by concatenating the feature vectors for all of the blocks and applying a majority vote classifier. These approaches differ from the *VA* since images are classified based on their global features— not on the structure of local components (i.e., a user defined model of scene structure). In addition, the algorithms proposed in many of those systems are specific to the classifiers being built. For example, in the work of

[264] the features that the algorithms use were chosen by the authors, based on the different classes being considered (indoor, outdoor, city, landscape, sunset, forest, and mountain).

Other related approaches perform scene level classification based on regions obtained from automatic segmentation [243][171]. The configuration of regions in different scene classes is used during classification. A typical beach scene, for example, contains blue regions at the top (sky), and yellow regions at the bottom (sand). This type of information is used in a training stage, and the configuration of regions in new images is used to determine the images' class. The structure, however, is limited to the global configuration regions in the images, and structured object (or scene) models are not used.

A related approach for object classification uses body-plans [113] in the construction of object models. Specialized filters, for detection of naked people and horses, are used first to select relevant regions in the image. A search for groups that match the body-plan is then performed over those regions. Although this approach allows the construction of multiple-level composition models (like the *VA*), the system differs from the *VA* because it uses specialized algorithms (e.g., filters), and object models built by experts. Likewise, the approach in [114] utilizes a specialized face detection algorithm [220]

Some related work has also been done in the area of Computer Vision [134][109][118]. The main difference between the *VA* approach and previous work in Computer Vision is the role the user plays in defining objects, and the lack of constraints imposed by the *VA* system (e.g., no constraints on lighting conditions, etc.). Other differences range from the representation of the data (e.g., features used), to the learning algorithms, application domain, and operational requirements (e.g., speed, computational complexity). The discussion in chapter 2 outlines differences between VIR and object recognition.

The FourEyes system [185] learns from labels assigned by a user. User input, however, consists of labeling of regions (not definition of models based on multiple levels like in the *VA*). Although multiple feature models (for feature extraction) are incorporated in that system, different learning algorithms are not used.

In FourEyes the system begins by creating groupings[21] of data using different features (regions are grouped based on color, texture, etc.). The way in which features are extracted, in turn, depends on different models (e.g., texture features can be extracted using MSAR texture [212], or other models). The user then creates semantically meaningful groupings by labeling regions in images, and the system learns to select and combine existing groupings of the data (generated automatically or constructed manually) in accordance with the input provided by the user. This way, different features can be used for forming different

[21] A grouping in FourEyes is a “set of image regions (patches) which are associated in some way” [185].

semantically meaningful groupings (e.g., skies are grouped by color features, trees are grouped by texture features). For example, blue skies could easily be grouped using color and texture, but other objects (e.g., cars) may be better grouped using shape information. The underlying principles in the system are that models (i.e., used to extract the features used to form groupings) are data-dependent, groupings may vary across users, and groupings may also change depending on the task (e.g., the same user may want to group regions differently at different times). Therefore, the system aims to form groupings in many different ways (based on user input), and combine the appropriate groupings so that they represent semantically meaningful concepts (e.g., sky).

FourEyes receives as input a set of images to be labeled by the user. In the first stage, within-image groupings are formed automatically, resulting in a hierarchical set of image regions for each image for each model. For example, if image A contains trees and a house, the first stage may place all tree pixels in the same group (similarly for the house pixels; thus two different models are used, one for trees and one for houses). Pixels that were used in the tree grouping may also be used in a grouping of leaves (thus resulting in a hierarchical set of regions for each image). In the second stage, the groupings obtained in the first stage are grouped, but this time across the images of the database. In other words, for a given image, the groupings of the first stage are themselves grouped in the second stage with similar groupings of other images. For instance, trees from different images would be grouped together, and houses from different images would also be grouped together.

The first two stages in FourEyes are performed off-line. The third stage corresponds to the actual learning process in which the user labels image regions as positive (e.g., sky) and negative examples (not sky) of a concept. The system then tries to select or combine the existing groupings into compound groupings that cover all of the positive examples but none of the negative examples labeled by the user. The selection and combination of existing groupings constitutes the learning stage of the system, since the goal is actually to find (and combine) the models or features that group the regions according to the labels provided by the user. The groupings learned by the system, that cover the positive (and none of the negative) examples labeled by the user, are utilized to classify new regions. Regions that have not been labeled by the user, then, are automatically labeled by the system according to the examples the user has provided (and therefore the groupings the system has found to be adequate). In this stage (when the system labels new regions based on what it has learned), the user can correct the labels assigned by the system in an interactive process. The system then uses this information to modify the groupings it had learned, making learning incremental— the system learns constantly from user input, as opposed to learning in a single training session (referred to as batch learning; see [83]).

The approach taken in FourEyes has several advantages: (1) several models are incorporated; (2) user feedback is allowed; and (3) user subjectivity [211] is considered. Inclusion of several models is important, because as the discussion in [185][212] demonstrates, no single model is best for all tasks (e.g., different ways of extracting texture features may be useful for different types of texture). User feedback and subjectivity are also very important, because as the discussion in Section 3.2 showed, multiple levels of indexing can be performed. This, however, can be a drawback in the FourEyes system: finding the right concept, given the groupings provided by the user and those generated automatically, can be difficult. The system performs grouping based solely on low-level features, while the groupings (examples) provided by the user may be based on higher-level semantic attributes. One way to improve this is to require more structured examples (i.e., not just regions, but hierarchies), and to constrain the application domain so that only appropriate features are used in the different grouping stages. Some of these issues are addressed by VA framework presented in this chapter.

Another approach to detect events of scenes in specific domains consists of exploring the unique structures and knowledge in the domain. A system developed in [275] includes multiple models (for handling color variations within the same type of sport game— e.g., different colors of sand in tennis) and uses manually constructed region-level rules (for exploring the scene structure). High accuracy was reported in detecting batting scenes in baseball, and tennis serving scenes. This work differs from the *VA* framework in several aspects. In particular, that approach uses domain knowledge programmed by an expert (specific rules for baseball/tennis). In addition, it includes an initial filtering stage based on a global color measure. In other words, the video scene is first analyzed at the global level using color histograms, and then detailed scene analysis is performed. The detailed scene analysis differs in the two approaches (use of rules constructed by experts [275] vs. no expert input in the construction of classifiers in the *VA*). The initial filtering, however, could complement the *VA* framework (e.g., like in [275], a filtering stage could be used to select different detectors, to deal with variations across different games, as outlined in section 4.4.1).

Alternative models that represent objects and scenes in terms of their parts have also been proposed in the VIR community [110][248][192]. The definition of Composite Visual Objects [110], for example, is similar to the *definition hierarchy* of the *VA* framework, with the difference that classifiers in the *Visual Apprentice* are learned automatically. The authors of [107] propose the use of Object-Process diagrams for content-based retrieval. Although the representation is hierarchical there is no learning component. Finally, it is also useful to note the similarity between the definition hierarchy and structures used in MPEG-7.

4.1.3 Outline

The rest of the chapter is organized as follows. In section 4.2 we discuss structure and the application of machine learning in VIR. In section 4.3 we discuss the *Visual Apprentice* framework. In particular, we discuss user input, feature extraction and learning, and classification. In section 4.4 we present experimental results. A general discussion of important issues within the framework, and possible extensions are discussed in section 4.5. Finally, in section 4 we present eye tracking experiments and discuss ways in which eye tracking results can be integrated with automatic approaches such as the VA.

4.2 LEARNING AND STRUCTURE IN VISUAL INFORMATION RETRIEVAL

As discussed earlier, different users may have different interests, and those interests (for a single user) may vary over time [211]. Using this premise, it is preferable to construct systems that can adapt to users' interests. One possibility is to build systems that can adapt by learning from users. An important issue, therefore, in the application of learning techniques in VIR, is deciding where learning techniques are suitable.

4.2.1 Recurrent Visual Semantics

The concept of *Recurrent Visual Semantics* (*RVS*) is helpful in identifying domains in which to apply learning techniques in the context of VIR.

We define *RVS* as the repetitive appearance of elements (e.g., objects, scenes, or shots) that are *visually similar* and have a common level of meaning within a specific context. Examples of domains in which *Recurrent Visual Semantics* can be easily identified include news, consumer photography [141], and sports. In professional Baseball television broadcast, for example, repetition occurs at various levels: *objects* (e.g., players), *scenes* (e.g., a batting scene), *shots* (e.g., the camera motion after a homerun occurs), and *shot sequence structure* (e.g., a homerun shot sequence often includes a batting scene, a scene of the player running, etc.).

The existence of *RVS* motivates the approach of using learning techniques in Visual Information Retrieval. Using this concept, it is possible to identify domains in which learning techniques can be used to build automatic classifiers (for objects or scenes). The existence of repetition facilitates training, and the domain constrains the future data inputs to the classifiers learned. Once a domain is selected, identification of its repetitive (but semantically meaningful) elements increases the possibilities of successfully applying machine learning techniques in the specific domain. At the same time, application of learning within the domain (e.g., baseball video only, versus all types of video) decreases the possibility of errors.

4.3 THE VISUAL APPRENTICE

4.3.1 Overview

The *Visual Apprentice* (VA) framework consists of three stages: (1) user input, (2) feature extraction and learning, and (3) classification. In the first stage, the user explicitly defines object or scene models according to her interests, and labels training examples (images or videos). In particular, each example image or video is segmented automatically by the system, and the results of the segmentation are manually labeled by the user according to an hierarchy defined by the user. In the second stage, the system extracts features (e.g., color, texture, motion, etc.) from each image or video example provided by the user. Then it learns classifiers based on those examples producing an hierarchy of classifiers (a Visual Object or scene Detector— VOD). In the third stage, the classifier (the VOD) is applied to unseen images or videos. The Visual Object or scene Detector performs automatic classification by first automatically segmenting the image or video, and then combining classifiers and grouping image areas at different levels.

4.3.2 The Definition Hierarchy

Studies in cognition and human vision have shown that during visual recognition, humans perform grouping of features at different levels [174][70]. The highest level of grouping is semantic: areas that belong to an *object* are grouped together. An *object*, however, can be separated into *object-parts,* which consist of *perceptual-areas*: areas that we perceive categorically. Categorical perception refers to the qualitative difference of elements across different categories. Colors, for example, are often "grouped" [253][126]— we say the shirt is green, although it may have different shades of green. These different levels of grouping motivate the model-based approach to the construction of *Visual Object Detectors (VOD)*[22] in the *Visual Apprentice*. In this framework, a *VOD* is defined as a collection of classifiers organized in a *definition hierarchy*[23] consisting of the following levels (Figure 23): (1) *region*; (2) *perceptual*; (3) *object-part*; (4) *object* and (5) *scene*.

[22] The *detectors* we describe refer to objects and scenes. We use the name VOD, however, for simplicity and to emphasize the local structure of the classifiers.

[23] We have chosen only five levels for the hierarchy because they provide an intuitive representation that is useful in practice.

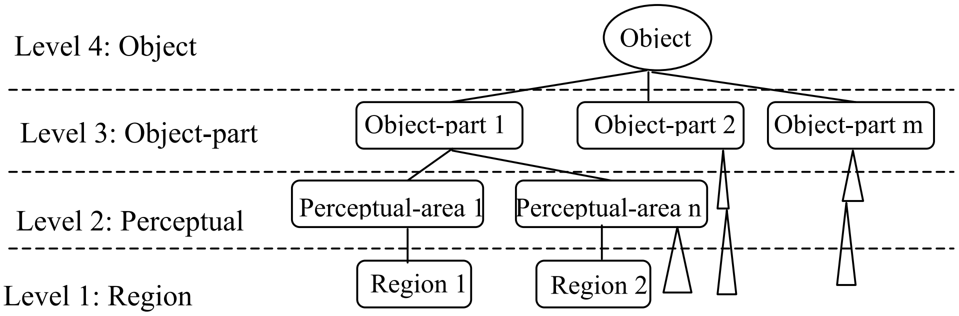


Figure 23. Definition Hierarchy. Note that a scene (not shown in the figure) is a collection of objects and corresponds to the highest level.

More specifically, a *definition hierarchy* is defined in terms of the following elements:

(5) *Scene:* structured[24] set of objects.
(4) *Object:* structured set of adjoining object-parts.
(3) *Object-part:* structured set of perceptual-areas.
(2) *Perceptual-area:* set of regions that are contiguous to each other and homogeneous within the set.
(1) *Region:* set of connected[25] pixels

According to this definition, every node *e* in our *definition-hierarchy* has a conceptual interpretation (e.g., "object"), and represents a set of connected pixels in an image or video. Nodes are image areas and arcs indicate parent-child relationships (from top to bottom)— a node is composed of all of its children. For example, in Figure 23 object-part1 is an area composed of n perceptual areas, each of which is composed of a number of regions.

[24] The word *structured* is used to emphasize the importance of spatial relationships between elements in the particular set.

[25] Regions, which are at the lowest level of the hierarchy, constitute the basic units in the framework and can be extracted using any segmentation algorithm based on low-level features such as color, texture, or motion.

In addition, the following restrictions are placed on the construction of *valid hierarchies* (please refer to Figure 23, where each node represents a set): (1) a set of level *i* (i≠5) is a subset of only one set of level *i+1 (e.g.,* an *object-part* can only belong to one *object*; a node in the hierarchy can only have one parent); (2) a set of level *i* cannot be a subset of a set of level *i-1*, unless the two sets are equal (e.g. an *object* cannot be part of a *perceptual-area*; a face can be equal to a single perceptual area); (3) sets at the same level are disjoint (i.e., intersection of two sets of the same level is empty; two *object-parts* cannot intersect); (4) *regions* do not contain subsets (i.e. *regions* are the basic units and cannot be separated); (5) No. sets at level i <= No. sets at level i-1; (6) all sets are finite and can contain one or more elements; (7) every set is equal to the union of its children.

Figure 24 shows a batting scene as defined by a user. Note that every node in an hierarchy has a conceptual meaning (e.g., pitcher), corresponds to an image area in a training example (e.g., a set of connected pixels in each image), and, as will be shown later, corresponds to a classifier (e.g., pitcher *object-part* classifier). The user, in this case, has decided to model the scene using only four levels (*region, perceptual-area, object-part*, and *object*): it is not necessary for a detector to have every level or every node of the hierarchy of Figure 23.

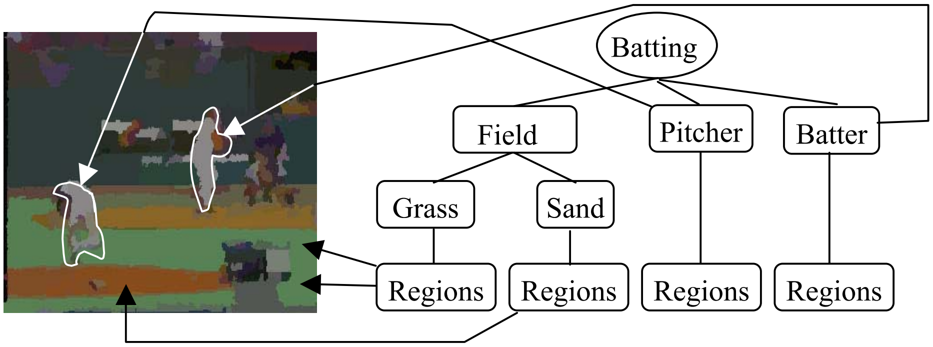


Figure 24. Automatically segmented Baseball image. This example shows how a scene can be modeled using the *hierarchy*. The white outlines were drawn manually to illustrate how the regions map to the hierarchy. Note that the user decided to model the scene using only four levels.

After the user defines an hierarchy and provides the training examples, features are extracted and classifiers are learned (stage 2). Classification (stage 3) occurs at the levels of Figure 23: *regions* are classified first and combined to obtain *perceptual-areas,* which are used by *object-part* classifiers. *Object-parts*, in turn, are combined and the results are used by *object* classifiers, etc.

The framework allows the construction of diverse hierarchies that depend on a specific user's interests and the specific application. The user can choose the number of nodes in the hierarchy and the number of training examples. The specific features and learning algorithms used depend on the implementation and application domain. The computational complexity of a particular VA detector will depend on the features, learning algorithms, and in some cases number of examples used to construct the detectors. In the best case complexity can be constant but it can be exponential or hyper-exponential depending on the factors just mentioned.

In the sections that follow we discuss, in detail, each of the three stages of the *VA* (user input, feature extraction and learning, and classification).

4.3.3 User Input

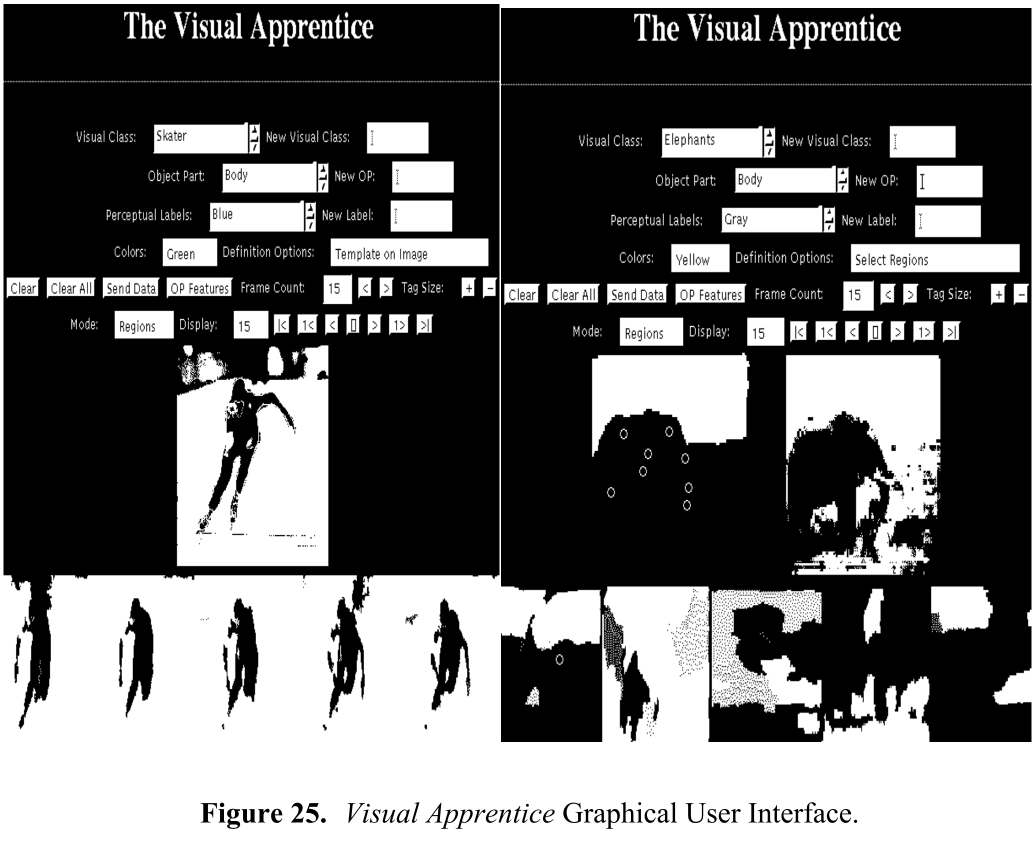
Different users have different interests. In order to accommodate this subjectivity we allow users to build different models (i.e., *definition hierarchies*) based on their individual preferences. The way in which *VODs* are constructed, therefore, is subjective and may vary between users (or for a single user over time). The main goal of this stage is to let the user construct a *Visual Object Detector*, without any low-level knowledge about features, or learning algorithms[26].

During training, the user performs the following tasks: (1) creation of a *definition hierarchy* by defining the labels to be used for each node; (2) labeling of *areas* (e.g., regions, perceptual-areas, etc.) in each training image or video according to the *hierarchy*.

Using the interface, the user defines the hierarchy by creating labels for nodes and expressing the connections between them. The label "batter region of batter

[26] In section 4.5 we discuss possibilities of additional user input (e.g., decisions on learning algorithms to use, etc.).

object-part," (Figure 24) for example, clearly defines the connections between the batter *region*, the batter *object-part*, and the batting *object*. Using a simple user interface (Figure 25), the user can set the corresponding labels (e.g., the "*object-part*" field would say "batter", the "scene" field would say batting, etc.). Once the labels are created, during training, the user labels *regions, perceptual-areas, object-parts*, and *objects,* in *each* training image or video. In the current implementation an image or video example corresponding to a particular hierarchy must contain all of the nodes defined in the hierarchy (e.g., all batting scene examples must contain a field, a pitcher, a batter, etc.).



Visual Class: Skater New Visual Class:
Object Part: Body New OP:
Perceptual Labels: Blue New Label:
Colors: Green Definition Options: Template on Image
Frame Count: 15 Tag Size:
Mode: Regions Display: 15

Visual Class: Elephants New Visual Class:
Object Part: Body New OP:
Perceptual Labels: Gray New Label:
Colors: Yellow Definition Options: Select Regions
Frame Count: 15 Tag Size:
Mode: Regions Display: 15

Figure 25. *Visual Apprentice* Graphical User Interface.

Labeling of image or video examples according to the hierarchy can be done in several ways: (1) by clicking on regions obtained from automatic segmentation, (2) by outlining areas in the segmented/original images or videos. Usually, labeling is almost exclusively done on the segmented images directly. Furthermore, in most cases it is only necessary to label individual regions (without outlining areas), because the interface of the *VA* facilitates training by automatically grouping regions that are connected and have the same label. The groups generated are assigned the label of the parent node of the regions used in the grouping. For example, in Figure 24, the user labels all of the pitcher regions (from automatic segmentation) with the name “pitcher region”. Then the system automatically groups all contiguous “pitcher regions” (those that are connected) and labels that group “pitcher *object-part*” (since the parent of the “pitcher regions” label is “pitcher *object-part*”). In some cases, however, the user may wish to manually outline *objects*, *object-parts* or *perceptual areas* (note manual outlines in white in Figure 24) and bypass the automatic grouping algorithm. The difference between using the automatic grouping provided by the system and manually outlining components is that manual outlining eliminates segmentation errors that would otherwise be incorporated. Again in Figure 24, note that in the segmented image a pitcher region contains some pixels that belong to the background. Manually outlining the pitcher eliminates that error, since the user drawn outline excludes those background pixels in the “pitcher *object-part*” example.

User input for video is similar since only the first frame, in each example video shot, must be labeled— in the segmentation algorithm used [274], regions are automatically segmented and tracked in each video frame. On that first frame, the user identifies regions that correspond to each node in her *definition hierarchy*: all of the sand *regions*, all of the sand *perceptual-areas*, *object-parts*, etc. The labeled region is tracked by the system in subsequent frames. For each region, then, it is possible to extract motion-related features (discussed below). It is important to note, however, that this is a simple approach to deal with video. A more sophisticated technique would consider all of the temporal changes in the definition hierarchy.

As a result of user interaction, we obtain the following sets for a defined class *j* (e.g., batting scene of Figure 24):

- Conceptual definition hierarchy: H_j.
- Example Element Set: EES_j = {{(e_{11}, l_{11}), (e_{12}, l_{12}), ..., (e_{1n}, l_{1n})}, ..., {(e_{k1}, l_{k1}), (e_{k2}, l_{k2}), ..., (e_{kp}, l_{kp})}, ..., {(e_{m1}, l_{m1}), ..., (e_{mq}, l_{mq})} where in each tuple, e_{ki} corresponds to the *i^th^* element (i.e., an area of a training image) of level k and l_{ki} is a label of level k associated with the element (e.g., (op_{31}, l_{31}) = (pitcher *object-part*, pitcher label)). Label level distinctions emphasize that labels must be different at different levels of the *hierarchy*. Regions in the example images or videos that are not labeled by the user are automatically assigned the label "unknown" and included in the set EES_j.

> This way, using the closed-world assumption [83], those regions can be used as negative examples during training.

Note that the user also has the option of including additional images or videos/regions to be used as negative examples, and that each image or video example for a given hierarchy must contain all the nodes in the hierarchy defined by the user. In other words an image or video example corresponding to a particular hierarchy must contain all of the nodes defined in the hierarchy (e.g., all batting scene examples must contain a field, a pitcher, a batter, etc.). As discussed in section 4.5, it would be possible to modify this constraint to provide further flexibility.

In the training stage, then, user input consists solely of defining the *definition hierarchy* (by creating the appropriate labels), and labeling example image or video areas according to the hierarchy. The labeling is done by clicking on image regions, or outlining image areas in each image or video example.

4.3.4 Feature Extraction And Learning

4.3.5 Feature Extraction

As discussed earlier, an element e_{ki} of our model (node in the *hierarchy*) is a set of connected pixels (i.e., an area of the image). Therefore, user input produces, for each example image or video, a set of image or video areas, labeled according to the *hierarchy* defined by the user. For each element in the Example Element Set, we compute a *feature vector*, which is an attribute-value tuple representation of the features of the element (e.g., color, shape, etc.). By computing feature vectors for all elements in the set EES_j, we obtain a training set of examples (attribute value pairs) for each class *j* (e.g., batting scene of Figure 24):

- $TS_j = \{\{(fv_{11}, l_{11}), (fv_{12}, l_{12}), ..., (fv_{1n}, l_{1n})\}, ..., \{(fv_{k1}, l_{k1}), (fv_{k2}, l_{k2}), ..., (fv_{kp}, l_{kp})\}, ..., \{(fv_{m1}, l_{m1}), ..., fv_{mq}, l_{mq})\}$ where fv_{ki} corresponds to the i^{th} feature vector element of level k and l_{ki} is a label of level k associated with the feature vector (e.g., (op_{31}, l_{31}) = (pitcher *object-part* feature vector, pitcher label)). Note that all examples for a particular node in the hierarchy (e.g., pitcher region) will have the same label.

Two types of feature vectors are used in the framework, those that contain *raw features*, and those that contain *spatial relationships* (described below). The raw vectors in our current implementation consist of a *superset* of 43 features. These features can be placed into five different groups.

- *Area and location:* area, bounding box center (x, and y), orientation, major axis length, major axis angle, minor axis length. [225]
- *Color:* average L, U, and V, dominant L, U, and V (LUV quantized to 166 colors [246]).
- *Shape:* perimeter, form factor, roundness, bounding box aspect ratio,

compactness, extent. [225]

- *Texture:* mean Maximum Difference, mean Minimum Total Variation (MTV), horizontal, vertical, diagonal, and anti-diagonal Mean Local Directed Standard Deviation (MLDSD), edge direction histogram (see [218][264]).
- *Motion trajectory:* maximum/minimum horizontal and vertical displacement, absolute horizontal/vertical displacement, trajectory length, displacement distance, average motion angle, average horizontal/vertical speed/acceleration.

Feature extraction occurs for all nodes, according to the hierarchy defined by the user. By computing feature vectors for each element, a *training set* is obtained for *every node* of the hierarchy. Recall that during user input (section 4.3.3), grouping occurs between regions that are connected and have the same label (e.g., in Figure 24 pitcher regions form a pitcher *object-part*; sand regions are grouped at the sand *perceptual-area* node). For each image example, when the grouping is performed (or a manual outline is used), a new area of the image is used for feature extraction. In other words, the features of the *regions* of the pitcher are used at the pitcher region node, but at the parent node (pitcher *object-part* in this case) a new set of features is computed from the image area that results from merging all connected pitcher regions together. The connected (labeled) pitcher regions, then, serve as a mask that is used to extract new features for the parent node of the region node used in the grouping (again in Figure 24, pitcher *object-part* for pitcher regions, sand *perceptual-area* for sand regions, and so on).

Elements of the hierarchy that are structured (i.e., *scenes, objects*, and *object-parts* in the definition hierarchy of section 4.3.2), and have more than one element (i.e., field *object-part* and batting *object* in Figure 24) are treated differently during feature extraction in the sense that they are characterized in terms of the elements they contain and the spatial relationships between those elements. For example, in Figure 24, the feature vector for the field *object-part* does not contain the 43 features discussed earlier. Instead, it contains two elements (grass, sand), and their spatial relationships. Note the difference between the following feature vectors:

Pitcher *region* = {label = pitcher, color = white, texture = coarse, etc.} (i.e., a region and its 43 features from the set described above)

Field *object-part* = {label = field_object_part, grass *perceptual-area* contains sand *perceptual-area*} (e.g., an *object-part* in terms of its perceptual areas and their spatial relationships)

To represent the structural relationships in structured sets that have more than one element, (e.g., between *perceptual-areas* within *object-parts*, or *object-parts* within *objects*, etc.), *Attributed Relational Graphs (ARG)* [209][186][226] are constructed. In an *ARG*, nodes represent elements and arcs between nodes represent relationships between elements. In the *VA*, nodes in the *ARGs* correspond to labeled elements from the *hierarchy* being considered, and arcs represent simple spatial relationships between those elements. In particular, the following relationships are used: above/below; right of/left of; near; far; touching; inside/contains.

It is important to note that using this representation an *ARG* will contain labeled elements only (see Field feature vector above), and their relationships. This is important because in the classification stage matching of graphs that contain unlabeled objects, which is a hard combinatorial problem, is avoided. Additionally, to avoid the difficulties of searching in complex relational representations (e.g., Horn clauses), the *ARG* is converted from its original relational representation to an attribute-value representation: [186] elements in the *ARG* are ordered (according to their label) and a feature vector is generated. With such transformation, existing learning techniques that use feature vector representations can be applied directly (e.g., decision trees, lazy learners, etc.).

The result of the feature extraction stage, then, is a set of feature vectors *for each node* of the corresponding hierarchy. Note that in the set TS_j the positive examples for a particular node are those feature vectors in TS_j that have the label for the corresponding node. The rest of feature vectors in the set TS_j are negative examples, for that node, under the closed-world assumption [83]. In essence, if there are *n* nodes in the hierarchy, there will be *n+1* different labels (including the "unknown" label) in the set TS_j. This means that there will be *n* different classes (one for each node), and therefore *n* different classification problems, each of which contains a set of positive and negative examples. This is important because it emphasizes that the result of the training stage is a set of *different classification problems*, one for each node.

4.3.6 Learning of Classifiers and Feature Selection

A *classifier* is a function that, given an input feature vector, assigns it to one of *k* classes. A *learning algorithm* is a function that, given a set of examples and their classes, constructs a classifier [103]. These two definitions are of extreme importance in Machine Learning and in particular in the framework of the *VA*. Using the labeled feature vectors, learning algorithms are applied at *each node* of the hierarchy defined by the user to obtain classifiers. This is done for each node at the five levels defined above: *(1) region, (2) perceptual, (3) object-part, (4) object* and *(5) scene.*

As depicted in Figure 26, all classifiers in an hierarchy could be constructed independently using a single learning algorithm. For example, it would be possible to choose one of the most widely used learning algorithms [83][186] (e.g., decision trees, lazy learners [65], neural networks, etc.) and apply it at each node to obtain the corresponding classifiers. The difficulty with this approach is that no algorithm will outperform (in terms of classification accuracy) all other algorithms in all tasks. In other words, since the *VA* is meant to allow users to define their own classes, it is not possible to choose, a priori, a learning algorithm that will produce classifiers that will always perform better than classifiers produced by any other learning algorithm. Of course, other factors could be considered (discussed further in section 4.5), including availability of resources (computational, number of training examples, etc.), speed requirements (during training and during classification), and desired accuracy.

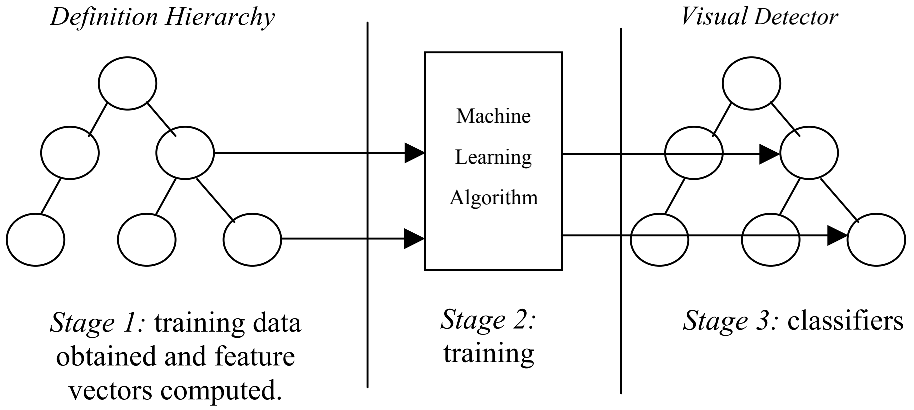


Figure 26. Overview of the learning process for each node in the hierarchy. A learning algorithm, applied to the training data for each node, produces a classifier for the corresponding node.

In order to allow flexibility, we propose a different approach (discussed in the next section), which consists of applying several *learning algorithms* to the same training data (at each node), to obtain a collection of binary classifiers for each node.

Regardless of the approach chosen to construct classifiers (using one or several learning algorithms), it is well known that selection of features can have a strong impact on classifier performance, even with learning algorithms that incorporate some form of feature selection. The justification and benefits in performance of selecting features in decision trees, for example, is given in [199][162][154]. This is because different features may be better for representing different concepts. For example, "good" features to represent a field might be color and texture, but good features to represent pitcher might be spatial location and aspect ratio (see Figure 24). Therefore, using the same features for all hierarchies (or for all nodes within a given hierarchy) may not yield the best results.

In many content-based approaches, specifically in interactive ones (query-by-sketch and query by example techniques [89][246][200]), users typically select the features to use. This, however, is often a difficult task. Automatic feature selection, used in the *VA* framework, serves to shield the user from the difficulties inherent in deciding which features are more important for a specific task (i.e., node in a hierarchy, or *VOD*). Given a set of features *A* (e.g., the *superset* described in section 4.3.4) with cardinality *n*, we wish to find a set *B* such that $B \subseteq A$, and where *B* is a better feature set than A [151]. The criterion for a "better" feature set *S* can be defined in terms of a *criterion function* *C*(*S*), which gives high values for better feature sets and lower values for worse feature sets. One possibility is to define the function as $(1-P_e)$, where P_e is the probability of error of a classifier. In such case, the value of the function *C*(*S*) depends on the *learning algorithm*, the *training* set, and *test* set used.

Since the goal is to find a set *B*, such that $B \subseteq A$, *feature subset selection (FSS)* [151] can be characterized as a search problem. The search for a better feature set can be conducted using several heuristics that aim to avoid exhaustively analyzing all possible feature sets. In particular, the search can look for optimal or sub-optimal results, and can be based on a filter or wrapper approach [159]. In the filter approach, the search for best features is independent of the learning algorithm and classifier that will be used. In the wrapper approach, which is used in the *VA*, the criterion function (1- P_e) is dependent on the *learning algorithm* and *data* used. Feature selection, therefore, is performed with respect to a particular algorithm and data set. In particular, a learning algorithm is repeatedly run on a data set using various feature subsets so that each run produces a classifier that uses a different set of features (Figure 27). The performance of the classifiers learned using each feature set is measured (using k-fold cross-validation, described below), and the best feature subset is chosen according to the performance. Once the features are chosen, the learning algorithm is used to construct a classifier using only those features. In the *VA*, best-first forward search [226] (a sub-optimal non-exhaustive technique) is used to find the best features.

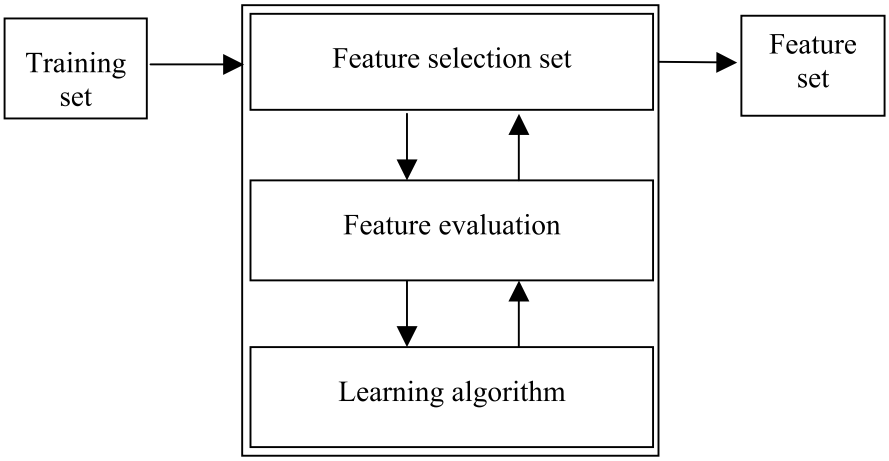


Figure 27. Feature selection in the wrapper model. Features are selected using different learning algorithms.

The search for a better feature set is performed, using a given learning algorithm, by building different classifiers using different subsets of the original feature set. Once the best feature set is found (for that particular algorithm), a classifier, using that algorithm is constructed. Since we use several algorithms (next section), it is then necessary to compare the different classifiers constructed by the different algorithms, at each node.

4.3.7 Selection and Combination of Classifiers

In the machine-learning community, an important goal is often to compare the performance of different learning algorithms [103][229]. The criterion in those cases is often the performance of the algorithms on standard data sets (e.g., UC Irvine repository [195]), or in particular domains. Instead of trying to find the best algorithms for classifying visual information, the goals in the *VA* framework center on determining the best *classifiers* for specific tasks (e.g., nodes of the *hierarchy*). The goal of the training stage is to obtain the best possible classifier (*not* learning algorithm) for each node. Since different learning algorithms may produce classifiers that perform differently on the same training data, the system simultaneously trains several algorithms (ID3, Naïve-Bayes, IB, MC4) [161] and selects the classifier that produces the best results. Note that, as discussed in the previous section, *FSS* is performed with respect to each algorithm, so when the system compares classifiers (this stage) it is already using the best features found, for each algorithm, in the previous step.

The performance of each classifier is measured using *k-fold cross-validation:* [160][189] the set of training examples is randomly split into *k* mutually exclusive sets (folds) of approximately equal size. The learning algorithm is trained and tested *k* times; each time tested on a fold and trained on the data set minus the fold. The *cross-validation* estimate of accuracy is the average of the estimated accuracies from the k folds. In the *VA* accuracy is determined as the overall number of correct classifications, divided by the number of instances in the data set. The process is repeated for each classifier being considered.

The best classifier is chosen according to its performance estimate given by the cross-validation accuracy: for each node, the classifier with the highest accuracy is selected. The process occurs for every node of the *hierarchy* defined by the user.

An alternative to selecting the best classifier is to combine all or some of the classifiers resulting from the cross validation process. In [146] other ways in which classifiers can interact in the *VA* framework were presented (also see [205] for a different combination strategy in a different framework).

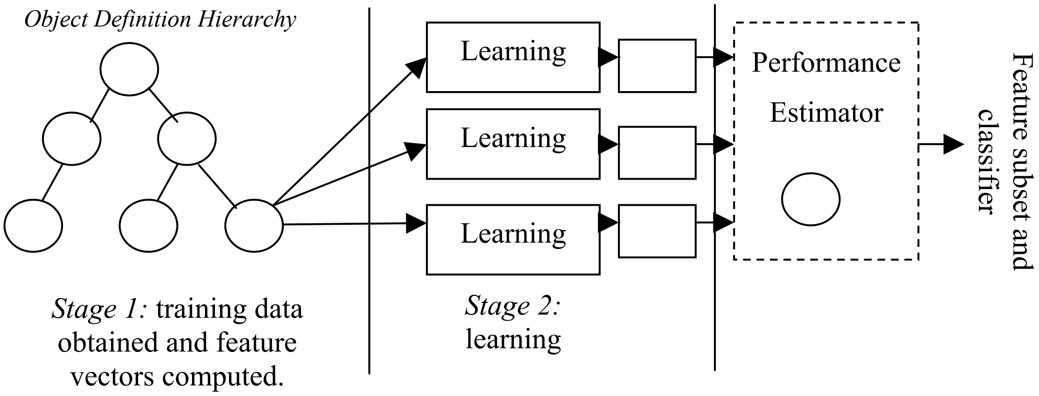

Object Definition Hierarchy
Learning
Learning
Learning
Performance
Estimator
Feature subset and classifier
Stage 1: training data obtained and feature vectors computed.
Stage 2: learning

Figure 28. Overview of the learning process for each node in the hierarchy. Note that each classifier is produced with the best feature set for the particular learning algorithm being applied.

4.3.8 Classification

When a *VOD* is applied to a new image or video, the first step is automatic segmentation of the image or video. Classification then follows the bottom up order of the *hierarchy* defined by the user (Figure 25). First, individual *regions* are classified (in a selection process similar to [253]) and, then, *perceptual-areas* formed (i.e., *regions* are classified *perceptually* and *groups* are found). Those *groups* are then combined to form prospective *object-parts*, which form *objects* that form *scenes*. Classification, however, depends on the specific hierarchy defined by the user. To detect a pitcher *object-part* (Figure 24), for example, the corresponding *VOD* would find pitcher regions first, and then try to find groups of pitcher regions that may correspond to the pitcher *object-part*. The process would be similar for grass and sand *perceptual areas*— regions are selected and groups of regions are used by the parent classifier (of the corresponding region classifier). Note that this is similar to the grouping performed by the system in the training phase (section 4.3.3). During training, *regions* are labeled by the user, so the system knows exactly which regions (those labeled) must be taken as a group at the parent node. In the classification stage, the labels are assigned by a region classifier. The classifier of the parent node, therefore, must search the space of possible groups.

In the first step, then, regions are selected by a region classifier. Given a universe U of elements, a function $c_j(i)$ is a classifier for class j that determines membership of i ($i \in U$) in the set j. In binary classifiers, $c_j: U \rightarrow \{0,1\}$ where, $\forall\ i \in U$, $c_j(i) = 1$ if $i \in j$ and $c_j(i) = 0$ if $i \notin j$. In fuzzy-classifiers [157] the function is not binary, but continuous, thus $c_j: U \rightarrow [0,1]$. In this case, j is a fuzzy-set since each element in j has been assigned a degree of membership in that set (e.g., if $c_j(i) = 0.75$ and $c_j(l) = 0.68$ we say that i is a *stronger* member of class j than l).

Region classification results in a set of region-membership tuples $R_{op} = \{(r_1, m_1, s_1), (r_2, m_2, s_2), ..., (r_n, m_n, s_n)\}$ where in each tuple (r_i, m_i, s_i), r_i is a *region* that belongs to the current *region* class with degree of membership m_i. The variable s_i is used to differentiate *weak* (s_i=0) and *strong* members of the class (s_i=1) where the value of s_i depends on m_i and is determined using a threshold over the region membership values. This is useful because weak isolated regions can be discarded.

We apply a grouping function g to R_{op}, to obtain a set $PG = \{g_1, g_2, .., g_n\}$ where every element g_i is a *group* of adjoining *regions* (e.g. *group* of pitcher *regions*). The goal then becomes to find the most likely group candidates from the set *PG* (e.g., determine which group of pitcher *regions* is more likely to be a pitcher *object-part*). Each group g_i may contain strong and weak *region* candidates, or just strong candidates. Groups of only weak regions are not considered because it is very unlikely for such groups to be important (e.g., unlikely pitcher *regions* are unlikely to form a pitcher *object-part*).

A search, then, must be performed over the space of possible *groups* of regions from the set *PG* to find the best possible ones. This can be treated as a classical search problem in Artificial Intelligence [118][226], and therefore, we can use heuristic techniques to reduce the search space. In particular, we use an Evolution Algorithm [184], treating each element g_i in the set *PG* as an individual in a population. Individuals evolve from one generation to the next through genetic operations such as mutation (an individual's characteristics are changed) and cross-over (two or more individuals combined to produce a new one). During the evolution process (generation to generation), only "strong" individuals survive—that is, individuals that meet certain fitness criteria.

What follows is a description of our algorithm:

1. Initialize population (P = PG).

2. Evaluate individuals in P using as a fitness function, the classifier of the parent node of the function used to select the regions to form PG. If the maximum number of iterations has been reached, or an element in P satisfies the criterion function, stop. Otherwise, continue to step 3.

3. Select and mutate individuals (e.g., remove a region from a selected group, etc.).

4. Go to step 2.

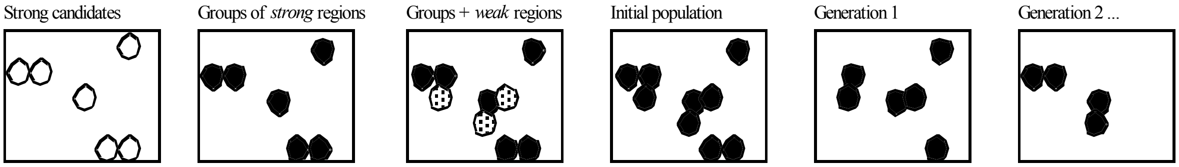


Figure 29. Evolution process.

To summarize, strong *region* candidates are first grouped and then merged with adjoining weak candidates. This eliminates from consideration isolated weak candidate *regions*. The evolution program then considers each *group* of regions. At every generation step, each *group* is mutated, thus generating a new individual. A new individual's fitness in the population is measured by the region candidate's parent node classifier. Note that each individual in the population (a group of regions) corresponds to an area in the image or video being considered.

Therefore, features (recall raw feature set of section 4.3.4) are extracted from that area, and the classifier that was learned during training is applied. Examining the example of Figure 24 again, region classifiers for the following nodes are applied: grass, sand, pitcher, and batter. The grouping using the evolution program is performed for each of these, and the groups are judged by the corresponding parent classifiers (grass and sand perceptual-areas; pitcher and batter object-parts). The field classifier, then, receives as input a feature vector that contains the grass and sand perceptual areas found (with their spatial relationships, as explained in section 4.3.4). The batting classifier, then, receives as input three elements and their spatial relationships (field, pitcher, and batter).

A final decision is made by the *Visual Object Detector (VOD)*, then, based on the decisions of all of its classifiers. In particular, all elements of the hierarchy must be present for a *VOD* to detect the object. For the batting scene of Figure 24 to be found, all elements must be found (i.e., pitcher, field and its parts, etc.).

4.4 EXPERIMENTS

Applying VIR techniques, and in particular those that use learning, in a real world scenario can be a challenging task for many different reasons. Therefore, we will describe some of the issues we encountered applying the VA, and experimental results. In each of the experiments reported in this section (baseball, handshakes, skies) the amount of time required to complete the training stage was less than two hours. Classification times varied across the different sets, but did not exceed 3 seconds on UNIX and PCs. The classifiers were constructed using the MLC++ software library [161] and the 43 features of section 4.3.5 (motion features were only used for Baseball video). We used the following learning algorithms [83][186][161]: ID3, MC4, Naïve-Bayes, and k-Nearest Neighbor (with k=1, 3, and 5).

4.4.1 Baseball Video

Television broadcasts of professional Baseball games were selected for these experiments because, as suggested by the concept of *RVS* of section 4.2, it was possible to identify meaningful objects and scenes that are visually similar and repeat. First, we identified the batting scene of Figure 24 as a meaningful candidate for a *VOD*. Then we collected and examined data.

In constructing a classifier for the batting scene of Figure 24, we encountered several issues of importance (see Table 15 discussed in previous work [147]). We divided such factors into those that are related to visual appearance (i.e., independent of the signal), and those that are related to the broadcast signal itself.

Table 15. Some quality factors found in professional Baseball broadcast.

Visual Appearance
Time of game (day: natural light, evening: artificial light)
Daytime weather (sunny, cloudy, raining, etc.)
Evening weather (raining, foggy, etc.)
Stadium (natural vs. artificial field, sand color)
Teams (color of uniforms)
Broadcast Network (camera angles, text-on-screen, etc.)

Signal Quality
Reception (from Cable TV, antenna, satellite, etc.)
Origin (live game, videotaped game)
Internal-network transmission (via satellite, etc.)
Analog recording (VHS, S-VHS, EP/SP mode, etc.)
Digital encoding (MPEG-1,2, parameters, etc.)
Noise, human error

These factors cause variations in the visual appearance of the *content* used by the algorithms, and therefore on the value of the *features* (segmentation, color, shape, etc.) used. The effect varies from minor to significant. For example, the time of day (morning, afternoon, night) can significantly affect the lighting conditions, which have an impact on the perception (and encoding) of color and texture. It is interesting to note that, due to the length of some Baseball games (several hours), it is possible to observe significant variations in weather (from sunny to overcast to rainy) and lighting (it is not uncommon for a game to start in the afternoon and end in the evening). Other elements, such as the players' uniform and the field remain constant within a game, but can vary significantly across different games. The way the games are broadcast (e.g., number of cameras, angles, etc.), on the other hand, is fairly standard within a game and across different broadcasters in a given country. Analyzing the data carefully, however, it is easy to observe variations that although minor for humans, can have a severe impact on VIR algorithms. Examples include "small" variations in camera angles (or camera distance), text on the screen, and others.

The second set of factors was also surprisingly important. Variations in the signal, even within the same game were sometimes significant. Colors changed, and noise was visible in many cases. Some of these are due to human error at origin, while others are related to the broadcast mechanism itself (live over satellite, etc.). Of course, variations in digitization of the signals can also have a strong impact.

For humans, most of those factors have no impact on the ability to recognize different scenes. Issues such as clutter (i.e., presence of unknown/unmodeled objects), occlusion, variations in lighting, and others, [138] are well known in Computer Vision and can have a strong impact on automatic algorithms. Most of these issues, however, are ignored in most of the experiments reported in VIR, mainly because in most cases the data used for training/testing comes from a single "collection" (e.g., a particular news source, etc.).

In order to test the framework with a batting scene detector, we used videos from several broadcasters. The set used in the experiments included games played in different stadiums (natural and artificial turf), times of day (night/day), weather conditions (overcast, sunny), etc.

Innings from 6 different games were digitized in MPEG1 format at 1.5 Mbps (30 frames/sec, at resolution 352x240 pixels). All scene cuts were obtained manually (scene cuts could be detected automatically using [183]), and each shot was forced to a length of 30 frames, which is long enough to represent a batting (pitching) action. The batting (or pitching scene) lasts approximately 1 second in a television broadcast.

A set of 376 baseball video shots (of 30 frames each) was used for the experiments. The set contained 125 batting scene shots (Figure 24). The set of 376 was divided into independent training and testing sets. The training set consisted of 60 batting scene shots. The test set, then, consisted of 316 shots (65 batting scenes and 251 other types of scenes), and *different* games were used in the training and test sets. The definition hierarchy used differed slightly from the one in Figure 24: the field *object-part* was divided into three perceptual areas: mound, top grass, and bottom grass.

Since classification starts at the region level, we examine classification results for different region classes. As discussed in section 4.3.7, different classifiers may perform differently on the same training data, and therefore it may be beneficial to use different algorithms for different classification problems. For our particular experiments, this is illustrated in Figure 30. The figure shows the learning curves for *region* node classifiers constructed using the ID3 algorithm [186], and a k-Nearest Neighbor classifier, also known as Instance Based (IB5 for k=5). In the *VA* framework, cross-validation accuracy is used, over the *training set*, to select the best features and classifiers. For illustration purposes here (in Figure 30 *only*), we show the learning performance over the *entire set* (training and testing sets together). The overall batting scene classifier *does not have access to the test set* during training, but we show it here because the differences between the curves are easier to observe than on the training set alone— the point of this discussion is to emphasize that an algorithm will perform differently on different sets of data. The curve for ID3, for example, suggests that an ID3 classifier will perform better on pitcher and grass nodes. An IB-5 classifier shows similar performance variations on different sets of data. At the same time, the plots show that the ID3 algorithm is more likely to perform better for the batter regions than the IB5 classifier. In the actual cross-validation experiments over the training set (not shown), different algorithms and features were selected for the construction of classifiers at different nodes (some examples are presented in [147]). Performance variations over the training set varied depending on the node, and most of the region level classifiers achieved around 80% accuracy on the independent test set.

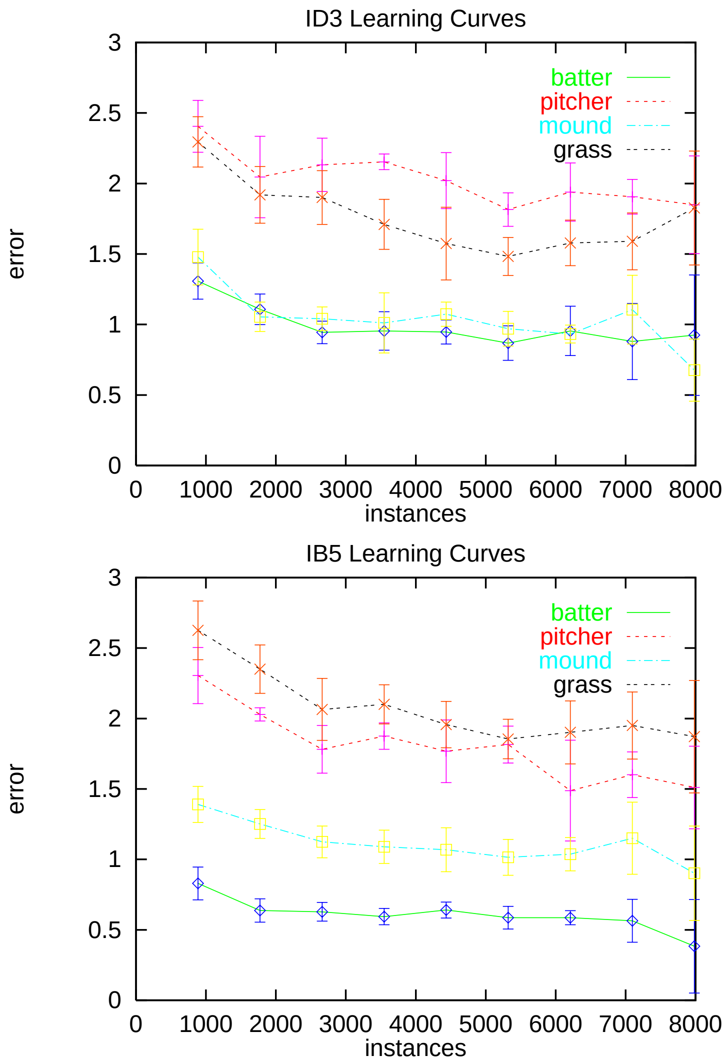


Figure 30. Learning curves that show number of training examples vs. error rate, for two different algorithms (top: ID3 and bottom: K-Nearest Neighbor, K=5) on the same set of data. The error bars represent 95% confidence intervals, and the error corresponds to the total percentage of misclassifications.

Detection of video shots of the batting scene (Figure 24) using the entire hierarchy resulted in an overall accuracy of 92% (overall % of correct classifications) with 64% recall, and 100% precision on the independent test set of 316 shots (65 batting scenes and 251 non-batting scenes). High precision was achieved in the *VA* in these experiments because the current implementation of the framework requires the detection of all nodes in the hierarchy. In other words, a batting scene can only be detected if all components of the hierarchy are found in the scene/shot. Therefore, a detector for this scene is unlikely to encounter false positives for all of the components of the hierarchy within a single scene. This mechanism, however, also causes a drop in recall: a classification error (miss) in one node of the hierarchy can cause a dismissal of a batting scene shot. In general, therefore, a hierarchy with more nodes is likely to yield higher precision and lower recall. Fewer nodes are more likely to yield lower precision and higher recall. Indeed, the shots that were missed by the classifier were missed because not all of the nodes were present. In particular, in most of the misclassifications the smaller elements (e.g., mound, batter) could not be found. This was due to segmentation errors, and errors at different levels of the hierarchy. In some cases text (and borders) surrounding the scene caused the errors— it is not uncommon for the entire scene to be reduced by a border with text (e.g., statistics or information from other games being played at the same time), making detection difficult without a pre-processing step.

Detection of the batting scene across different games (with the variations outlined in Table 15) is a difficult problem on which the *VA* has performed well. Preliminary experiments using global features (quantized LUV color histogram and coarseness) as well as block-based global scene classification (breaking up each image into 16 blocks, classifying the blocks and assigning the image the majority of the block labels [254]) produced poor performance. Although more experiments are required to compare the *VA*'s performance with other approaches (e.g., using the same features as in [254][264] and testing the implementation with a set of similar images), an analysis of the data and the preliminary experiments suggest that scene-level classification (i.e., not using structure information) may not yield good results for this particular problem.

One of the biggest difficulties is the variation that occurs across different games. The important components of the batting scene (i.e., those included in the hierarchy of Figure 24) usually occupy around one third of the image (scene). A global approach to classification, therefore, is likely to be affected by the remaining two thirds of each scene. Because of variations in the stadium, for example, the background (e.g., wall behind the pitcher) can be significantly different across different scenes. The *VA* framework takes this into account in the sense that if the background is not included in the hierarchy, it may not have a direct impact on the detection of the object or scene. A related observation is that, in this particular application, there are many similar scenes that do not match the model. There are many scenes that show a field and a crowd in the background. Such scenes, however, do not contain a batter (and pitcher), so a *VOD* that includes such elements would be able to differentiate between one of those "field" shots and a batting scene. It would be more unlikely for a global (or block-based) classifier, on the other hand, to be able to make such distinctions.

A possibility to alleviate the problem of variation present in the data, is to perform a filtering of the shots [275]. In that approach, incoming video scenes are first automatically assigned to a "color model" based on unsupervised learning (e.g., different models for different games— night, sunny, etc.), and a subsequent process uses manually constructed rules at the segmented region level (to account for local scene structure) to perform classification. Promising preliminary results (precision 96%, recall 97%) were reported in detecting batting scenes in broadcast videos [275]. However, note that unlike the *VA* the approach uses manually constructed region-level rules, and adaptive filtering that automatically selects the global color model depending on color variations of the new video. Indeed, it seems promising to incorporate the adaptive filtering as a pre-filter before applying the *VA* detector.

4.4.2 Handshakes and skies

We have also constructed classifiers for handshake, and sky images (see object hierarchies for handshakes and skies in Figure 31). For the handshake tests, 80 training images, and an independent test set of 733 news images were used. Out of the 733 images, 85 were handshakes. An overall accuracy of 94% (94% of the set of 733 images, were correctly classified) was achieved (74% recall, 70% precision) with 89 images automatically labeled as handshake by the system. Sky detection was performed on a set of 1,300 images that contained 128 skies (with an independent training set of 40 images, see [206]). An accuracy of 94% was achieved (50% recall, 87% precision), in a set of 134 images retrieved.

The results reported for the different types of hierarchies show that the Visual Apprentice framework is flexible, allowing the construction of different types of detectors. More importantly, performance in each case was similar to performance reported for similar classifiers using other techniques (overall accuracy around 90% and higher). The experiments show encouraging results for the construction of dynamic approaches to classification. Next we describe some possible extensions, and improvements.

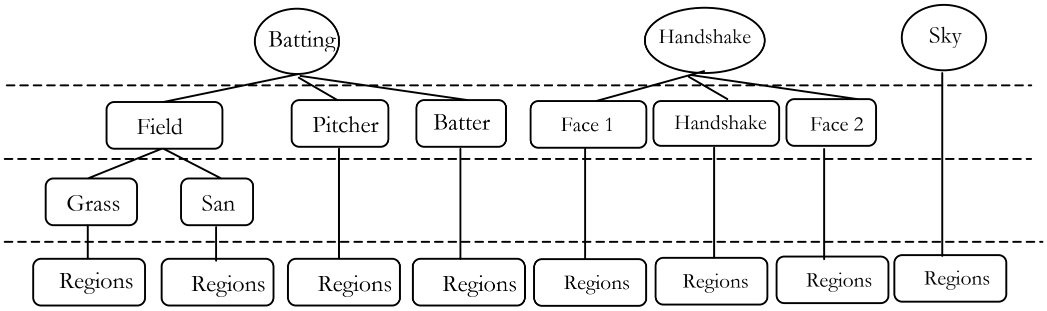


Figure 31. Example object definition hierarchies. The first hierarchy was not used in experiments, but shows how the close-up of a player could be modeled. The other two hierarchies were used in the experiments reported.

4.5 DISCUSSION

4.5.1 Extensions to the framework

The framework of the *VA* shows several desirable characteristics of VIR systems. The system uses learning techniques to automatically build classifiers, and therefore detectors can be easily constructed without the need for specialized algorithms. Since classifiers are built independently (for each node of the hierarchy), however, specialized algorithms can be easily incorporated. For example, in the handshake classifier, a face detection module could be used instead of the face node classifiers. Similarly, a domain-specific segmentation algorithm could be used to improve performance. In the current implementation a "standard" set of parameters is used with the segmentation algorithm. The parameters, however, could depend on the specific class (and hierarchy) being constructed by the user, or even learned by the system based on correct/incorrect segmentation results (labeled by the user).

The construction of the hierarchy, as discussed earlier, is subjective and will depend on the user. Therefore, two hierarchies for the same class (e.g., batting scene) may lead to different classification results. It is conceptually possible to build an hierarchy automatically, or semi-automatically. This issue is somewhat related to the learning of belief networks [205], and research in which the goal is to automatically detect Regions of Interest (ROIs). ROIs are areas that would roughly correspond to nodes in an hierarchy (i.e., areas of the image which are more important than others [214]). In this chapter, for example, experiments were presented to explore the use of eye-tracking results for automatic classification. Potentially this type of interaction could replace the current mode of interaction in the training stage of the *VA*, and help in the automatic or semi-automatic construction of hierarchies. A related issue is flexibility in the construction of hierarchies, and the application of the *VODs*. For example, instead of requiring all nodes to be labeled (and present during classification), it would be possible to extend the framework to allow the omission of nodes. A batting scene, then, could be detected (with a smaller confidence score), even if a pitcher is not detected.

Another possible extension of the *VA*, could include (at the user's expense) additional input parameters that could be used by the system to guide the training process. Information on issues such as desired computational efficiency (e.g., training/classification speed), for example, could be used internally in the selection of classifiers, and in providing training guidelines (e.g., size of training set, etc.).

Future work includes further research into classifier combination, semi-automatic hierarchy construction, and active learning for facilitating the annotation process. Other topics of future research also include feature and classifier selection with a small number of samples, the development of a theoretical framework for the hierarchical classification scheme we propose, and the inclusion of additional multimedia features (e.g., audio).

4.5.2 On Applications

With the *VA* it is possible to construct classifiers for objects/scenes that are visually similar *and* have a structure that can be clearly defined. Applications in sports commercial domains (e.g., databases of retail objects) seem promising. The approach, however, may be unsuitable for classes in which variation in visual appearance is too significant, or in which a well-defined structure is not easily identified (e.g., indoor, outdoor images). Although it is conceivably possible to build several disjoint hierarchies for such classes (rather than having a single one), it is likely for other approaches that have produced promising results (e.g., [254], [206]) to be more suitable.

It is also important to point out that in some domains, specialized algorithms may be better than flexible frameworks (e.g., the *VA*). An interesting possibility, however, is the combination of approaches like the *VA* with approaches that use expert knowledge. The *VA*, for example, could be used by an expert to construct rule-based classifiers, and those classifiers could be manually refined by the expert to improve their performance in a specific application. The framework could also be used to quickly examine feature variations for different types of objects (e.g., analyzing the training data), and to construct basic components to use in expert-constructed systems (e.g., use of a sky detector in a larger framework). The batting scene rules in the sports event detection discussed earlier, [275] for example, were constructed by an expert by analyzing features extracted by the *VA* during training. High accuracy was achieved in that system using this approach, suggesting that the *VA* framework can also be a useful tool for experts constructing domain-specific classifiers.

Some of the issues encountered in the *VA* framework are common to many VIR systems, particularly to those that use learning techniques. More specifically, we discussed some of the following issues: user subjectivity, feature and classifier selection, choice of training data, and problems that arise when applying such techniques in real-world scenarios.

4.5.3 Visual Apprentice Summary

We presented a new approach to the construction of dynamic classifiers for VIR. In the *Visual Apprentice (VA),* a user defines visual object or scene models, that depend on the classes in which she is interested, via a multiple-level *definition hierarchy* (*region, perceptual-area, object part, object*, and *scene*). As the user provides examples from images or video, visual features are extracted and classifiers are learned for each node of the *hierarchy*. At each node, the best features and classifiers are selected based on their performance, using k-fold cross-validation over the training set. The resulting structured collection of classifiers (a *Visual*

Scene/Object Detector) can then be applied to new images or videos. A new image or video is first segmented automatically, and then the classifiers (*region, perceptual-area, object part, object, scene*) are applied according to the hierarchy.

The concept of *Recurrent Visual Semantics (RVS)* was also discussed. *RVS* is defined as the repetitive appearance of elements (e.g., objects, scenes, or shots) that are *visually similar* and have a common level of meaning within a specific context. Using that concept, it is possible to identify where and when learning techniques can be used in VIR. We used baseball video as an example of the existence of structure and RVS.

Experimental results were presented in the detection of baseball batting scenes, handshake images, and skies. The results of the experiments are promising. The framework is flexible (users are allowed to construct their own classifiers, accommodating subjectivity); no input is required on difficult issues, such as the importance of low-level features, and selection of learning algorithms; and performance is comparable to that of other approaches. One of the main advantages of our framework is the flexibility of defining object or scene hierarchies and detailed user input at multiple levels. The approach allows users to specify multiple level composition models, which are absent in most existing approaches to VIR.

The framework presented in this chapter can be used to index visual content at several levels of the pyramid presented in chapter 3. The labels generated by the

*VOD*s can be used to index images or videos at the *generic scene* and *generic object* levels (please refer to Figure 16). Furthermore, the framework can be extended to index visual information at the *specific scene* and *specific object* levels. For example, a face recognition module can be used for a face node classifier to produce a label for the scene ("this is a picture of Bill Clinton") or a specific object in the scene ("the person in the image is Bill Clinton"). In addition, during the application of the *VOD*s syntactic features are extracted at the *local structure* level. Given the structure represented by the *VODs*, it is conceivably possible to also assign abstract labels to the images or videos, based on the combination of structured syntactic and semantic elements.

In the section that follows we study the viewing patterns of human observers in still images within and across different categories. This work is directly related to chapter 3 in that one of the goals with the eye tracking study is understanding visual information and users. Eye tracking patters are also strongly related to the construction of *VOD*s in relation to the training stage in the *VA* (labeling of regions) and the classification stage (selection of important regions and grouping).

4.6 STUDYING HUMAN OBSERVERS' EYE MOVEMENTS FOR DETECTING VISUAL STRUCTURE

In this section we present a study of human observer's eye tracking patterns when observing images of different semantic categories (i.e., handshake, landscape, centered object, crowd, and miscellaneous). We discuss ways in which the eye tracking results can be used in the *Visual Apprentice* framework.

Our hypothesis is that the eye movements of human observers differ for images of different semantic categories, and that this information can be effectively used in automatic content-based classifiers[27]. We present eye tracking experiments that show the variations in eye movements (i.e., fixations and saccades) across different individuals for images of 5 different categories: *handshakes* (two people shaking hands), *crowd* (cluttered scenes with many people), *landscapes* (nature scenes without people), *main object in uncluttered background* (e.g., an airplane flying), and *miscellaneous* (people and still lives). The eye tracking results suggest that similar viewing patterns occur when different subjects view different images in the same semantic category. Using these results, we examine how empirical data obtained from eye tracking experiments across different semantic categories can

[27] This is joint work with Jeff Pelz at the Rochester Institute of Technology and his students, Tim Grabowski, and Jason Babcock. Diane Kucharczyk and Amy Silver from R.I.T. also contributed.

be integrated with existing computational frameworks, or used to construct new ones. In particular, we examine how such results could be used in the *Visual Apprentice*. In the VA a user manually selects and labels regions to construct a classifier. Instead, it could be possible to automatically track a subject's eye movements as he observes example images from the category of interest and use the eye tracking results to automatically select regions for training.

Although many eye tracking experiments have been performed, to our knowledge, this is the first study that specifically compares eye movements across categories, and that explores the links between category-specific eye tracking patterns and automatic image classification techniques.

Overview

Eye movement traces of ten subjects were recorded as they viewed a series of 250 randomly interleaved images from the following categories: *handshake* (two people shaking hands), *crowd* (e.g., many people), *landscape* (no people), *main object in uncluttered background* (e.g., an airplane flying), and *miscellaneous* (people and still lives). We analyze, in the viewing patterns: (1) *within image variations* (similar/dissimilar patterns for an image, across several subjects); (2) *across image variations* (subject's pattern depends strongly on the image); and (3) *within/across image category variations* (similar/dissimilar patterns for images in the same category, across several subjects). In addition, we explore different ways in which these results can be used directly in the construction of automatic classifiers in a framework like the *Visual Apprentice*.

Related work

Eye tracking experiments have been performed for a many years. Several studies have examined eye movements of individuals as they perform natural tasks (e.g., [279][285][288]). Others have focused on the way humans observe pictures (e.g., photographs in [283], paintings in [294]). Many of these types of studies have been very useful in the development of theories of visual perception and recognition (e.g., [285][297]). For example, it has been suggested that humans move their eyes over the most informative parts of an image [294], and that eye movements (i.e., fixations and saccades, discussed later) are strongly influenced by the visual content of the image [282], and by the task being performed by the observer (e.g., describe image; search for an object [297]). No studies, to the best of our knowledge, have tried to compare differences in eye movements across different semantic categories.

Computational techniques that use information from eye movements include [292] and [293]. Unlike the work presented in [292], and [293], we focus on studying the *differences* in the way humans look at images *across* different categories, and on the usefulness of those differences to construct automatic classifiers for the same categories. The relation between computational (i.e., by computer algorithms), and human selection of Regions of Interest (i.e., areas of an image deemed important by an observer) was studied in [290]. Our work differs from [290], since our goal is to explore ways in which the results of category-specific eye tracking experiments can be used to construct classifiers.

Outline

In section 4.6.1 we discuss the motivation behind our approach. In section 4.6.2 we discuss important aspects of eye movements. In section 4.6.3 we explain our eye tracking set up. In section 4.6.4 we present the human observer experiments we performed, and in section 4.6.5 we discuss ways in which those results could be used in automatic classifiers (e.g., the Visual Apprentice). We conclude in section 4.6.6.

4.6.1 Why Study Eye Movements?

Eye tracking studies have been performed for many years (e.g., one of the earliest reported in [278]), for many different purposes. One of the main goals of such studies has been to understand the human visual system and, in particular, the visual process itself. It is now well understood that humans move their eyes, in part, because visual acuity falls by an order of magnitude within degrees of central vision [277]. Therefore, for some tasks, eyes must be moved to shift the point of regard to regions requiring high spatial resolution. Humans, however, also move their eyes to objects or regions of interest even when foveal acuity is not required by the immediate task. Because people move their eyes to targets of interest, monitoring the eye movements of observers can provide an externally observable marker of subjects' visual strategies while performing tasks such as manual image indexing or passive image viewing. Analyzing eye movements for these tasks can be useful in understanding how humans look at images, not only in terms of the recognition strategy used (e.g., which areas are observed and in which order; how much time person spends looking at certain types of objects, etc.), but also in determining what is deemed as important during the process (i.e., areas looked at are probably more important than areas not looked at).

Automatically classifying images (e.g., photographs) and video is an important task since it facilitates indexing, which allows searching and browsing in large image collections (e.g., images on the internet). Various computational approaches that perform automatic classification (mostly in the field of Computer Vision) have drawn on theories of the functionality of the human visual system [286]. In order to limit the amount of information to be processed, for example, some techniques detect *Regions of Interest* (*ROI*s) so that only regions that may be relevant to the problem at hand are selected for analysis. Therefore, understanding the selection performed by humans, and the visual process, is very useful in the construction of algorithms to perform classification.

In spite of the similarities between human processing and automatic techniques, most computational approaches are based on general theories that do not directly link the specific problem with the information obtained from experiments involving human subjects. For example, when we observe an image of two people shaking hands, perhaps we always move our eyes in a specific path to fixate on areas that we deem important (e.g., two faces, handshake). The areas that we observe, and the order in which we make those observations depend highly on the content of the image (e.g., handshake vs. landscape), and the task (e.g., recognize a person; find an object in the image). It may be possible, however, to find patterns in the way different individuals look at images in the same category. Nonetheless, information on how humans perform these specific tasks is seldom included in computational approaches. Analyzing the way humans look at images, however, could lead to important improvements in the construction of such classifiers because, if class specific observation patterns exist, decisions regarding the computational recognition process could be made based on data collected from human observers.

4.6.2 Eye Movements

The photoreceptor array in the image plane of the human eye (the retina) is highly anisotropic; the effective receptor density falls dramatically within one degree of the central fovea. The acuity demands of most visual tasks requires the high resolution of the fovea, but observers move their eyes to objects or regions of interest even when foveal acuity is not required by the immediate task. People make well over 100,000 eye movements every day. When humans move their eyes they either hold their gaze at a stationary point (*fixations*) or move them quickly between those fixations (*saccades*). While observers could gather a great deal of information from images while holding fixation, subjects free to view images without instruction regarding movements of the eyes typically make several eye movements per second. It has been known for some time that eye movement patterns are image dependent and to some degree idiosyncratic [277][287]. In addition to image-dependence, the pattern of eye movements and spatial distribution of fixation points also varies with the instructed task. Yarbus ([297]) demonstrated that subjects adopted dramatically different eye movement patterns when viewing one image when the instructions were changed. For example, the pattern seen under free-viewing was different when a subject was asked to remember the location of objects in the image, or to estimate the ages of people in the image.

The eye movements necessitated by the limitations of the peripheral visual field are driven by the scene and task, but in general make approximately three to four saccadic eye movements per second. In between those eye movements that are made to shift the point of gaze from one point in the scene to another, the retinal image must be stabilized to ensure high acuity. When the observer and scene are static, the eyes are stationary in the orbit, resulting in a static image projected on the retina. These *fixations* allow high acuity vision. When the observer and/or the scene are in motion, other mechanisms are necessary to stabilize the retinal image. A number of oculomotor mechanisms provide this stabilization. Objects moving through the field can be tracked with *smooth pursuit* eye movements. Large-field motion also elicits smooth eye movements to stabilize the image on the retina. Image motion due to movement of the head and body are cancelled by the *vestibular-ocular reflex,* which produces rotation of the eyes to compensate for head and body movements [284]. The *saccades* are rapid, ballistic movements that reach velocities of over 500 degrees/second. Saccades from less than one degree in extent to over 90 degrees are seen in subjects performing a number of tasks. The duration of the saccades varies, but are typically completed in less than 50 msec. Because of the speed with which the eyes move during a saccade, the retinal image is blurred during the eye movement. Subjects are not aware of the blurring caused during saccades because of a slight reduction in the systems sensitivity, but the effect is due primarily to a phenomenon termed *backwards masking*, in which the retinal image captured at the end of the saccade tends to mask the blur that would otherwise be evident.

4.6.3 Eye Tracking

Several methods can be used to track a subject's gaze. Several systems are in use today, each offering advantages and disadvantages. One system uses coils of fine wire held in place on the eye with tight-fitting annular contact lenses [291]. Eye position is tracked by monitoring the signals induced in the coils by large transmitting coils in a frame surrounding the subject. *Scleral coil* eyetrackers offer high spatial and temporal resolution, but limit movement and require the cornea to be anesthetized to prevent pain due to the annular contact lens. Another system offering high spatial and temporal resolution is the *dual-Purkinje* eyetracker [280]. Purkinje eyetrackers shine an infrared illuminator at the eye, and monitor the reflections from the first surface of the cornea and the rear surface of the eyelens (the second optical element in the eye). Monitoring both images allows eye movements to be detected independently of head translations, which otherwise cause artifacts. Another type of eyetracker is the *limbus* tracker. Limbus trackers track horizontal eye movements by measuring the differential reflectance at the left and right boundaries between the sclera (the 'white of the eye') and the pupil. Vertical eye movements are measured by tracking the position of the lower eyelid. While the limbus tracker provides high temporal resolution, the eye position signal suffers from inaccuracy, and there is significant cross-talk between horizontal and vertical eye movements. The class of eyetrackers used in this study illuminates the eye with infrared illumination, and images the eye with a video camera. Gaze position is then determined by analyzing the video fields collected at 60 Hz. Eye position data was collected with an Applied Science Laboratories Model ASL 504 Remote eyetracker. The system monitors eye position without any contact with the subject, an important factor to consider (Figure 32). The camera lens used to image the eye is surrounded by infrared emitting diodes (IREDs) providing illumination coaxial with the optical axis. The infrared, video-based eyetracker determines the point-of-gaze by using a video camera to extract the center of the subject's pupil and a point of reflection on the cornea. Tracking both pupil and first-surface reflections (i.e., on the cornea) allows the image-processing algorithms to distinguish between eye-in-head movements and motion of the head with respect to the eyetracker. This infrared/video eyetracker is limited to 60 Hz sampling rate and provides accuracy of approximately one degree across the field. The system automatically tracks subjects' head movements over a range of approximately 25 cm. Beyond that range, the tracker must be manually reset. The eyetracker signals such a track loss by setting the horizontal and vertical eye positions to zero.

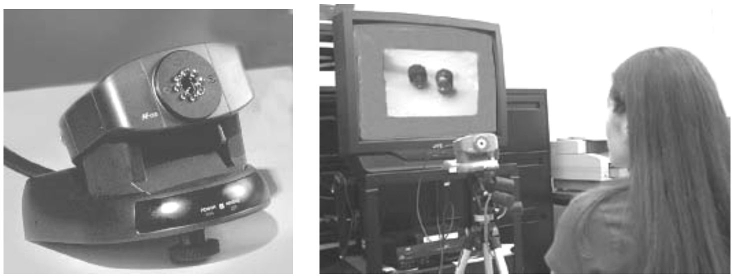


Figure 32. The 'remote' eye camera (left) is placed just below the subject's line of sight (right). The lens is surrounded by infrared emitting diodes to provide coaxial illumination.

While the retina absorbs most light that enters the pupil, the retina is highly reflective in the far-red and infrared regions of the spectrum. This phenomenon, which leads to 'red-eye' in photographs taken with a flash near the camera lens, produces a ‘bright-pupil’ eye image. In this image, the iris and sclera are the darkest regions; the pupil is intermediate, and the first-surface reflection of the IR source off the cornea is the brightest. The eye image is processed in real-time to determine the pupil and corneal reflection centroids, which are in turn used to determine the line-of-sight of the eye with respect to the head. Figure 33 shows an eye image captured with the ASL bright-pupil system. The image on the left shows the raw IR illuminated image; the image on the right shows the image with the superimposed cursors indicating pupil and first-surface reflection centroids determined by thresholding the image and fitting a circle to the pupil and corneal reflection. As the observer moves his eyes, the shape of the pupil reflection changes, and so does its centroid. The difference between the centroid of the pupil and the centroid of the corneal reflection (two points indicated in Figure 33) is used to determine the actual eye movement.

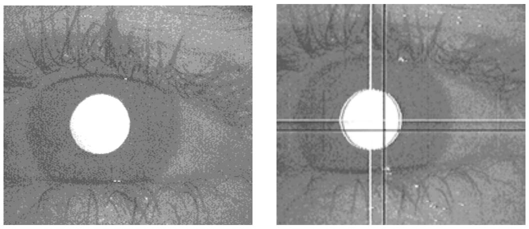


Figure 33. Image of the eye captured by the ASL eyetracking system (left); pupil centroid (white cross, right); and corneal reflection centroid (black cross, right)

Eye position is reported as a horizontal and vertical point of regard every 16.7 msec. The raw data is in arbitrary units based on display scaling, viewing distance, and subject calibration. The data is converted to image pixel units by scaling the output to the calibration points in pixel coordinates. The transformation corrected horizontal and vertical scaling, and offset the data to the center of the image display (i.e., [0,0] is the center of each image).

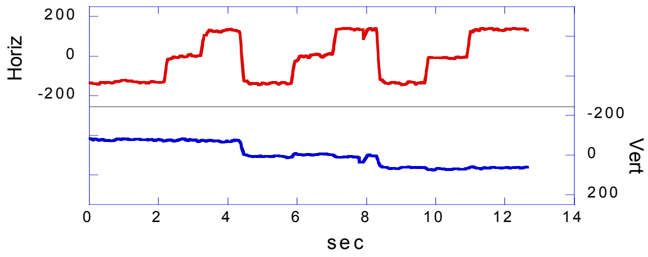


Figure 34. Horizontal and vertical eye position during a 9-point calibration sequence in image pixel units.

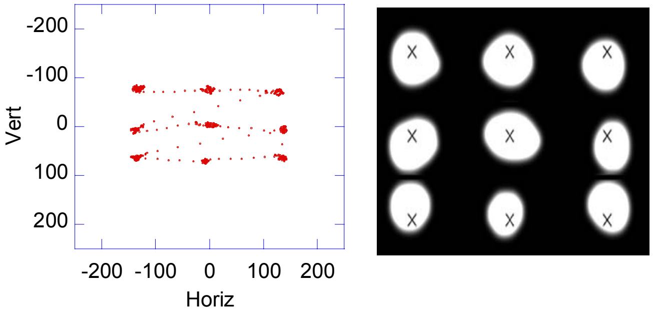


Figure 35. Horizontal and vertical position during the 9-point calibration sequence, in image pixel coordinates (left), and. fixation density mask overlaid on calibration grid (rescaled) (right).

Figure 34 represents the horizontal (top of the figure) and vertical (bottom of the figure) eye position of a subject reviewing nine calibration points (see also Figure 35). Calibration is necessary to determine the observer's position in space, and must be performed once for each subject as long as the general setup does not change (e.g., observer's physical location with respect to the camera). The fixation sequence was left-to-right, top to bottom. The repeated 'step' pattern in the horizontal trace shows the sequence of three horizontal fixation points on each line of the calibration target. The vertical record indicates the three rows that are scanned in turn. The zero-slope portions of the graphs (Figure 34) represent fixations on the calibration points; the transitions between fixations represent the rapid saccadic eye movements between the calibration points. Figure 35 (left) shows the same data as in Figure 34 plotted in two dimensions to show the spatial distribution of the scanpath. Horizontal and vertical eye position during the 9-point calibration sequence are seen scaled to image display coordinates. The fixation pattern can also be visualized as an image in which the lightness of a given pixel is proportional to the fixation density in that region. Figure 35 (right) shows the data from Figure 34 displayed as an image mask. The fixated regions of the calibration target are visible through the mask; the dark regions are image locations that were not fixated by the subject during the data collection, and the x's correspond to the actual nine calibration points.

4.6.4 Experiments

The goal of the eye tracking experiments was to determine whether individuals scan images of the same category in similar ways (and if there are differences across different categories)[28].

4.6.4.1 Image Data set

For the experiments we selected 50 color images from each of five different mutually exclusive categories (Figure 36): *handshake* (two people standing near each other, shaking hands); *main object in uncluttered background* (a prominent object around the center of the image, on an uncluttered background); *crowd* (cluttered scenes with many people); *landscape* (natural landscapes, without people); and *miscellaneous* (still lives, and people). The *handshake*, *crowd*, and *main object* images were collected from an on-line news service. The *landscape* and *miscellaneous* images were obtained from the collection of a photographer (the author). The main object category contained images without visible faces or people and had a variety of objects in the foreground (center) and background. No assumptions were made on the influence on viewing patterns of the *types* of objects in that category or the miscellaneous category. The landscape category also included only images without faces or people; the handshake category included only

[28] The images were selected and collected by the author and the eye tracking was performed at the Rochester Institute of Technology by his colleagues Jeff Pelz, Tim Grabowski, and Jason Babcock.

images in which people's faces and the corresponding handshake were visible. The miscellaneous category only included images that could not be placed in *any* of the other categories. All of the images used in the experiments had a resolution of approximately 240 x 160 pixels. Note however, that the important parameter is the angle subtended by each image (discussed in the next section).

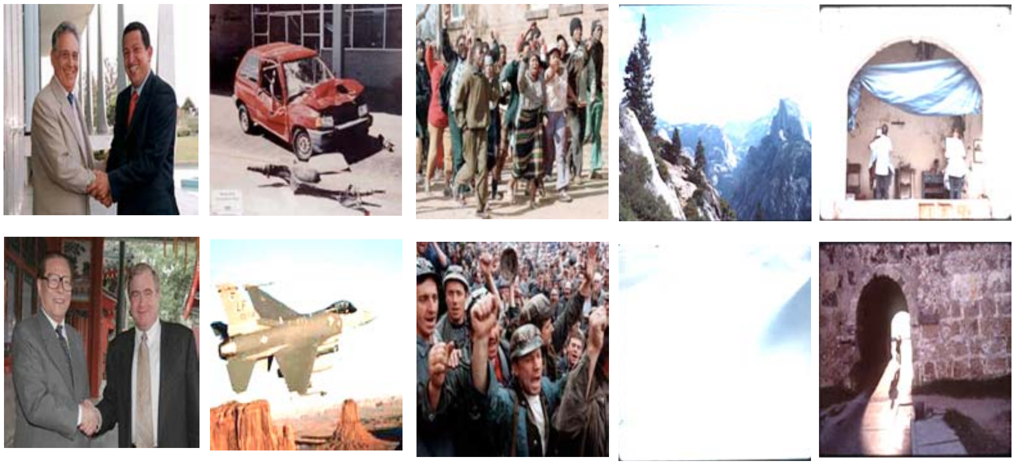


Figure 36. Example images from each of the five categories used in the experiments, from left to right: *handshake, main object, crowd, landscape,* and *miscellaneous*.

4.6.4.2 Subjects

Ten volunteers (four females and six males, all undergraduate students) participated in the experiment. All subjects were native English speakers, naïve as to the goals of the experiment, and had not participated in eyetracking experiments in the past. Before beginning the experiment, subjects read and signed an informed consent form describing the eyetracking apparatus. The observers were told to observe the images, but no explanations were given regarding the goal of the experiment, or the number of image categories. Since some of the images were obtained from a news source, it is possible that some of the subjects had familiarity with the persons and places in the photographs. However, no distinctions were made in this regard. The subjects viewed a total of 250 images. The images were interleaved in random order, and each was viewed for four seconds. The experiment was broken down into two sessions, each consisting of 125 images and lasting approximately 8.5 minutes. While subjects' heads were not restrained during the sessions, they were instructed to maintain their gaze on the TV display. Subjects took a self-timed break in between the sessions, typically lasting ~5 minutes before beginning the second set of 125 images.

The goal was to determine the primary areas of interest in the image, so a viewing time was selected that was sufficient to allow several fixations (typically 8-12 fixations), yet not long enough to encourage the subject to scan the entire image. Pilot experiments indicated that a four-second exposure was appropriate. For visual examination of an image, the important viewing variable is the visual angle subtended by the image. The angular subtense of the image was selected to approximate that of an observer viewing a photograph in a magazine or newspaper (e.g., ~15 cm wide image field viewed at a distance of 33 cm).

Subjects were seated about 1 meter from an NTSC television monitor with screen dimensions ~53 cm x 40 cm. The images subtended a mean of 25 x 19 degrees of visual angle (the exact value varied with head position, but was within 10% of the stated value). The remote eyetracker system was adjusted to be just below the line-of-sight to get the best view of the eye without obscuring any portion of the monitor (Figure 32). Thresholds for pupil and corneal reflection discrimination were set for each subject to optimize the thresholding that is used to determine the pupil and corneal reflection centroids, as seen in Figure 33. The system was calibrated to each subject by instructing the subject to fixate each point in a rectangular calibration grid on the display. The raw pupil and corneal reflection centroids recorded at each calibration point, along with the location of those points in image coordinate space, are used to establish the relationship between measured pupil and corneal reflection position and point-of-gaze in the image plane.

Figure 37 is a sample image from the 'miscellaneous' image class. Figure 38 shows the eye position as a function of time (left panel) and in two dimensions (right panel) for subject 6. While the pattern is less regular than the calibration set seen in Figure 34, it is clearly still made up of relatively long fixations separated by rapid saccadic eye movements. The two-dimensional plot also shows how the fixation patterns are tied to image content; while the eyes are moved across a broad area of the image, the majority of the trial is spent looking in a small number of image regions. The left panel of Figure 39 shows the fixation pattern superimposed over the target image, where each point represents gaze position at each video field (every 16.7 msec). The right panel shows the data mapped onto the image with one-degree circles indicating each fixation, defined as one-degree regions containing at least three data points (50 msec). Multiple fixations evident in the left panel result in larger indicators.

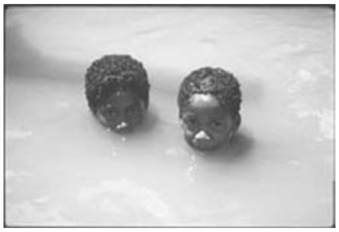


Figure 37. Example of image from 'misc' class (images were viewed in color)

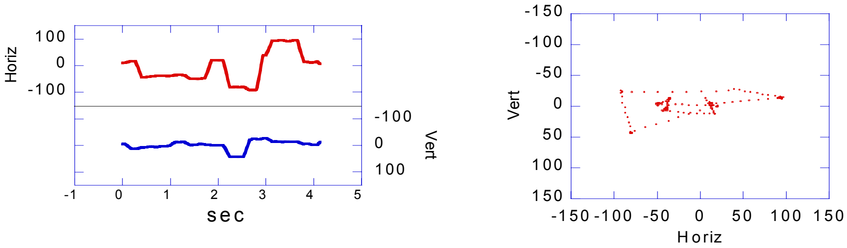


Figure 38. Eye position trace for one subject for the image in Figure 37 in time (left) and image pixel units (right). Each point on the right represents a 16.7 msec sample

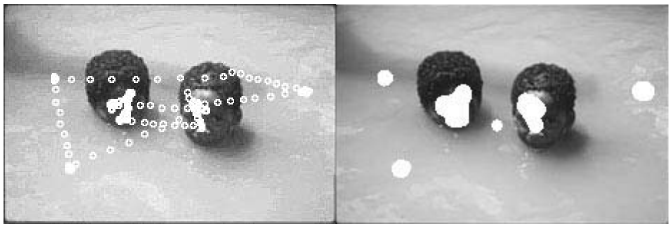


Figure 39. Scanpath overlay on display. Left panel indicates gaze position at 16.7 msec intervals. Individual fixations are indicated in the right panel with circles approximately one degree in diameter.

4.6.4.3 Eye Tracking Results

The experiments resulted in a fairly large amount of data, specifically eye tracking results for 10 subjects on 250 images in 5 categories. Since each subject viewed each image for approximately 4 seconds and the sampling rate of the eye tracker used is 60 Hz., we have an average of 250 data points per subject, per image, for a total of approximately 630,000 data points for the entire experiment.

It is possible to mathematically process the acquired data (e.g., mathematically compare fixations of different subjects within an image category, etc.), as discussed in section 4.6.5. In this section, however, we focus on our own observations about the viewing patterns observed (i.e., scanpath, including fixations and saccades). In particular, we discuss the following viewing patterns: (1) *within image variations* (similar/dissimilar patterns for an image, across several subjects); (2) *across image variations* (subject's pattern depends strongly on the image content); and (3) *within/across image category variations* (similar/dissimilar patterns for images in the same category, across several subjects).

In general, viewing patterns were similar between subjects viewing the same image, though idiosyncratic behavior was evident. Figure 40 shows fixation plots for four subjects as they viewed the same image. All four subjects fixated on the two main figures in the image, but each made a number of other fixations not common with the other subjects. This example shows an image for which consistent patterns were found, across different individuals. Note that in all cases the observers fixated on the people in the images, notably near the head.

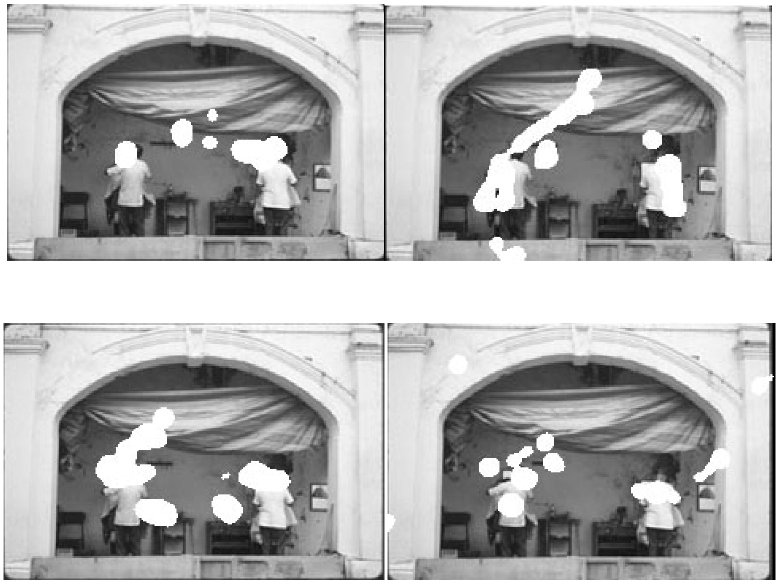


Figure 40. Fixation mask overlay for subjects 2, 3, 5, and 6.

Figure 41 shows similar data for a another image. Again, the fixation density plots for the four subjects are similar.

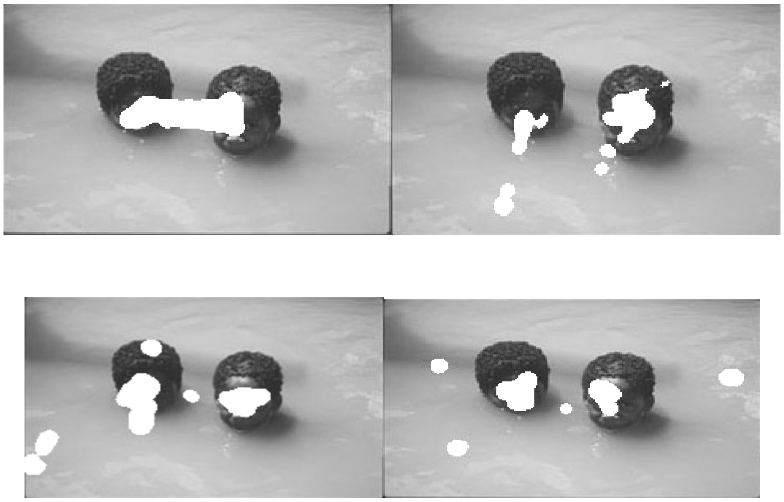


Figure 41. Fixation mask overlay for subjects 2, 3, 5, and 6. Note that all subjects fixated on the faces.

As the two previous examples suggest, there are cases in which it is possible to find *consistent viewing patterns*, across *different individuals*, for a *given image*. The viewing patterns themselves, however, varied, for a single individual, depending on the specific image being observed. For example, Figure 42 shows the fixation patterns of a single subject viewing four different images, highlighting the *strong image-dependency of viewing patterns*.

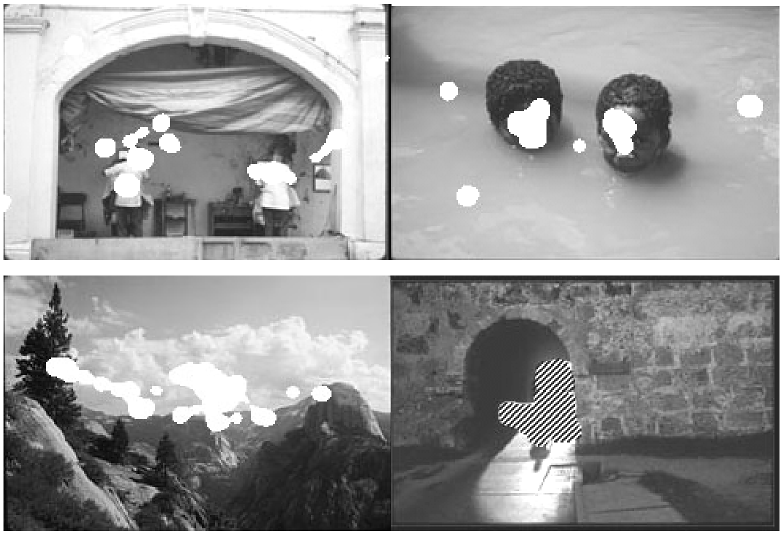


Figure 42. Fixation mask overlays for Subject 6 with (a pattern was used on the bottom right image to make the mask visible).

In some cases, it was also evident that wide variations in viewing patterns occurred, for *different subjects* viewing the *same image*. Figure 43 illustrates one of those images for which there was a wide variation. In terms of categories, we found the most consistency in the *handshake* and *main object* classes (Figure 44), while there was very little consistency in viewing patterns in the remaining three categories (*landscape, crowd*, and *miscellaneous*). Note that in Figure 44, for illustration purposes, we plotted *all data points* for *all subjects*, for *all images* within each of the two categories. No distinction is made between saccades and fixations in these plots, but fixation concentrations are readily seen where there is a higher concentration of points. Note, for example, that for the handshake class there are two visible clusters, resulting from the two faces that appear in each image in that class, and that *patterns are different across categories*.

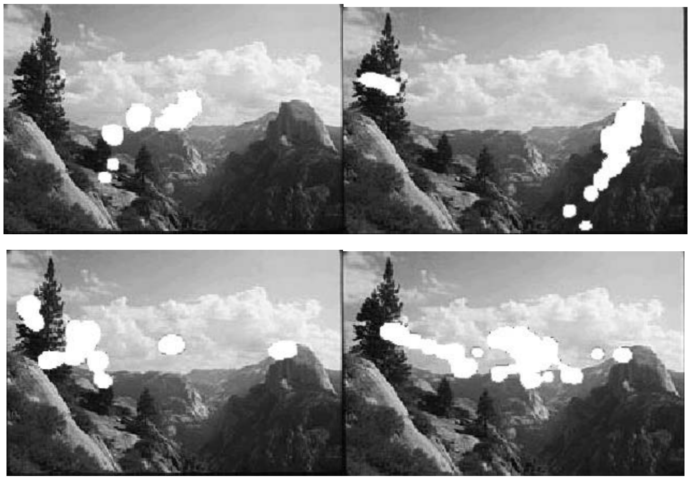


Figure 43. Dissimilar viewing patterns for a given image. Fixations of four subjects on the same image.

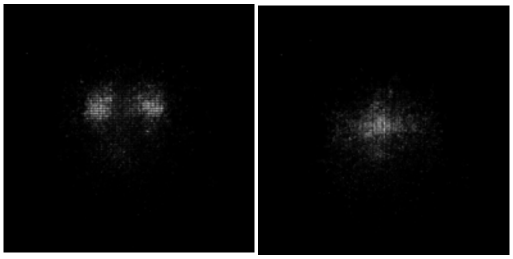


Figure 44. Data points for six subjects, for all images in the *handshake* category (left) and *main object* (right) category.

It is interesting to note that, in general, in the handshake class, subjects spent more time looking at the face on the left than at the face on the right. Additionally, it was somewhat surprising to find that in many cases the observers did not fixate (i.e., no data points occurred) at the handshake at all.

To summarize, we found *images with consistent viewing patterns* (several subjects viewed the same image in a similar way), *images with inconsistent viewing patterns* (several subjects viewed the same image in different ways), and *strong image dependence* (the same subject used different patterns on different images). In addition, we found the most consistent viewing patterns in the *handshake* and *main object* image categories, and that there were *differences across categories*.

The results suggest that human observers frequently fixate on faces (thus the patterns in the handshake category). When faces are not present, however, fixations are strongly influenced by composition, as the viewing patterns in the centered object and miscellaneous categories suggest. These results agree with the observations in [188] in which human subjects manually described images in semantic categories. The authors found that when people appeared in images the participants selected them as the most important cue in describing the images' category, and that color and composition play an important role in comparing natural scenes. The impact of differences between foreground and background objects was not examined in our work: in many of the images it is difficult to make such distinctions (see landscape and miscellaneous images in Figure 36).

4.6.5 Applications In Automatic Classification

In the previous section we discussed eye movement variations *for a subject*, *across images*, and *within/across categories*. One of the goals of this study was to determine whether results of eye tracking experiments like this one can be used in the construction of automatic classifiers. Therefore, data analysis across categories may be more useful because it could be used to construct classifiers for the classes studied (e.g., handshake).

In the Visual Apprentice, during training, the user manually clicks on regions that correspond to the *definition hierarchy* he defined. Therefore, one possible use of the eye tracking data for each class (instead of manual labeling), consists of using the fixation points (e.g., for all subjects for each image) to select the relevant training regions. In the handshake class, for example, we would expect the observers to fixate their gaze on the faces of the people shaking hands, and possibly on the handshake itself. A preliminary analysis of the data, however, showed that selecting regions for training that are obtained from automatic segmentation is not trivial.

In the current *VA* setup, the user manually clicks anywhere inside the regions that he wants to label, so a single pixel location (in x,y image coordinates) is sufficient to accurately select and label regions. Since a fixation point in the eye tracking experiments corresponds to several data points from the eye tracker (e.g., a fixation point might be defined as lasting 167 msecs, or 10 data points), those points must be clustered in some way and a fixation center (or fixation area) must be computed. Figure 42 shows an example of fixation areas that were obtained from the eye tracking results, and Figure 45 shows a set of image regions (obtained from automatic segmentation) automatically selected by the fixations points of 6 of the subjects. In other words, regions obtained from the automatic segmentation, that overlap (on the image) with fixations, are selected and shown in Figure 45. As the figure shows, some of the relevant regions are not selected (i.e., those that would be selected by a human training the system, like the handshake regions missing from the image in Figure 45), while some irrelevant regions are selected. Using the eye tracking results *directly*, therefore, may not be sufficient for region selection. Nevertheless, if viewing patterns are found within image categories, it is possible to train classifiers using the fixations (with additional processing to address the issues just described) to select important areas to be used during training. These areas need not correspond to regions obtained directly from automatic segmentation. In [289], for example, the authors compared automatically selected regions of interest with regions selected by human observers' eye movements, and proposed a technique to cluster fixations and compare them across different viewers.

In our experiments it is possible to apply the same approach to cluster points, but decisions regarding the use of the data (e.g., clustering algorithm, criteria to group fixations of different observers, etc.) are not trivial and can have a strong impact on the construction of automatic classifiers. One of the factors is that humans may fixate on certain areas of objects (e.g., a person's hair), but those areas may not yield the best results in terms of selecting the relevant regions for a particular detector (e.g., a face detector).

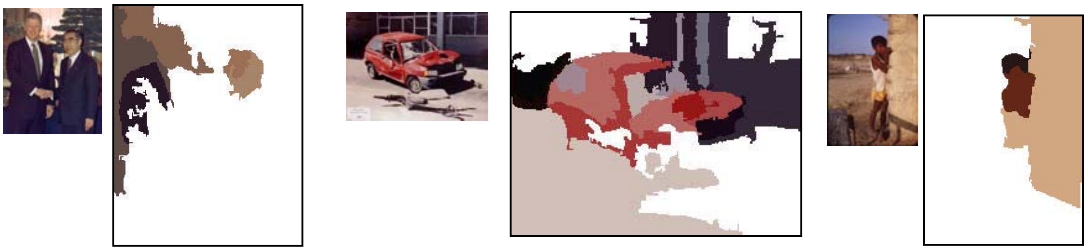


Figure 45. An example of the regions selected by the fixations of the human observers.

In addition to selecting regions automatically, based on fixations, it is possible to modify the algorithm of the *VA* and use the additional information provided by the eye tracking experiments. In that case, fixations could be used to give certain regions more weight than others (this could be easily included in the *VA* framework), and more importantly to also include regions selected by the saccades. Furthermore, scanpath order could be included in the classification strategy (i.e., classifiers would be applied according to scanpath order).

Alternatively, entirely new regions could be extracted from the training data (not using automatically segmented regions, but instead masks produced by the eye tracking experiments) and used to construct the classifiers. The actual classification approach, in that case, would also have to be modified so at the classification stage the regions would be extracted according to the training data (instead of extracting regions without any class-specific knowledge and then trying to classify them).

4.6.6 Eye Tracking Summary

We presented eye tracking experiments that show the variations in eye movements (i.e., fixations and saccades) across ten different individuals for color photographs of 5 different categories: *handshakes* (two people shaking hands), *crowds* (cluttered scenes with many people), *landscapes* (nature scenes without people), *main object in uncluttered background* (e.g., an airplane flying), and *miscellaneous* (people and still lives). In the viewing patterns we found the following: (1) *within image variations* (similar/dissimilar patterns for an image, across several subjects); (2) *across image variations* (subject's pattern depends strongly on the image); and (3) *within/across image category variations* (similar/dissimilar patterns for images in the same category, across several subjects). Specifically, we found more consistent patterns within the *handshake*, and *main object* categories. More importantly, we found the patterns were different between those categories (*handshake/main object*).

Using results from the experiments, we discussed ways in which this type of data can be used in the construction of automatic image classifiers in the *Visual Apprentice*.

The results of the experiments are encouraging since the existence of patterns allow eye tracking data to be used in the construction of automatic classifiers. In future applications it is feasible to consider a scenario in which a system learns classifiers directly from the viewing patterns of passive observers. Analysis of the data (e.g., selection and clustering of fixation points, use of scan order, etc.), however, plays a very important role since the criteria used can have a strong effect on the classifiers being built. Our future work includes more analysis of the data, and construction of automatic classifiers using these eye tracking results.

In chapter 5 we integrate some of the concepts of the previous chapters within the consumer photography domain. In particular, we develop an approach to cluster images based on composition (level 4 of the pyramid in Figure 16 of chapter 3) and develop a framework for the detection of non-identical photographs.

5 ORGANIZATION OF PERSONAL PHOTOGRAPHY COLLECTIONS

5.1 INTRODUCTION

In this chapter we address the problem of integration of generic visual detectors in solving practical tasks in a specific domain[29]. In particular, we present a system for semi-automatically organizing personal consumer photographs and a novel framework for the detection of non-identical duplicate consumer images.

In the multimedia applications of the future that we envision, consumers will more actively use their personal photography collections in exciting and innovative ways. Such applications will be driven by the core technologies that we develop today to deal with the fundamental problems of visual information organization.

[29] The work in this chapter was performed in conjunction with Alexander C. Loui and his colleagues at Kodak. Ana B. Benitez from Columbia University contributed to the clustering experiments and Enrico Bounanno, also from Columbia worked on the development of the STELLA interface.

In the consumer domain one of the most important goals should be to develop systems that *help* users subjectively manipulate and organize their own personal collections. Regardless of what those future applications will be, it is clear that certain trends will continue.

Consumers of the future will, undoubtedly, produce exponentially larger amounts of digital imagery. Consider the emergence of new applications and devices for digital visual memory [167] and wearable computing. Already many cellular telephones and other portable devices come with digital cameras, therefore personal collections are rapidly growing.

Two immediate implications, which we address in this chapter, clearly arise: (1) people will continue to want to organize their personal photography collections, and (2) the ease with which images are created will increase the number of *similar* images that people produce (Figure 46).

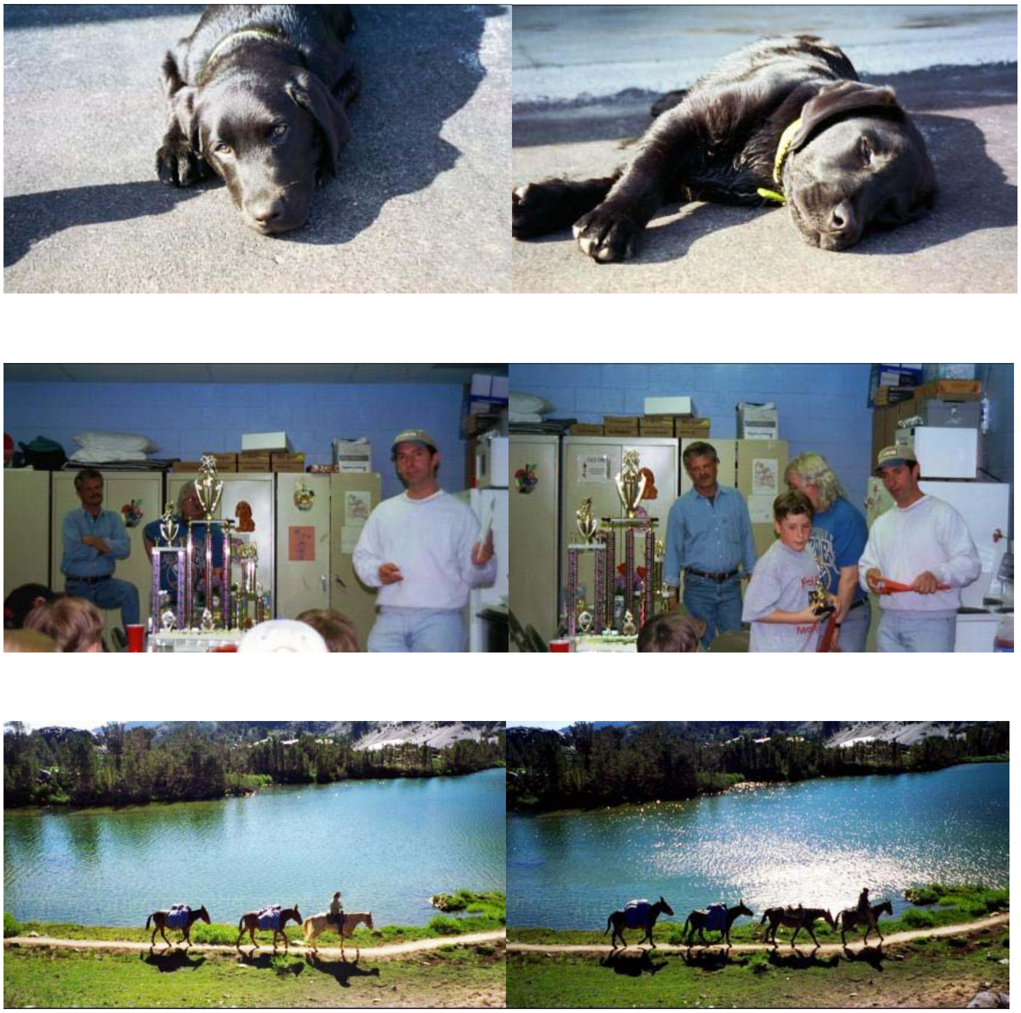


Figure 46. Non-identical duplicate images (top two rows) and similar non-duplicates (bottom row).

Developing techniques to help users organize their personal collections is clearly a goal many are trying to achieve. One important problem which has not yet been addressed, however, is the construction of computational frameworks to specifically help users organize similar images. These very similar images (non-identical duplicates) are important because, as we will show, they occur frequently in current consumer collections, and often they are made when important events are photographed (e.g., the group portraits we usually make one after the other).

In this chapter we use many of the concepts and techniques of the previous chapters to construct a specific application in which the fundamental issues of organization of personal digital collections play a key role. In particular, we present a system for semi-automatically organizing personal photography collections and a new framework for automatically detecting non-identical duplicate photographs.

We present a comprehensive study on a database of non-identical duplicate consumer images. This study is the first one of it's kind (on non-identical duplicates) and this is the first time the problem of non-identical duplicate consumer photographs is introduced. The computational framework we propose to detect non-identical consumer images is new and important because it integrates our knowledge of the geometry of multiple images of the same scene, generic visual detectors built with a flexible computational framework, and domain knowledge about the duplicate detection problem we are addressing.

5.1.1 Overview

In *STELLA (Story TELLing and Album creation application,* Figure 47), a system we have built for semi-automatically organizing consumer images, photographs are input into the system and automatically clustered hierarchically based on visual similarity and sequence information (i.e., the order in which they were made which corresponds to their sequence location in the roll of film). The resulting clusters are presented to the user so that he can subjectively organize his personal digital collection. Functionality in *STELLA* allows the user to easily browse the cluster hierarchy, select or modify individual clusters or groups of clusters, and add metadata to individual images or clusters at the levels of the pyramid of chapter 3.

One very important problem in a consumer photography application like *STELLA* is the existence of images that are considered "duplicates." These are images that are *not* identical, but that are very similar, such as those depicted in Figure 46.

Almost everyone who photographs has at least once (and most likely many times) made more than one photograph of the same *scene*, *object*, or *event*. Typically these duplicates are created in special occasions, for the most part because the photographer does not want to "miss the moment." We want to make sure we have at least one of those photographs.

Consequently, when we organize our images one of the first steps is to examine those non-identical duplicates either to select the best one or simply to put them in the same group and continue organizing the rest.

Duplicates should be in the same cluster

User can modify clusters and add metadata

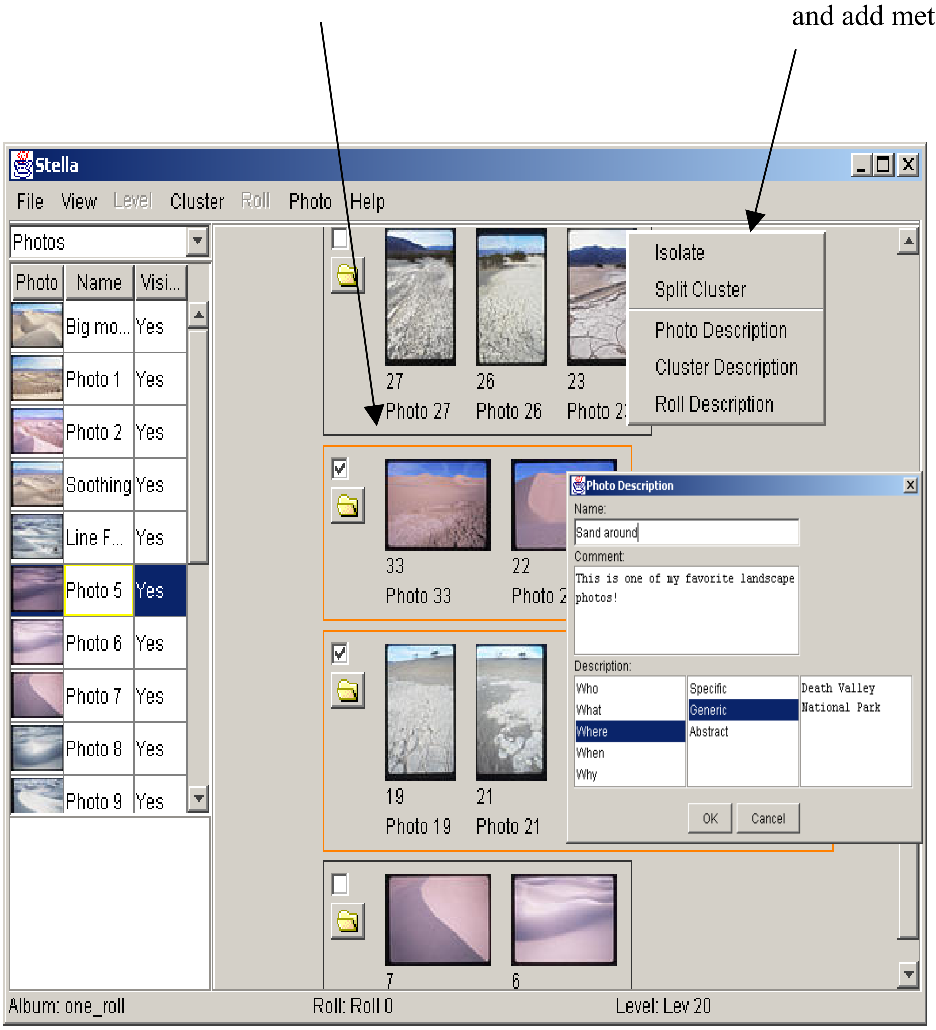


Photo	Name	Visi...
	Big mo...	Yes
	Photo 1	Yes
	Photo 2	Yes
	Soothing	Yes
	Line F...	Yes
	Photo 5	Yes
	Photo 6	Yes
	Photo 7	Yes
	Photo 8	Yes
	Photo 9	Yes

Name: Sand around

Comment: This is one of my favorite landscape photos!

Description:

Who	Specific	Death Valley National Park
What	Generic	
Where	Abstract	
When		
Why		

Figure 47. STELLA application

In Figure 46 we show two sets of images from our duplicate database. The images in this example were consistently labeled as "duplicates" (top) and "non-duplicates" (bottom) by ten participants in the experiments we present in section

5.4. As the example shows, the differences between "duplicates" can be visually significant, and the similarity between "non-duplicates" can be very high.

These characteristics of "duplicates" and "non-duplicates," discussed in more detail below, requires us to build a computational framework that is specific to the duplicate detection problem in consumer photography. Clustering techniques that use standard similarity metrics in *Visual Information Retrieval* are not sufficient.

Duplicate detection is necessary in current systems along with other basic functionality and could be used in innovative ways: for example, we can argue that duplicates are made only when there are important events and therefore could be used to automatically find those meaningful moments. It is very likely that if you retrieve all of the duplicate candidates in your own personal collection you will find many of the events that were most significant at the time the photographs were made.

In *STELLA* we introduce a novel image composition feature based on automatic region segmentation and a novel but simple variation of Ward's hierarchical clustering algorithm. In our clustering method, each photograph's location in the film is used in addition to visual content to create the clusters.

We develop a model of non-identical duplicate consumer photographs and introduce a new classification of different types of duplicates. Then, we introduce a novel framework that automatically distinguishes between non-identical duplicate and very similar non-duplicate images. Our approach is based on a multiple strategy framework that combines our knowledge about the geometry of multiple views of the same scene, the extraction of low-level features, the detection of a limited number of semantic objects, and domain knowledge. The approach consists of three stages: (1) global alignment, (2) detection of change areas, and (3) local analysis of change areas.

We present a novel and extensive image duplicate database[30] (255 image pairs from 60 rolls from 54 real consumers, labeled by 10 other people). We analyze labeling subjectivity in detail and present experiments using our approach.

5.1.2 Generic Objects in the Consumer Domain

Recurrent Visual Semantics (chapter 4), the repetitive appearance of elements (e.g., objects, scenes, or shots) that are *visually similar* and have a common level of meaning within a specific context, occurs in consumer photography at several levels. For example (Figure 48), in photos of a trip to New York, it is common to find photographs of the statue of liberty. Likewise, birthday photographs taken by different individuals often include similar scenes.

[30] The duplicate database we use was created from actual consumer photographs and labeled by researchers at Kodak. The analysis of the data was done by the author.

Let us consider a particular roll of film and determine, using the pyramid of chapter 3, what kind of repetition at the semantic level we can identify.

- *Generic Object:* trees, cars, buildings, etc.
- *Specific Object:* family members, friends, etc.
- *Generic Scene:* photograph on a beach, etc.
- *Specific Scene:* a specific scene repeats.

As we argued in chapter 4, such repetition lends itself to the application of flexible computational frameworks that learn to detect objects and scenes. Given enough training data, it could be very feasible to construct a library of detectors for consumer domains, perhaps specific to particular cities, locations within cities, or even individuals. Imagine a scenario in which your software, or your digital camera automatically "learns" from all of the photographs you make.

Alternatively, consider constructing a set of generic visual detectors using the framework of chapter 4 and applying them in systems for semi-automatically organizing personal image collections.

In this chapter we use the concept of *Recurrent Visual Semantics* to identify objects that repeat in the consumer photography domain and that can be integrated in the detection of non-identical duplicate consumer photographs.

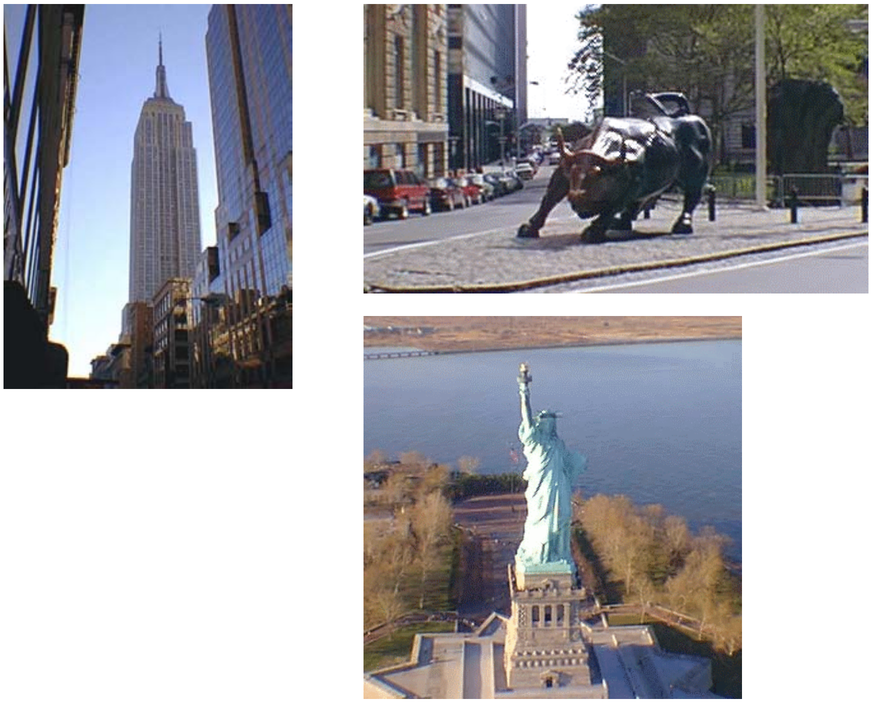


Figure 48. Examples of *Recurrent Visual Semantics* in consumer photography. The objects in these images frequently appear in consumer photographs of New York City.

Another level of repetition is described next.

5.1.3 Why Duplicates are Important

Consumers *very* often make images that are *almost* identical because they photograph the same scene, creating non-identical "duplicates" and similar "non-duplicates" (Figure 49). This repetitive consumer behavior is so prevalent that in Kodak's consumer photography database [168], for example, 19% of the images fall into the category of non-identical duplicate and similar non-duplicate images.

Clearly, consumers will often want to keep only one of those "duplicates" based on their own subjective judgments (e.g., this picture is nicer, she looks prettier in this picture, this one captured the right moment, etc.). Selection of these images, with an interface such as *STELLA*'s (Figure 47), could easily and quickly be done only if these "duplicate" images are clustered independently at the lowest level of the hierarchy.

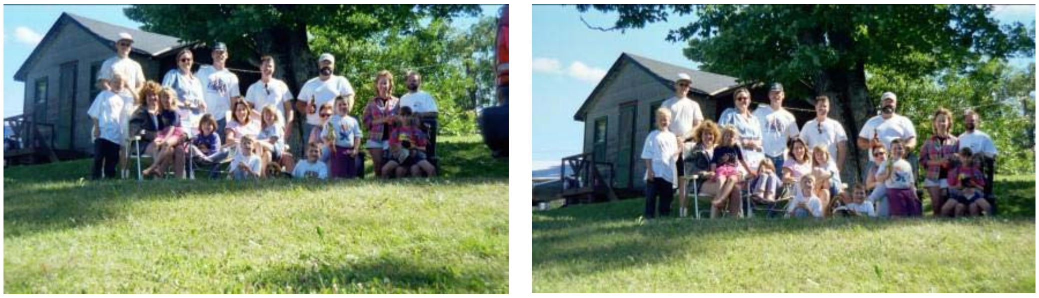


Figure 49. Almost identical images.

5.1.4 Metadata Does Not Solve the Problem

Most digital devices automatically save metadata with the digital images when they are created. This includes the time stamp and often other information such as *Global Positioning System (GPS)* location.

This information could be very useful in detecting duplicate images because it could eliminate from consideration image pairs that, for example, were made at very different times. The assumption would be that images taken at approximately the same time are likely to be duplicates, and images taken at sufficiently different times are unlikely to be duplicates. Time and GPS metadata, therefore, could be used to filter out some of the duplicate candidate pairs that are either far apart in time, or were made from different locations. This could potentially improve the performance of duplicate detection algorithms significantly because, as we will show, the most challenging image pairs are very similar non-duplicates. Eliminating these candidates from consideration would reduce the error rate of the algorithm to detect duplicate images using visual content analysis.

Unfortunately, as we will also show, many of the non-duplicates are photographs of the same *scene, object,* or *event,* and therefore are likely to be made at around the same time just like duplicates are. In other words, time and GPS information might eliminate from consideration some obvious non-duplicates (made far apart in time), but it will *not* be able to eliminate many of the non-duplicates that are visually similar (made at approximately the same time). We can safely assume that most of the images taken far apart in time are sufficiently different visually to be less challenging to the problem we are addressing.

In any case, the use of metadata as a pre-processing step should surely be investigated as there is no doubt that it will have a positive impact on the performance of the algorithms to detect non-identical duplicates. This is especially true with newer cameras and future capturing devices that store additional information such as camera parameters, and annotations generated directly by the consumer at the time the photographs are made. We can certainly foresee a time in the future in which such capture devices will save the *exact* camera location (3D coordinates in a world coordinate system) and parameters (e.g., focus, aperture, speed). Having access to this kind of metadata will be extremely useful.

In spite of having access to all of the metadata just described, it is clear that to determine if two images are duplicates, analysis of their visual content *must* still be performed.

Therefore, we focus on developing a framework for detecting non-identical duplicate consumer photographs based *only* on visual content. This, in essence, is the core problem in determining if two images are duplicates of each other. We do discuss, however, how time stamp and GPS information could be easily incorporated in our framework.

5.1.5 Related work

Many approaches have been proposed to organize personal digital image collections [235][164][168][213][295][296]. However, none of them exploit the *sequence* information available when images are scanned directly from film. In the DejaVideo framework [104] access to video segments is based on remembering clues from the currently viewed video segment. This is done by automatically extracting associations between video segments and other multimedia documents. In [168] and [213], the authors use time stamp information to perform the clustering. Time information is related to sequence information, but it provides significantly more useful detail for this task. However, images do not always have this kind of metadata associated with them.

The clustering algorithm we present, therefore, is novel because it uses this information and a new feature to measure similarity based on composition. In addition, none of the previous approaches have addressed the duplicate problem. An exception is the system in [250], which performs duplicate detection. No details of the approach have been published to this date, however.

In [235], the authors present impressive results on wide baseline stereo view matching. Images are spatially clustered so that images of the same scene from similar views are close together. The focus of that work is on *efficiently* performing wide baseline stereo view matching of multiple images. That approach could be used within the alignment stage of our framework (section 5.3.5.1), but the additional stages to analyze the differences between the images would still be required. In [235], the evaluation focuses on computational cost and complexity; no analysis or evaluation is performed on the overall quality of the matching, although the authors' examples in [235] and [236] suggest that the algorithm works well with the particular sets of images used in the experiments.

The authors of [67] construct mosaics from video to cluster scenes into physical settings. This approach could potentially be applied to collections of consumer photographs. For example, a children's party may take place partly indoors and partly outdoors. The approach proposed in [67] could be used on such sets of images to find the two clusters. This could potentially be useful within our framework because clusters always occur in the same physical setting. Detailed analysis of the difference areas, however, would still be required.

Previous approaches that do detect duplicate images do not handle viewpoint and scene changes and are *not* specific to consumer photography [121][88]. In [121], for example, the authors focus on TV commercials with variations due only to media encoding. Duplicate detection in 2-D binary image document databases has also been addressed [106].

In the field of *Computer Vision* two related areas are registration [237][85][73][175], and change detection [170]. Registration aims at finding correspondences between points in two images and change detection aims at identifying differences between two images. In registration a coordinate transformation is sought which maps the points in image I_1 to the points in image I_2. Registration, or search for correspondence is a required step in stereo matching [237][112]. Registration is also necessary in video motion estimation [249] and for estimating coordinate transformations in the construction of mosaics [68][221]. Change detection algorithms are used in medical image analysis [175], target detection, and surveillance, among others. Although many of these approaches are related to the duplicate detection problem, they assume either very specific domain constraints or minor variations between the images. In video motion estimation, for example, the differences between frames in a video sequence are assumed to be small and strong assumptions are made about the type of motion, the amount of motion and the scene. In medical image registration high accuracy is desired and techniques are often developed for specific types of images (e.g., X-ray, PET, etc. [175]). In consumer photography changes between potential duplicates are often significant and unconstrained making this a very challenging problem.

In the area of *Visual Information Retrieval* many approaches have been developed to measure image similarity (e.g., using color histograms, regions from segmentation, etc.). However, since duplicates can be very different and non-duplicates can be very similar, similarity techniques not specifically developed for non-identical duplicate detection are not suitable.

Other applications of duplicate detection include media tracking [121], copyright infringement detection [88], database integrity, security, and filtering, among others.

The framework we present differs from previous work in several aspects: (1) use of multiple-view geometry; (2) integration of generic visual detectors; (3) use of domain-specific knowledge to detect duplicates.

5.1.6 Outline

In section 5.2 we present the *STELLA* system, new composition features, and clustering. In section 5.3 we present a model of the duplicate problem and a classification of different types of duplicates. In section 5.3.5 we present our framework for detecting non-identical duplicate images. In section 5.3.6 we discuss the construction of our duplicate consumer image database and in section 5.4 we present experiments and discuss future work.

5.2 STELLA

Our goal is to provide the user with a preliminary organization of the images in his collection so that he can subjectively organize them according to his interests. In order to achieve this, first, duplicate images are detected. Then, the images in each roll of film are clustered using color and composition features. We will present our novel composition features and clustering algorithm before addressing the duplicate problem.

5.2.1 Color and Composition Features

As described in section 3.2.1.4, *Global Composition* refers to the arrangement, or spatial layout of *basic elements* (i.e. dot, line, etc.). There is no notion of objects and only basic elements or groups of basic elements are considered.

Our novel measure of composition is computed as follows. First, we automatically segment the image based on color and edges using the approach of [274], which was also used in chapter 4. Next, we extract features to represent the overall shape, orientation, location, color, and texture of image components. In particular, we extract the following set of visual features (section 4.3.5), for each of the *n* regions of the image: *d* = {*extent, roundness, aspect ratio, orientation, location,* *dominant color, minmax difference*}. We then compute a weighted average *Region Feature Vector* (*RFV*) for each of the features extracted, as follows:

$$RFV = \frac{\sum_{i=0}^{n}(f_i * a_i)}{n}$$

Where *n* denotes the number of regions in each image and *fi,* and *ai* are the feature value (i.e., for a feature from set *d* above), and size of the area of region *i,* respectively. Region sizes are used to weight the features because larger regions have a stronger impact on composition. To perform clustering, we create a feature vector that concatenates the weighted averages for each of the features in *d*, with a "number of regions" feature, and a color histogram *ch*. The color histogram is computed converting the image to the LUV color space and quantizing it into 166 levels as described in [242]. The number of regions is important because it may vary considerably for images with distinct compositions, and the histogram is useful in representing the image's color distribution.

For each image, therefore, we extract the following *Composition Feature Vector* (*CFV*):

$$CFV = (RFV, n, ch)$$

The composition feature vectors are then used to cluster the images. Images with similar and distinct composition feature vectors are depicted in Figure 50.

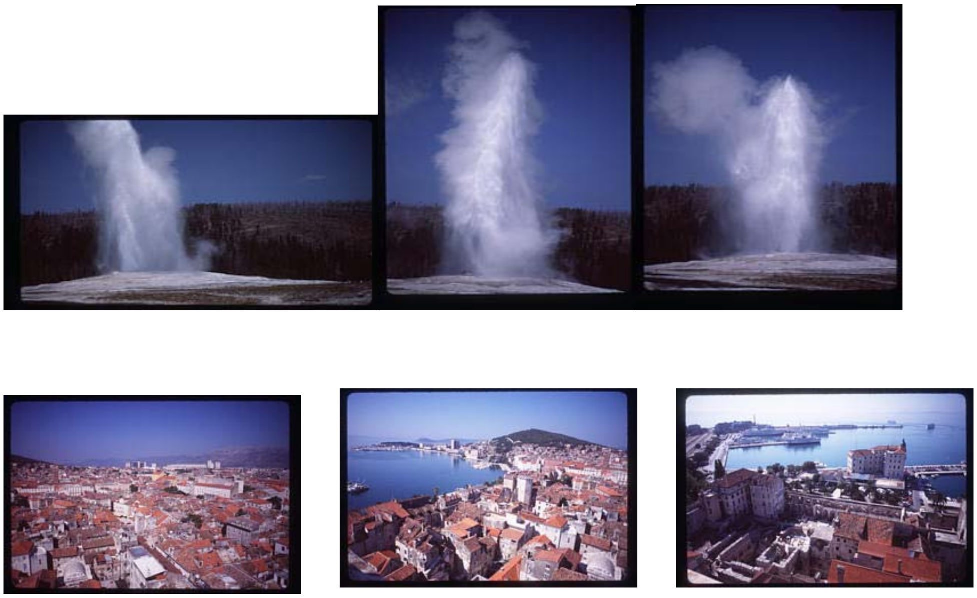


Figure 50. Images that have similar and dissimilar composition feature vectors.

5.2.2 Sequence-weighted Clustering

The goal of *STELLA* is to serve as a tool that helps users organize their personal image collections. Therefore, the images are first clustered automatically and then presented to the user who may subjectively manipulate the clusters according to his interests.

Photographs from a single lens camera are always created in a sequence and therefore this sequential ordering information is always available in film because the frames are numbered. In digital cameras the number of the frame is also sequential and time information is often available.

Since photographs are created sequentially, it is natural to assume that *some* photographs that are close to each other in the sequence are related: proximity can imply similarity because often the photographs might be of the same *scene*, *object*, or *event*. We exploit this sequential structure with a novel algorithm to perform sequence-weighted hierarchical clustering on the photographs in each roll of film. Our approach is general in the sense that it does not rely on time stamp information. Instead, in our clustering algorithm we use only the sequence number of each picture. However, the algorithm could be easily modified to use time instead of just sequence information.

Although there are many clustering algorithms they are often placed in one of two categories: (1) hierarchical; (2) non-hierarchical. Hierarchical clustering algorithms partition the data into a multiple-level hierarchy, producing a dendogram, while non-hierarchical algorithms partition the data at only one level. Hierarchical clustering algorithms are further subdivided into agglomerative and divisive. Agglomerative algorithms start with all elements, where each element is considered a cluster, and merge them iteratively until the entire set becomes a single cluster. Divisive algorithms start with the full set of elements and divide it iteratively until all elements are single clusters.

For our application, hierarchical clustering is advantageous over non-hierarchical clustering because it can be very useful for visualization and for manipulating different groups that contain the same image. Based on the advantages of hierarchical clustering we propose a modified version of Ward's hierarchical clustering algorithm [150]. Even though we emphasize the difficulties of selecting an algorithm for a general data set, several comparative studies have shown that Ward's method outperforms other hierarchical clustering methods [150].

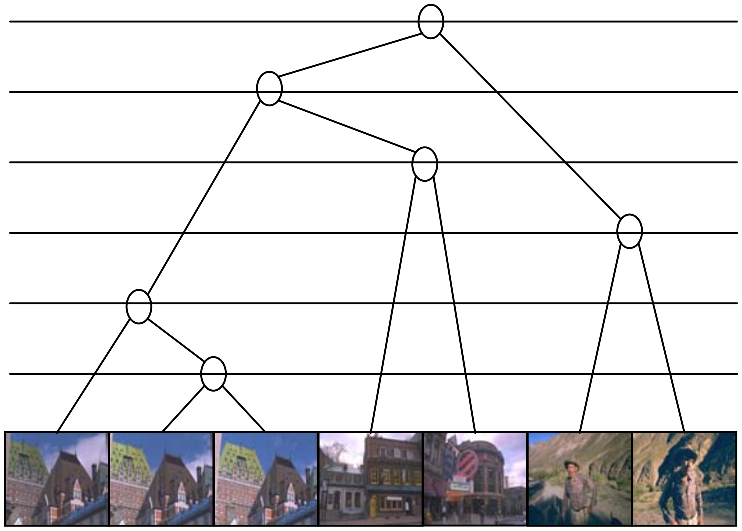


Figure 51. Hierarchical clustering in Ward's algorithm.

In Ward's algorithm, the within-cluster squared error is used to merge clusters. The algorithm starts with a set of *n* data points, each of which is considered a cluster with one element and is represented by its within-cluster squared error. At every step of the algorithm the two clusters with the minimum squared error are merged to form a new cluster so that if the data set contains *n* points, the resulting hierarchy will have *n* levels. This is depicted in Figure 51. Note that in such hierarchy the images that are most similar (e.g., duplicates) should be in the same cluster and should be merged at the lowest levels of the hierarchy. As we discussed in section 5.1.1, detecting duplicates requires a specialized algorithm because many non-duplicates are very similar. It is possible, therefore, to use the clustering algorithm proposed here to cluster all of the images, and to use the duplicate detection algorithm (section 5.3) to place duplicates in separate clusters.

The quality of a clustering algorithm is highly dependent on the data being clustered, and in particular of the distribution of the data (i.e., desired cluster shapes), so in general it is not possible to select a priori the best clustering algorithm without any knowledge of the data. The selection can be made, however, based on general algorithm characteristics, or computational requirements, among others.

We have modified Ward's algorithm to be sequence-weighted so that it accounts for the location of each image within the sequence (i.e., roll of film). In Ward's algorithm the squared error for a cluster is computed as the sum of squared distances to the centroid for all elements in the cluster. In our algorithm we penalize images that are far apart in the film using two additional variables in the calculation of the cluster's squared error. Consider the sequence of images depicted in Figure 52. Between two images (or two clusters), the number of times a skip occurs is *nskip*, and *tskip*, is the total number of frames skipped. In our algorithm the calculation of the within cluster squared error is modified to include the *nskip* and *tskip* variables as follows:

$$sqerr += ((sqerr + tskip * penalty_1) + (nskip * penalty_2))$$

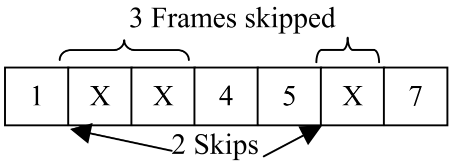


Figure 52. Number of times a skip occurs and total number of skips in a cluster.

These *nskip* and *tskip* values are used to modify the cluster's squared error, so the resulting clusters tend to group images that are closer in the film sequence. This approach differs from the clustering method presented in [272] in which images outside a pre-determined time window are not allowed to belong to the same cluster. The "hard" window criterion used in [272], however, could be replaced with a soft one (e.g., using an exponential decay weight function to define the window boundaries). It also differs from the work in [213], in which an image is added to a cluster only if it is contiguous in the film or sequence to an image already in the cluster.

The approach we propose has strong benefits over the previous ones because it allows *very similar* images to be in the same cluster (even if they are very far apart in the film), at the same time that it discourages non-contiguous images in the same cluster. Figure 53 shows an example of clustering results using our approach in comparison with Ward's algorithm (not sequence-weighted).

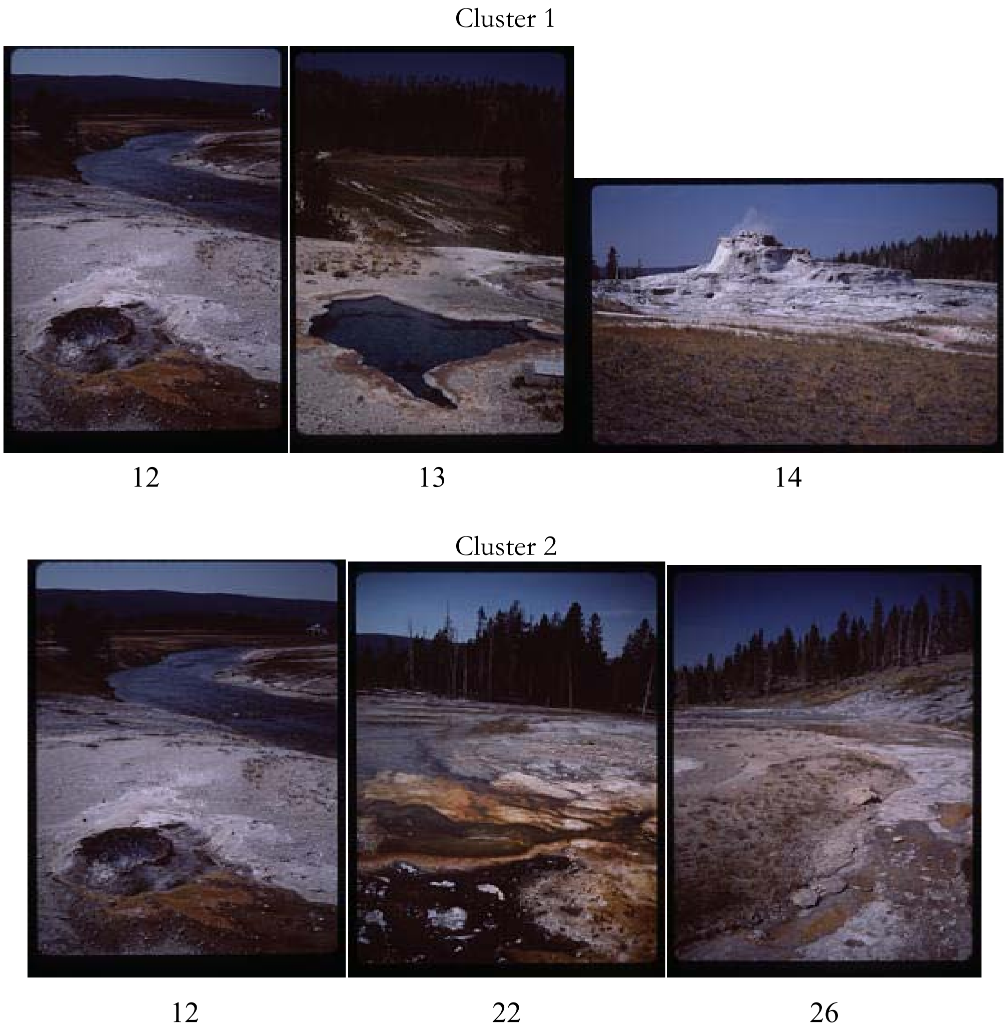


Figure 53. Comparison of original Ward clustering (cluster 1) and sequence-weighted clustering (cluster 2). The number below each image indicates its location in the roll of film.

5.2.3 Cluster Analysis

We analyzed the clustering results of our algorithm qualitatively by applying it to a set of approximately 1,700 professional and amateur travel photographs (over 45 rolls of film) and visually examining the results. The collection included images from the *MPEG-7* content collection [193], photographs by Philip Greenspun [210], and by the author (made independently of this project). Several parameter values (for *penalty1* and *penalty2*) were used in the analysis— we found that different values worked well for different rolls of film.

5.2.3.1 Cluster Evaluation Strategies

We decided to analyze the clustering results qualitatively after exploring several quantitative options. In particular, we explored three different strategies for evaluating the clusters generated automatically by our system: (1) comparison with a human-constructed clustering ground truth; (2) use of cluster validation techniques; and (3) evaluation through user interaction.

In the *first* strategy, 16 rolls of film (approximately 570 images) of the photographs made by the author, from the collection described above, were randomly chosen and images in each of the rolls of film (of approximately 36 photographs each) were independently clustered by the author and his colleague Ana B. Benitez. As a result of this process, we obtained a ground truth similar to the one obtained by a single person in [213] (see Figure 54 for some example images). No similarity or clustering guidelines were set: the two participants clustered each roll of film independently and subjectively.

Although we did not quantitatively compare the resulting clusters, we found little agreement between the two participants. The clusters themselves were quite distinct and the number of clusters generated by each of the two participants for the same rolls differed substantially. For a set of 8 rolls, for example, the average number of clusters for one participant was 19 and the average for the other one for the same images was 10. After comparing the clusters and discussing the criteria that each participant had used, we agreed that for the collection of images we were using, it was very difficult to decide how to cluster the images. In other words, the individual criteria were not well defined. There is no single correct way of clustering the images in Figure 54, for example.

It is important to note that one of the participants was the author of the photographs. This had an important implication, which motivated the final implementation of our system: the author of the photographs has knowledge about the events and places depicted in the photographs. Therefore, clusters formed by the author of the photographs are based on semantic information that may not even be depicted in the images (i.e., what the images are *about*; which corresponds to the abstract levels of Figure 16). Based on this implication, we conclude that unless our algorithm is able to at least cluster the images at the generic semantic levels of the *Multi-level Indexing Pyramid* of chapter 3, it does not make sense to evaluate the algorithm against a ground truth, for this particular set of images. Creating the ground truth here, was found to be very difficult in part for the following reasons, which prevent us from using this strategy: (1) subjectivity in creating the clusters, (2) disagreement between the two participants, (3) use of semantic level information in the creation of the clusters.

As discussed in [179], clustering of images is a subjective task, in which often there is little agreement between users (in the experiments reported there, 34.6% higher than agreement expected by chance). In some cases [168], however, creating a ground truth is possible and very useful, particularly if the goal is to evaluate fully automatic techniques. Other authors [75] have used cluster evaluation approaches from a ground truth given by the collection's textual annotations.

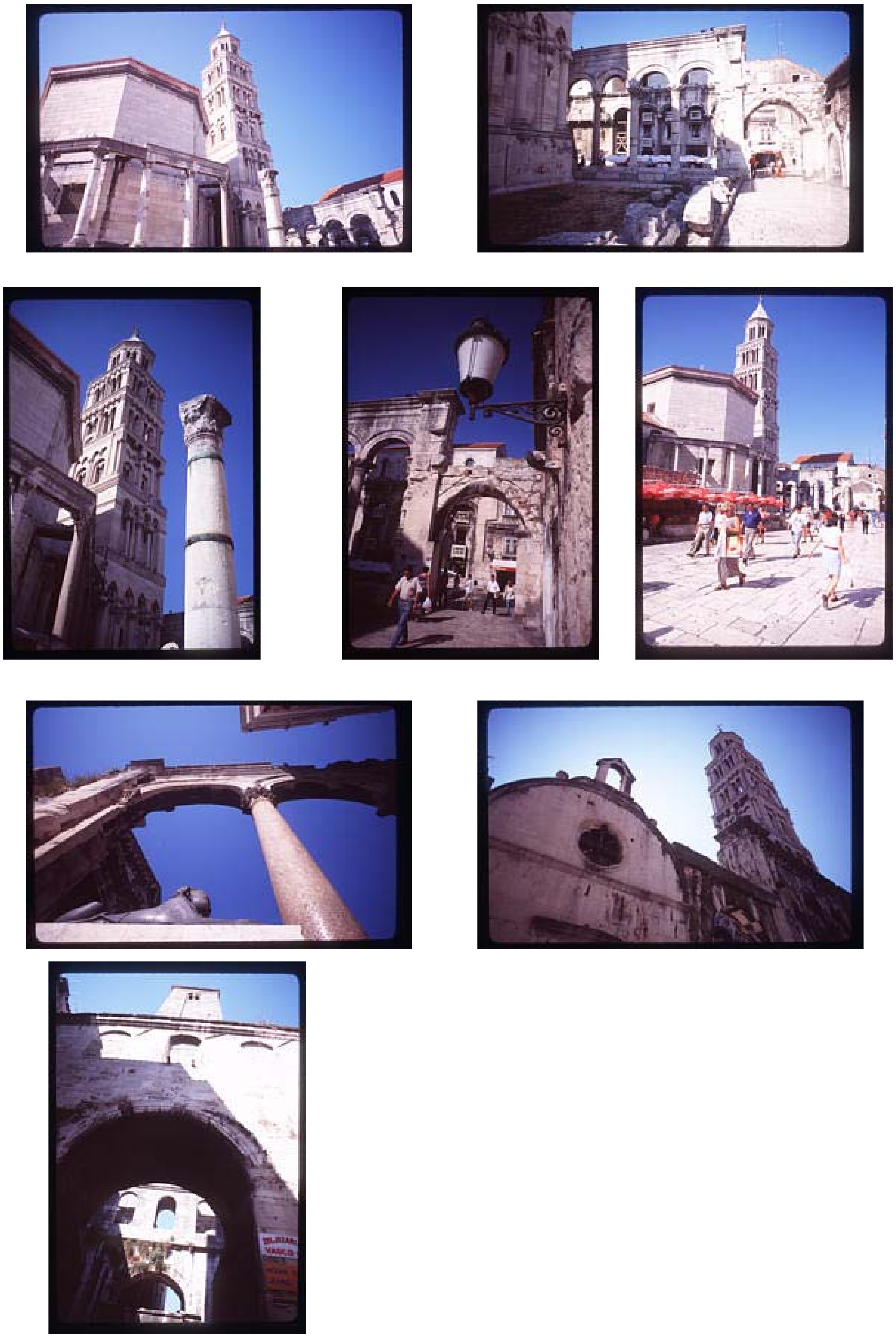


Figure 54. Example images used in the clustering experiment. It was not possible to find consistent clusters.

The *second* strategy to evaluate our algorithm was to use cluster validation techniques [150]. However, external criteria (i.e., does the cluster hierarchy match an expected hierarchy?) are difficult to implement, particularly for hierarchical clusters, because expected hierarchies are not usually available, as is the case here. Internal (i.e., does the hierarchy fit the data well?), and relative criteria (i.e., which of two hierarchical clusterings fits the data better?), on the other hand, usually require a baseline distribution. Such distribution is difficult to obtain for general consumer photographs. Therefore, it is difficult to apply cluster validation techniques with these types of photographs.

The *third* strategy, evaluation through user interaction consists of providing functionality in *STELLA* to track the changes made by a user as he modifies the clusters that are automatically generated. Recall that in *STELLA* the user inputs one or more rolls of film, and the system creates a cluster hierarchy based on the techniques just presented. A considerable advantage of using hierarchical methods is that the results can be easily used for browsing. As the user browses an hierarchy, he is able to perform several operations to *organize* and *select* images. The user can move and modify existing clusters, eliminate clusters, or select satisfactory clusters at different levels (see Figure 55).

The user can also add metadata to individual images or clusters at any level, effectively generating the full range of semantic attributes of the *multi-level indexing pyramid* and *semantic information table* of chapter 3. The environment provided in *STELLA* is flexible enough to facilitate such operations.

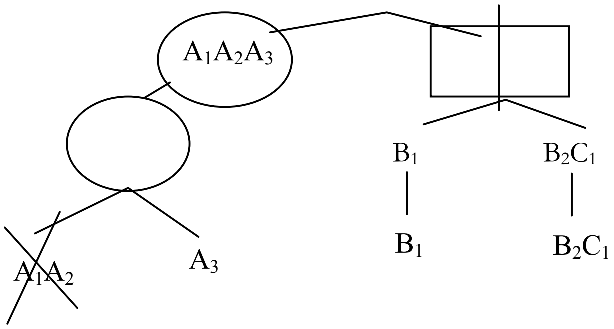


Figure 55. User operations on the output produced by *STELLA*. "Good" clusters are circled, "bad" clusters are crossed out, and others edited (square).

The success of the system can be measured in terms of the user's interaction (not the clustering accuracy). For example, the user performs some of the operations described above and weights are assigned to each operation (e.g., corrections to existing clusters), using measures that are task, content, and user-specific. Although this is an interesting direction, we did not perform any user studies with this approach. The functionality was implemented in STELLA, but a meaningful study here would have required input from several subjects and careful design and evaluation of the graphical user interface, among others. We discuss other possibilities in section 5.5.

5.2.3.2 Analysis of the Results

Given the difficulties in cluster evaluation just described, we decided *not* to explicitly measure the performance of our clustering algorithm. Instead, we performed several clustering experiments, using different features and qualitatively analyzed the results at different levels of the clustering hierarchies.

Our analysis suggests the following: (1) the combination of histogram and composition features provides better results than either one of those features alone; (2) sequence-weighted clustering produces better results than it's non-weighted equivalent.

Even though we did not evaluate our algorithms quantitatively, we were able to identify some of the open issues that prevented this evaluation and suggested possible directions to carry it out.

In the next section we present our work on non-identical duplicate detection.

5.3 NON-IDENTICAL DUPLICATES

In this part of the chapter, we introduce the problem of non-identical duplicate consumer image detection, present a novel framework to address it, and present a new comprehensive study of non-identical duplicates using a consumer image database. We conclude with experiments to evaluate our framework.

In consumer photography it is common to have one or more photographs of the same scene. In Kodak's consumer image database [168], on average, 19% of the images, per roll, are perceived to be either "duplicates" or similar "non-duplicates," making this an important problem for the development of systems that help users organize their photographs [164][213][295][296], and other consumer imaging applications [240].

The majority of the "duplicate" images, however, are *not* identical, but *look* identical or almost identical to the human eye. Figure 56 shows two pairs of images used in our experiments in which we asked 10 people to label image pairs as "duplicates" or "non-duplicates". In the pair labeled "duplicate", there are differences in viewpoint, subject, and exposure. The "non-duplicate" pair contains images of two different people. As this example shows, differences between duplicates can be visually significant, and similarities between non-duplicates can be very high. In fact, often high-level semantic information is used to decide if two images are duplicates or not.

Although much work has been done in several related areas in *Computer Vision*, *Image Processing*, and *Visual Information Retrieval*, the challenging problem of non-identical duplicate consumer image detection has not been previously addressed[31].

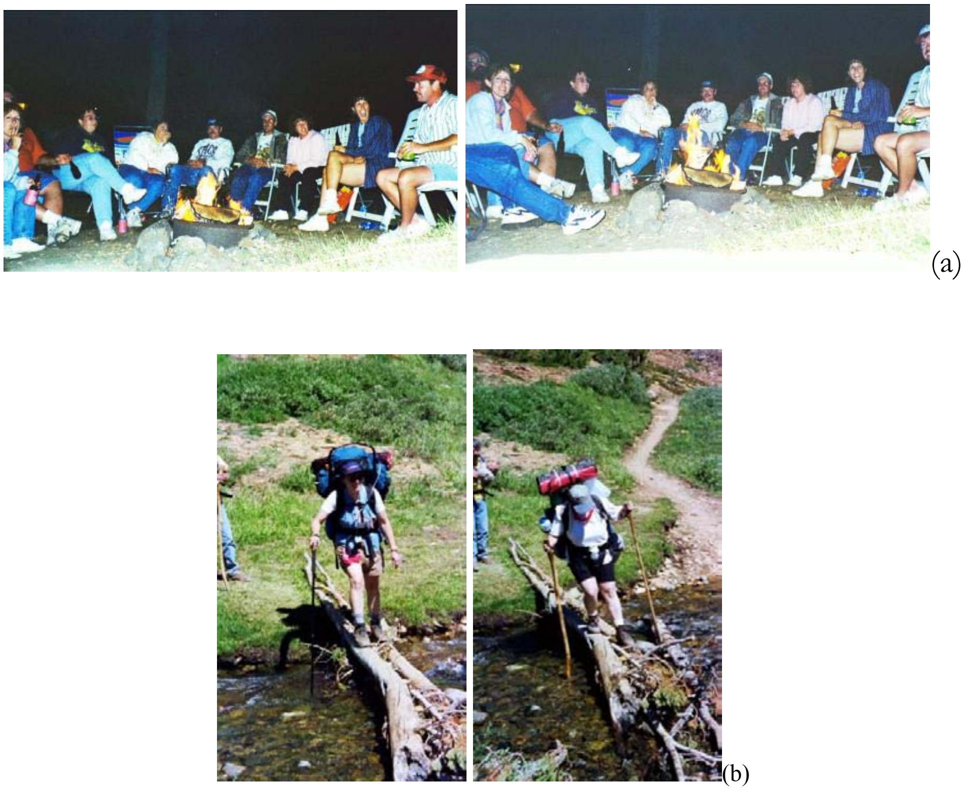


Figure 56. The pair in (a) was labeled "duplicate" eight out of ten people. The pair in (b) was labeled "non-duplicate" by ten out of ten people.

5.3.1 Do we need special algorithms?

As we show throughout this chapter, image pairs that are labeled "duplicates" are often visually dissimilar, and image pairs that are labeled "non-duplicates" are

[31] The system in [250] has a duplicate detection function. Details of the approach, however, have not been published to this date.

often very similar. Therefore small differences must be carefully analyzed and consequently traditional, non-specialized approaches to measure similarity (e.g., based on histograms or simple block-matching) are unsuitable.

Consider the color segmentations in Figure 57 (using the approach of chapter 4 from [274]), and the luminance histograms in Figure 58. Each pair corresponds to the duplicate and non-duplicate pairs of Figure 56. Differences between non-duplicate images cannot be distinguished in the global histograms and the results of segmentation vary widely between two images of the same scene. Even for minor lighting variations or changes in the scene, the segmentation results can vary significantly.

5.3.2 Can this problem be solved?

Issues of subjectivity and the need for high level semantic information make the detection of non-identical duplicate consumer images a very challenging problem. Although it is not possible to solve the problem for the *entire* range of duplicate images, we argue that for some classes the problem can be solved. We aim, rather than to provide a complete solution, to present a general framework for further research, that can integrate algorithms that address some of important issues.

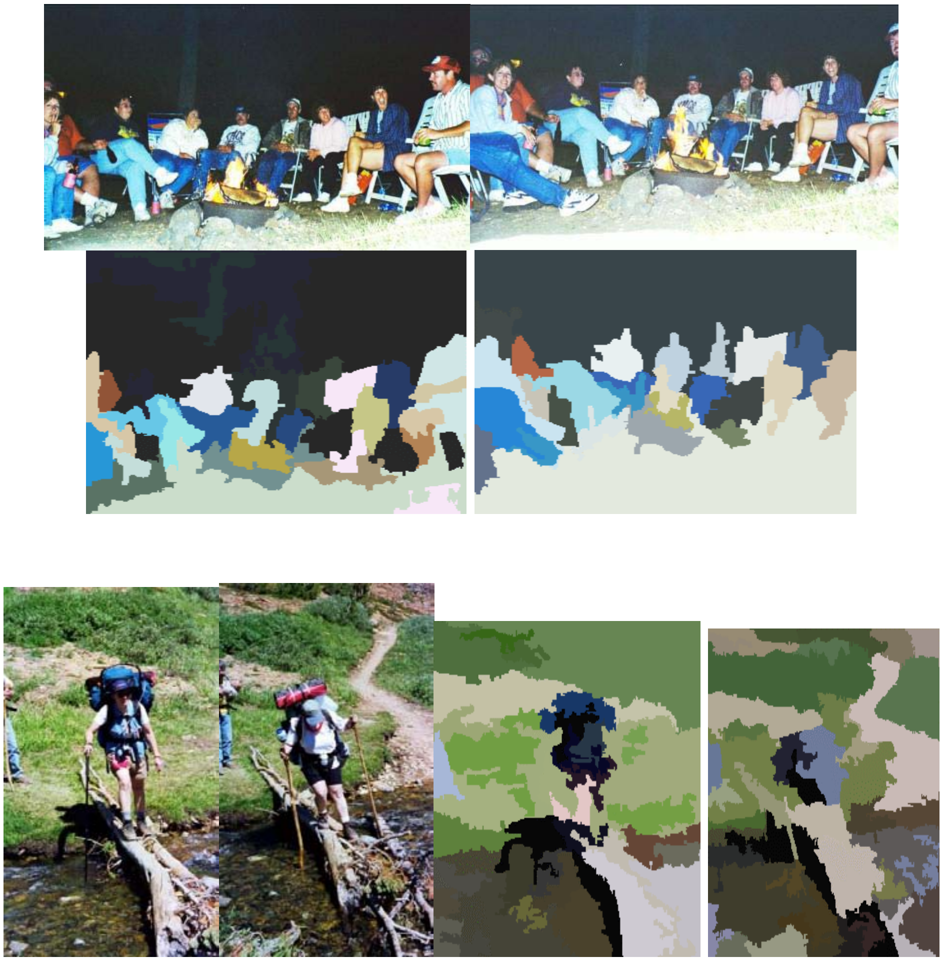


Figure 57. Color segmentation maps for the pair of duplicate (top) and non-duplicate (bottom) images of Figure 56.

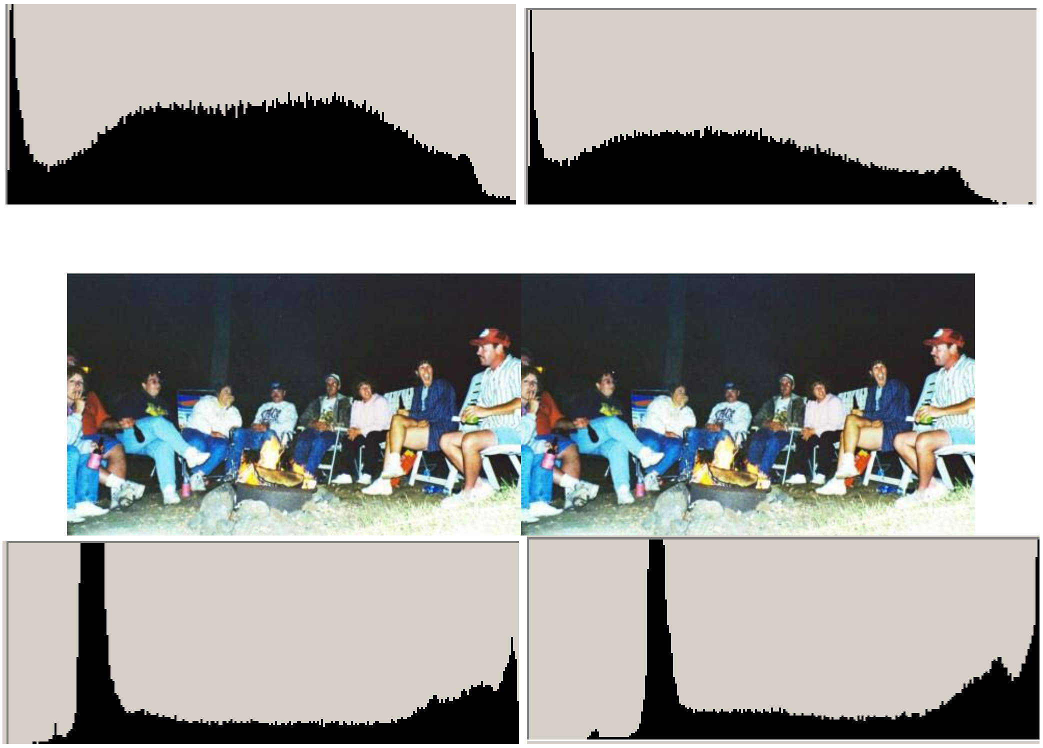


Figure 58. Histograms for a pair of non-duplicate (top) and duplicate images (bottom). These correspond to duplicate and non-duplicate pairs of Figure 56.

5.3.3 Overview

First we present a simple model that accounts for the changes between two photographs of approximately the same scene and present a new classification of different types of duplicates. Then, we present a novel framework for automatically differentiating between non-identical duplicate and very similar non-duplicate images. Finally, we present a new duplicate image database, detailed analysis, and experimental results.

5.3.4 A Duplicate Model

As our experiments show, determining if two images are duplicates is a subjective task. The definition that follows, which was created by researchers at Kodak and used to instruct the participants in the experiments below, however, serves as an initial guideline.

Duplicates: Two photographs are duplicates if they have the same content and composition, and were taken from the same angle and range.

We can expand on Kodak's definition as follows[32]: an image is a duplicate of another if it *looks* the same and does not contain *new* and *important* information. Two images (i_1, i_2), therefore, do not have to be identical (pixel by pixel) to be duplicates. Whether two images are duplicates or not depends entirely on the *differences* between them. These differences can be measured at the levels of the pyramid of chapter 3. The *syntactic* level: *type* (color/B/W), *global distribution* (e.g., histograms), *local elements* (e.g., regions), or *composition* (e.g., arrangement of regions), and the *semantic* level: *generic* (e.g., face), *specific* (e.g., Bill), or *abstract or affective objects* and *scenes* (e.g., sad). Two almost identical portraits of a person might be considered non-duplicates only because in one image the person is smiling (happy), but not in the second one (sad). In this chapter we are concerned

[32] In the construction of the database, the participants were *only* given the Kodak definition, *not* the analysis we present.

primarily with the syntactic level, particularly with respect to local elements, and with a limited set of generic objects (e.g., sky, face, vegetation) at the semantic level.

We can characterize the differences in terms of the *scene* (what is photographed), the *camera* (the capturing device), and the *image*[33] (the digital representation) (Figure 59). The scene may change due to movements of the subject, change of the subject (i.e., a subject is added, removed, or replaced), and the existence of non-stationary elements (clouds, water, etc.).

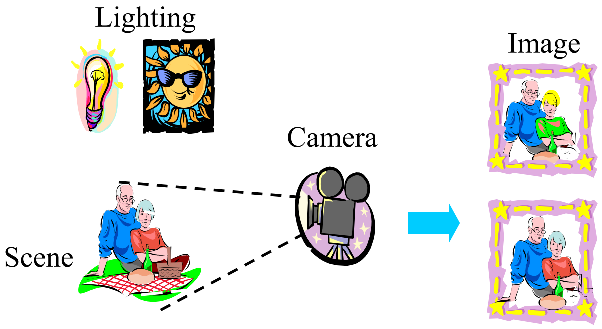


Figure 59. A simple camera model.

[33] The image does not in fact need to be digital. In a camera obscura, for example, the image is the projection of light on a surface.

The Scene

The physical layout of the scene itself may change due to movements of the subject, change of the subject (i.e., a subject is added, removed, or replaced), and the existence of non-stationary elements (e.g., clouds, water, etc.). Subject change or movement often results in occlusion or cropping (e.g., one area is covered and another one is uncovered; a person moves and is partly left out of the image).

Lighting

The scene is affected by *lighting*, and from a specific perspective (the camera's) can often be divided into *subject*, and *background*. Although the subject and background distinction is subjective and not always desired, it can account for important differences between the images. Changes in lighting occur due to quick changes in weather (e.g., sunny or overcast), the use of flash, and locally due to reflections.

The Image

In the simplest case, there are no changes in scene, the location, or the parameters of the capturing device. Differences between two images, therefore, are due to noise or differences in encoding (e.g., work in [121][88]).

The Camera

The photographer can create changes in *internal* and *external* camera parameters. Internal parameters include *exposure*, *focal length*, and *aspect ratio*. External parameters include changes in *viewpoint* (i.e., viewpoint translation) and *viewing direction* (e.g., rotations such as *pan*, *tilt*, and *roll*). Note that a viewpoint change is a change in the location of the optical center of the camera.

Changes in the exposure parameters of the camera are common, mainly because the majority of consumers use point and shoot cameras that automatically change *aperture* and *shutter speed*. Therefore, small changes in the scene can sometimes lead to significant exposure changes, leading to differences in color and luminance.

As our discussions will show, there are three important scenarios to consider in modeling the differences between the *external* parameters of a camera used to photograph the same scene: (1) the camera's viewpoint and viewing direction do not change (no zoom, no rotations); (2) the camera undergoes a rotation (about its optical center); (3) the camera undergoes a translation.

Changes Between Images

We can state the problem of determining if two images are duplicates as a two step process: (1) measuring the differences between two images i_1 and $i_{2,}$ and (2) deciding if the images are duplicates based on those differences. According to our model (Figure 59) those differences depend on three components: (1) *the scene*; (2) *the camera*; (3) *the image*. Table 16 lists specific types of changes within each component.

Table 16. Changes between images

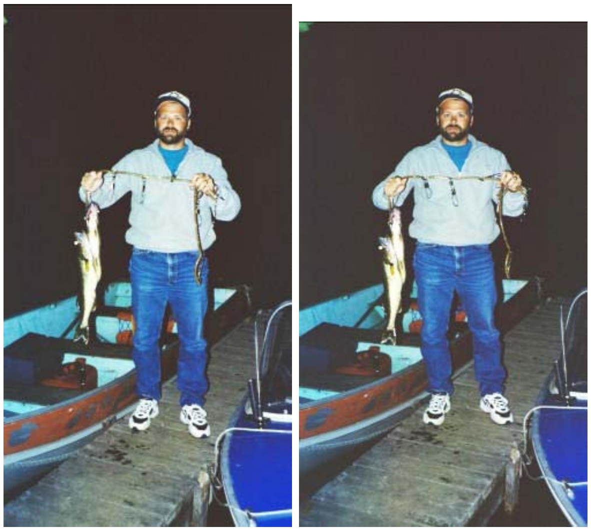
Component		
Scene	Lighting -Flash/no flash/Light/Dark - Sunny/overcast	Subject/Background - Move/Replacement -Non-stationary
Camera	Exposure - Aperture - Shutter speed	Viewpoint - Rotation (pan, tilt, roll) and zoom - Translation
Image	- Noise - Luminosity - Color - Overlayed text (unusual)	

Using this model, it is possible to divide duplicate candidate pairs into several categories, each of which represents the largest visible change between the two candidate images (Table 17): *zoom, angle change, framing, horizontal translation, vertical translation, subject move, subject change, different background, several changes,* and *no change*. A duplicate candidate pair is placed in a given category when the category label's parameter undergoes the largest change. For example, if the biggest difference between two images is the framing, they can be placed in the framing category. Images in the *several changes* category are visually similar, but cannot be easily classified into any other category because they undergo various significant changes. Images in the *no change* category are almost identical. Note that we've focused on the camera and the scene, but it is also possible to include additional categories related to the image itself (e.g., color variations in video). Also note that the distinction between subject and background could be eliminated without significant impact.

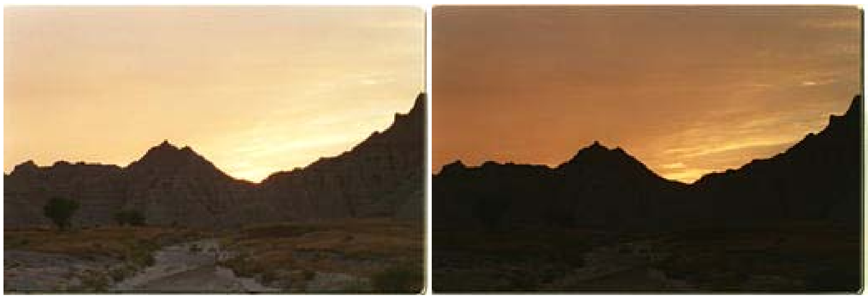
Table 17. Duplicate categories

Category
Angle
Different background
No change
Framing
Horizontal shift
Vertical shift
Subject move
Subject change
Several changes
Zoom

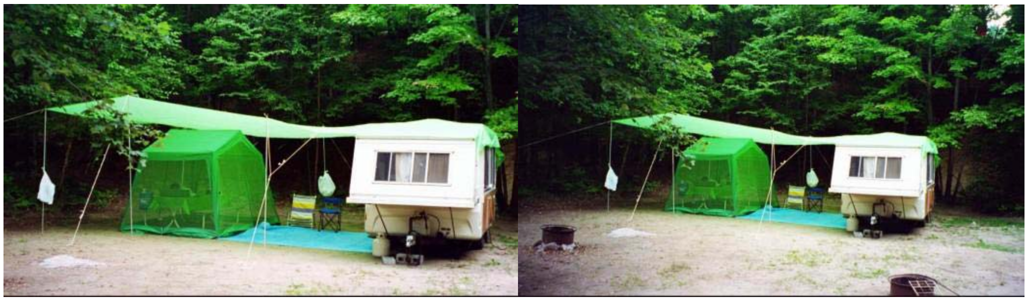
It is possible to strictly characterize duplicates in terms of the parameters of our model. In practice, however, the distinction between different classes may not be that strict and some assumptions can be made. For example, in the first image pair in the examples of Figure 60 there is subject movement (notice the position of the fish relative to the body) and the camera motion in the images in the zoom example is not only a zoom.



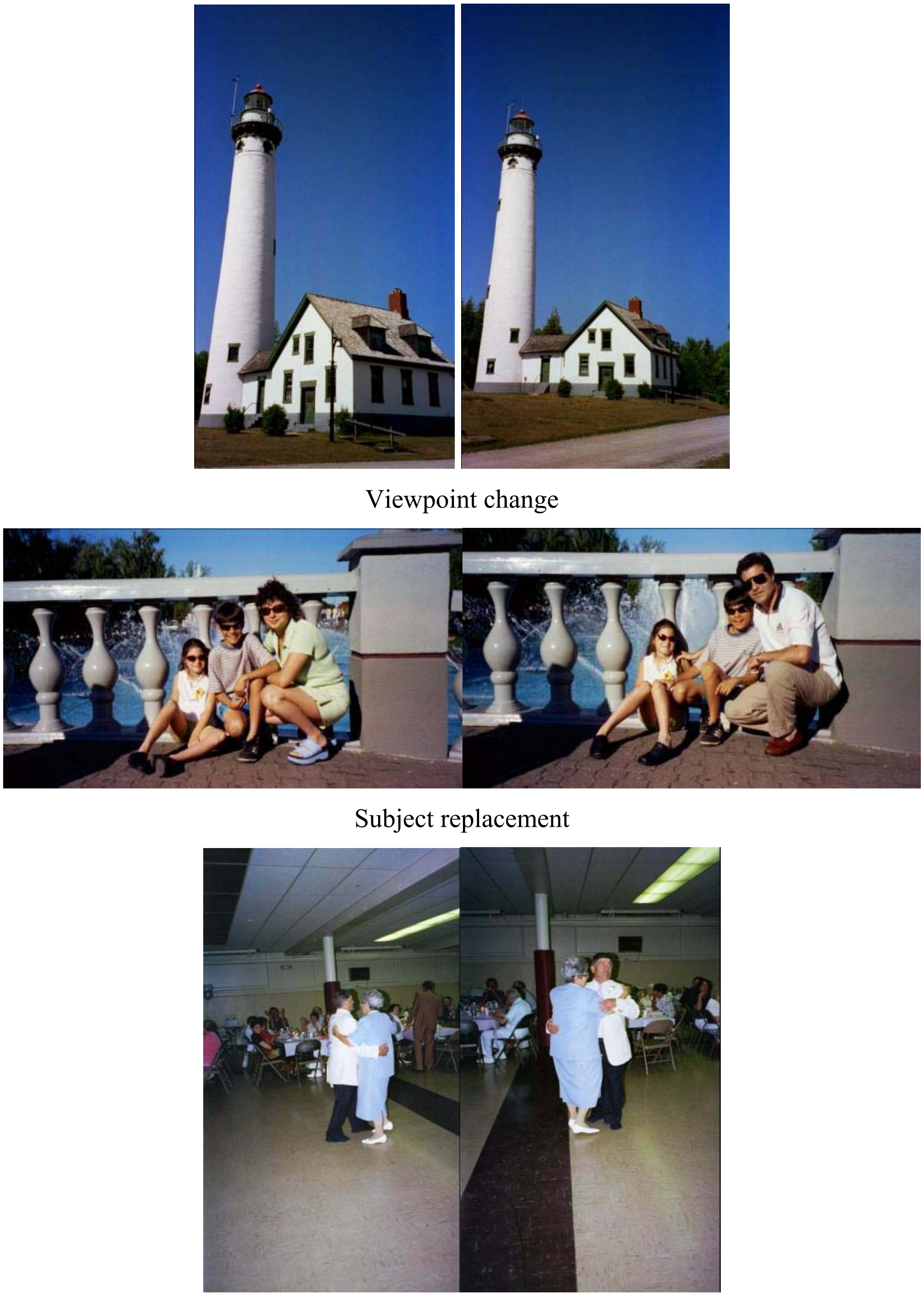
Almost identical



Bracketing



Zoom



Viewpoint change

Subject replacement

Miscellaneous changes

Figure 60. Duplicate examples.

Each component of Table 16, can be viewed as an independent dimension and whether two images are duplicates or not depends on where in that n-dimensional space they reside. For simplicity we show a 2-dimensional space in Figure 61 in which duplicate candidate pairs are represented by circles. Intuitively, those pairs closer to the origin are more likely to be duplicates.

In some cases it is beneficial, from a computational standpoint, to consider independent dimensions. Next we discuss a special case that occurs when only exposure changes and cases in which the camera' changes.

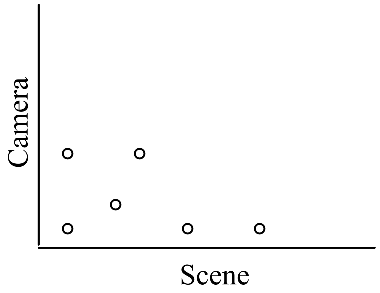


Figure 61. Each component (camera, scene) can be viewed as an independent dimension. Image pairs are represented by circles in this 2D space.

The case in which only the camera's aperture and or shutter speed change(s) is referred to a "bracketing." Bracketing is not common in consumer photography because most consumers use point and shoot cameras that automatically set exposure parameters. Bracketing, however, is very common in amateur and professional photography. In the work presented in [141], for example, bracketing occurred in 20 out of 45 rolls of film (3.25 average number of bracketing cases per roll) in the collection of approximately 1,700 images from amateur and professional photographers (this is the same set used in section 5.2.3). For these duplicates, the element that visibly changes is the exposure (see camera in Table 16).

In terms of a computational approach we know from multiple view geometry that two images of the same scene can be related by a *homography* or by the *Fundamental matrix* (explained below). Therefore, we can examine duplicates from the perspective of projective geometry, and the properties of images of the same scene from multiple viewpoints. As will be evident in the discussion that follows, two cases are important when we consider two images of the same scene:

- Zooms and changes in viewing direction that are only pure rotations about the optical center of the camera. These images are related by a homography.

- Arbitrary viewpoint changes including translation, but no scene changes. The images are related by the Fundamental matrix. If the scene can be assumed to lie on a plane (or lies at infinity), however, these images are related by a homography.

Much of the work in Computer Vision may be used to deal with aspects of some duplicate types (e.g., [235] for view matching related to viewpoint changes). We aim at constructing a general framework to accommodate different types of duplicates.

Multiple-View Geometry

Differences in camera parameters have been studied at length in multiple view geometry and computer vision [112][125]. Therefore, it is useful to utilize existing concepts and techniques in our model of non-identical duplicate images.

One of the important considerations is the geometry that we choose to model these differences, particularly for finding correspondences between points in the scene and points in the image, or between points in multiple images. When we photograph a frontal planar scene, for example, and translate the camera by 1 inch, we can model the difference between the two images using Euclidean geometry. The second image is a translated version of the second one so we can model the difference as a translation transformation.

Different geometries describe different types of transformations and different properties remain invariant to specific transformations. For example, the Euclidean transformation we just described preserves distance, and angles, among others. In projective geometry, on the other hand, distance and angles do not remain invariant: in general, parallel lines in 3D space are not parallel under perspective projection; parallel lines may convert towards a vanishing point depending on the observer's viewpoint. Table 18 from [112] summarizes the particular transformations in each geometry and also the properties that are left invariant by the transformations.

Table 18. Ordering of geometries [112]. Particular transformations and properties left invariant by the transformations. Each geometry is a subset of the next and more general transformations mean weaker invariants.

	euclidean	similarity	affine	projective
Transformations				
Rotation, translation	X	X	X	X
Isotropic scaling		X	X	X
Scaling along axes, shear			X	X
Perspective projections				X
Invariants				
Distance	X			
Angles, ratios of distances	X	X		
Parallelism, center of mass	X	X	X	
Incidence, cross-ratio	X	X	X	X

As we will later show, of particular importance for our purposes are projective transformations. A planar projective transformation is a linear transformation on homogeneous 3-vectors represented by a non-singular 3x3 matrix [124], x'=Hx:

$$\begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

where x' and x are points in 2D.

Projective transformations, which form a projective linear group consist of several specializations, or subgroups. Each of these subgroups has specific invariant properties and degrees of freedom (Table 19).

Table 19. Geometries invariant to common planar transformations [124]. The matrix A=[a_{ij}] is an invertible 2x2 matrix, R=[r_{ij}] is a 2D rotation matrix, and (t_x, t_y) is a 2D translation.

	Euclidean	**Similarity**	**Affine**	**Projective**
Matrix	$\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} sr_{11} & sr_{12} & t_x \\ sr_{21} & sr_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} h_{11} & h_{12} & h_x \\ h_{21} & h_{22} & h_y \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$
Degrees of freedom	3	4	6	8

Now, if we assume a pinhole camera and projective geometry, we can represent the projection from a 3D scene to the 2D image plane as a linear transformation given by a 3x4 camera *projection matrix* **P** such that x≅**PX** so a point **X** in 3D space is projected onto point x in the image. **P** can be written as the product of intrinsic and extrinsic camera matrices **P≅K[R t]** [112]. The intrinsic matrix **K** gives the internal geometry of the camera. If we assume no shear in the axes and no radial and tangential distortions, the matrix has four degrees of freedom (*focal length 1/f, aspect ratio a, principal point P_x, P_y*). The focal length is expressed in pixel units in each dimension and describes the total magnification of the imaging system resulting from both optics and image sampling. Their ratio, which is usually fixed, is called the aspect ratio (*a*). The extrinsic matrix gives the camera's external orientation and position {**R**, **t**}, where **R** is a 3x3 rotation matrix and **t** is a translation 2-vector. The general projection matrix can be written as follows:

$$\mathrm{x} = \begin{bmatrix} a & 0 & -P_x \\ 0 & 1 & -p_y \\ 0 & 0 & {}^{1}\!/_{f} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{X}$$

Which can be written as:

$$\mathbf{x} \cong \mathbf{PX} \text{ and } \mathbf{P} = \mathbf{KP_o}[\mathbf{R} \mid \mathbf{t}] \text{ , } \mathbf{K} = \begin{bmatrix} a & 0 & -P_x \\ 0 & 1 & -p_y \\ 0 & 0 & {}^{1}\!/_{f} \end{bmatrix}$$

Going back to the differences between two images of the same scene, recall that the external camera parameters can change due to a change in *viewing direction* or a change in *viewpoint.*

When changes are due to viewing direction *only* (not viewpoint) we can relate the two views by a *homography*, which is a planar projective transformation which maps points in one image to points in another image. If an image is a rotated version of another image, for example, a homography can be used.

When there is a change in *viewpoint* (e.g., *translation*, *rotation* and *translation*) we can no longer find a transformation that maps points in one image to points in the other image (unless the scene is approximately planar). Instead, we can use *epilolar* *geometry* (described below) to find correspondences between points on one image and lines in the other. This will become clear in the discussion below.

The important issue is that in the duplicate problem we have three scenarios concerning the camera: (1) no change in viewpoint, zoom, or viewing direction; (2) change in viewing direction; (3) change in viewpoint. To handle changes in viewing direction (and zooms) we use a homography, and to handle changes in viewpoint we use the Fundamental matrix. The Fundamental matrix, which is explained below, maps points in one image to lines in the other image.

As we will see in section 5.3.5, we use a homography and the Fundamental matrix to reduce the search for correspondences between two duplicate candidates.

5.3.4.1 Homography and Fundamental Matrix

A homography is a planar projective transformation that can be used to map points from one image to points in another image. It is defined as follows:

$$x' = \mathbf{H}x$$

Where **H** is a non-singular 3x3 matrix and x' is the image of x. From this equation, **H** can be determined uniquely by solving a system of linear equations, as long as there are four correspondences between $[x, y, z]^T$ and $[X, Y, Z]^T$ and no three points are collinear (so that the equations are linearly independent).

A *homography*, then, is a linear transformation of $\mathbf{P}^2$. In other words, we can perform the mapping between two planes using just linear operations and four reference points, without the need to use more complicated representations such as rotations, translations, and camera parameters.

Once we find the homography, we can map the points from image i_1 to image i_2 and vice versa simply by using the equation x' = **H**x, where x is in i_1 and x' is in i_2.

The *Fundamental Matrix* **F**, on the other hand is a 3x3 matrix that defines the *epipolar constraint* which maps points in one image to lines in another image. If p_2, is a point in I_2 corresponding to p_1, a point in I_1, it must lie on the epipolar line $l=\mathbf{F}p_{1.}$.

Consider the 3D point M in Figure 62 and two cameras with optical centers O and O'. Point M will project to point m in the image plane of the left camera. Note, however, that all points along the line OM project to points along the line that connects e' and m' (in the right image). This can be readily observed in the figure and occurs because there is no 3D information about the scene, so the exact location in 3D space, of point M, is not known.

As can be observed in the figure, correspondence between m and m' constraints a point m in the left image to be on the epipolar line of the other image. This is known as the *epipolar constraint*, and the points e and e' are called the *epipoles*. If we are able to find correspondences between several points in the images (e.g., points m and m') we can estimate the Fundamental matrix, which determines the epipolar geometry. Once we have the fundamental matrix, for any point m in one image, we can find its corresponding epilpolar line in the other image. Therefore, the search for correspondence between the point m and the point in the other image that is a projection of point M is constrained to one dimension.

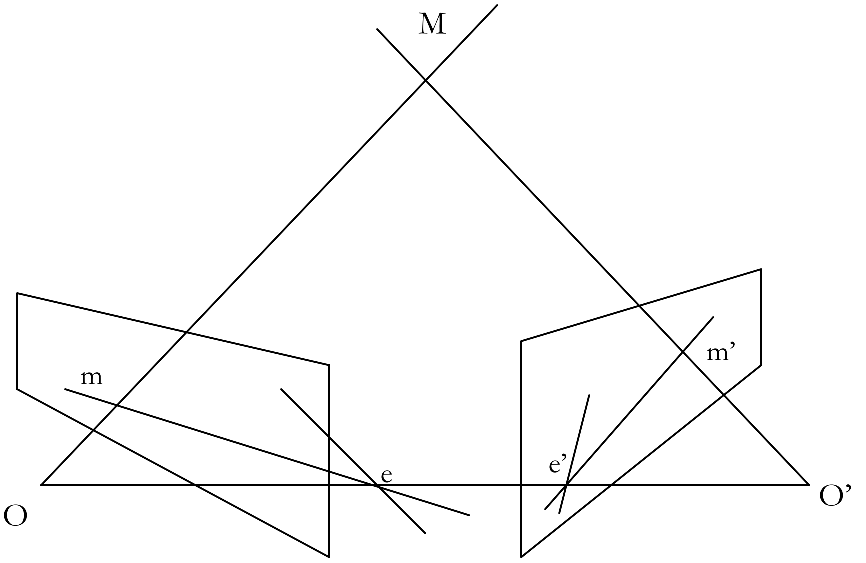


Figure 62. Epipolar Geometry.

Next we present a model based on the three components (*camera, image, scene*) and the knowledge that we can use a homography to map points between planes (e.g., camera rotation) and the Fundamental matrix to find correspondences between points in one image and lines in the other one when there is a change in viewpoint.

5.3.5 A Duplicate Framework

We propose a multiple strategy framework that combines our knowledge about the geometry of multiple views of the same scene, the extraction of low-level features, the detection of a limited number of objects, and domain knowledge about the types of objects that appear in consumer photography and their role in deciding if two images are duplicates (e.g., differences between people's faces and clothing in portraits are more important than differences between vegetation).

As we discussed in section 5.1.4, metadata can play a useful role in detecting duplicate images. Our focus, however, is on visual content, so in our framework metadata can be used as a *pre-processing* step to select the duplicate candidate pairs. The database used in our experiments in section 5.4, for example, contains only images that are very similar. Almost all of the pairs are contiguous in the rolls of film from which they were selected. Even though this selection was manual (the goal was to collect construct a duplicate database), the metadata could easily be incorporated automatically.

The basic steps in our framework (Figure 63) are *alignment*, *change area detection*, and *local analysis*. Alignment is performed using multiple strategies, namely estimation of a homography and the Fundamental matrix, and, if those fail, 2D translation estimation. Detection of change areas is performed using a block-based approach that combines color and edges for matching. Once significant change areas are detected, local analysis of those areas is performed using interest points, object detectors (face, sky, vegetation), and domain specific rules (e.g., image regions below faces correspond to clothing). The final decision is made based on the analysis of the change areas, which is largely influenced by the results of detecting generic objects. We present a novel and extensive image duplicate database (255 image pairs labeled by 10 people). We analyze the subjectivity in the labeling results in detail, discuss the variations in the images in terms of our model, and present experiments based on our framework.

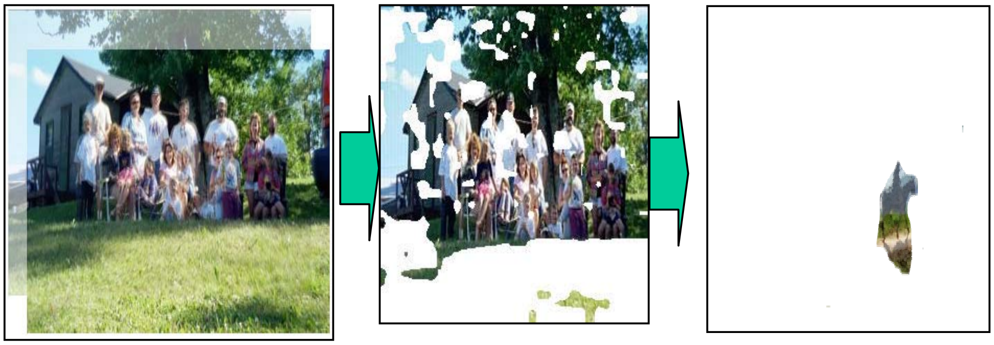


Figure 63. Three basic steps in our framework: (1) alignment, (2) detection of change areas, (3) local analysis of change areas.

Our algorithm is depicted in Figure 64. The input to the algorithm consists of two duplicate candidates I_1 and I_2. First interest points [123] are detected in the two images. Then interest point matches between I_1 and I_2 are found. If the number of matches m is above a threshold t_1, a Homography **H** and the Fundamental matrix **F** relating I_1 and I_2 are computed independently. Otherwise a simple translation is estimated.

A homography will find correspondences between points in the two images if the difference between them is due to a zoom or a change in the viewing direction of the camera (rotation of the camera about its optical center or an arbitrary viewpoint change when the scene is approximately planar). The homography can be used to align the images by applying an inverse global transformation (e.g., the images in Figure 63). Once the images are aligned we can detect the change areas. In the ideal case, a simple subtraction of the aligned images would yield the difference areas. But perfect alignment is usually not possible so we use a block-based correlation approach to measure the quality of the alignment and to find the areas that differ the most between the two *aligned* images.

If the difference in viewpoint is due to a translation, we can no longer find point-to-point correspondences between the two images (unless the scene is approximately planar). In this case, computing the fundamental matrix **F** will yield correspondences between points in one image and lines in the other. In other words, a point in the first image must lie along its epipolar line in the other image.

Computing **F** and **H** in our framework is important because they make the matching more efficient and because they are useful in handling cases that the simple block-based matching algorithm cannot handle well. For example, if an image is a 90^{o} rotated version of another one, the search for correspondences using a block-based algorithm would have to consider rotations. This can easily be handled by the homography instead.

The quality of the estimates of **F** and **H** are measured using a block based correlation approach. If the correlation score *s* is below a threshold t_2, the algorithm discards those estimates and performs a simple translation estimate instead. The step after alignment consists of obtaining a mask of significant global change areas (Figure 63). The mask is computed by thresholding the global registration results obtained from block correlation. The local analysis of the change areas then uses the original interest points, a self-saliency map (used to obtain important areas in each image independently), and results of object detection (face, sky, and vegetation). The final duplicate similarity score is assigned using a set of rules based the local analysis of the change areas.

Sometimes the differences between I_1 and I_2 include a 90° rotation of the camera or cropping of the scene (e.g., panoramic views in APS cameras). The corresponding normalization is applied before any processing. We assume that the images are oriented correctly, so the width and height of the images can be used trivially for this task. Of course, this step could also be performed using the homography.

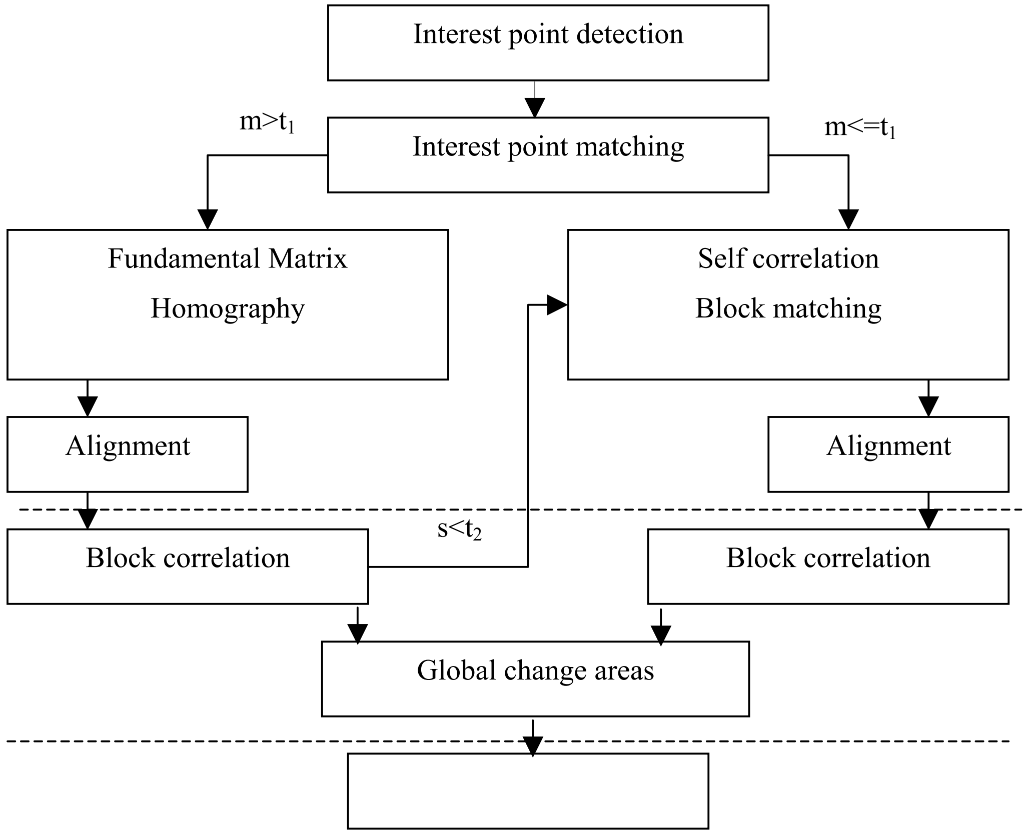


Figure 64. Algorithm overview.

5.3.5.1 Step I: Global Alignment

First interest points are computed in each image using a Harris corner detector [123] from [258]. Then, for each interest point p_i in image I_1 we find its corresponding point p_2 in image I_2 using *Sum of Squared Differences (SSD)* [85][237] also from [258].

As is shown in Figure 64 the next step depends on a threshold of the number of matching interest points. If the number of matches is above a threshold t_1, a homography and the fundamental matrix are computed. Otherwise 2D correlation alignment is performed.

When there are no changes in the scene and the difference in viewpoint or viewing direction is not too large, the number of matches *m* will be above a threshold t_1. The two images might therefore be related by a *Homography* **H** or by the Fundamental matrix **F** [112].

Without any prior knowledge about the camera or the scene, however, it is not possible to know in advance whether **H** or **F** should be estimated (as pointed out in [259], the geometric model is usually selected manually). The more robust alternative, then, is to estimate both. We use the *RANSAC* algorithm [135][258][124][112] to estimate **H** and **F**. Once the Fundamental matrix is computed we can obtain, for any given point in image i_1, its corresponding epipolar line, and for *any* set of epipolar lines, a common intersection point called the epipole (please refer to Figure 62). If the Fundamental matrix is not correctly computed, of course, epipolar lines will not meet at a common epipole. However, having a commong epipole is not sufficient to determine that the Fundamental matrix has been correctly computed.

The location of the epipoles in the images and the orientation of epipolar lines depend on the relative position of the two cameras (see [112] for a detailed analysis). In what is referred to as the "rectified" case, the epipolar lines correspond to the horizontal image scanlines. This occurs only if one camera has translated, in parallel to the image plane.

If the difference between the two photographs is due to a movement of the camera along its optical axis, the epipole will appear inside the image plane. In this case, when the epipoles are *inside* the images, rectification is avoided because mapping the epipole to infinity will cause undesired transformations to the images [116].

If the **F** estimate yields epipoles *outside* the images, image rectification is applied to make the epipolar lines parallel to the horizontal axis.

At this stage we have already computed the fundamental matrix **F** and have found the epipoles **e** and **e'** in the two images. We outline one possible algorithm to compute rectification [124]:

(1) Apply a projective transformation **S'** that maps the epipole **e'** in one image to the point at infinity $(1,0,0)^T$. In particular use **S'=GRT** where **T** is a translation taking an arbitrary point of interest $\mathbf{x_o}$**'** to the origin, **R** is a rotation about the origin taking the epipole **e'** to a point $(f,0,1)^T$ on the x-axis and **G** is the following mapping, which takes $(f,0,1)^T$ to infinity:

$$G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1/f & 0 & 1 \end{bmatrix}$$

(2) Apply a mapping to the second image to match its eipolar lines with the epipolar lines of the first image. In particular, find the matching projective transformation **S** that minimizes the following least-squares distance:

$$\sum_i d(Hx_i, H'x'_i)$$

where **x***ᵢ* and **x'***ᵢ* are matching points that were computed earlier between the two images. This is done by minimizing the following expression:

$$\sum_i (ax_i + bx_i + c - x'_i)^2 + (y_i - y'_i)^2$$

which is equivalent to minimizing the following expression ((y_i-y'_i)^2 is constant):

$$\sum_i (ax_i + by_i + c - x'_i)^2$$

we have

$$S = \begin{bmatrix} a & b & c \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(3) Resample the fist image according to the projective transformation **S** and the second image according to the projective transformation **S'.**

The process just described, then, rectifies the two input images: the epipolar lines in the two images coincide and are parallel to the x-axis. With this result, finding the correspondences between points in the two images requires only searching in the corresponding scan lines (i.e., a point p in image I_1 will lie on somewhere *on* the corresponding scan line in image I_2).

The homograpy **H**, which is estimated using the *RANSAC* algorithm [124] yields a 3x3 matrix that can be applied to align the two images.

Once fundamental matrix **F** and a homography **H** are applied *globally* to the images and they are rectified, the block-based method of section 5.3.5.2 is used to globally match the rectified images. For each pair of transformed images, therefore, we obtain a match score *s*. If *s* is above a threshold t_2 the algorithm detects the change areas as explained in section 5.3.5.2. It is important to note that the parameters of the block-matching algorithm are different for the two cases: when the images are rectified using **F**, the *y* coordinates of corresponding points will be equal, but the *x* coordinates will differ.

Since the two geometric transformations model the camera motion, it would not make much sense to determine different matrices from local areas (e.g., a homography for every block in the image), except in special cases. For example, it would be possible to find *planar* patches in the images and match those patches using a homography estimated from correspondences in the matches. The advantages of this kind of approach in our duplicate problem, however, are not immediately apparent. In general, we are interested in matching the entire scene and then finding the local differences between the two images.

Translation Estimation

When there are changes in the scene, the images are similar but not of the same scene, or the differences in viewpoint are large, the number of interest point matches will be small or the estimate of $\mathbf{H}$ and $\mathbf{F}$ will not align the images well. In either case the block-based correlation approach of section 5.3.5.2 is used to find individual block displacements. Each block is assigned a weight given by its self-saliency score (explained below) and a global coordinate Euclidean transformation (2D translation) is computed using a weighted average of the individual block displacements. The inverse of this transformation is applied to one of the images to "roughly" align I_1 and I_2. The next step consists of matching the images to find areas in which they differ.

5.3.5.2 Step II: Detection of Change Areas

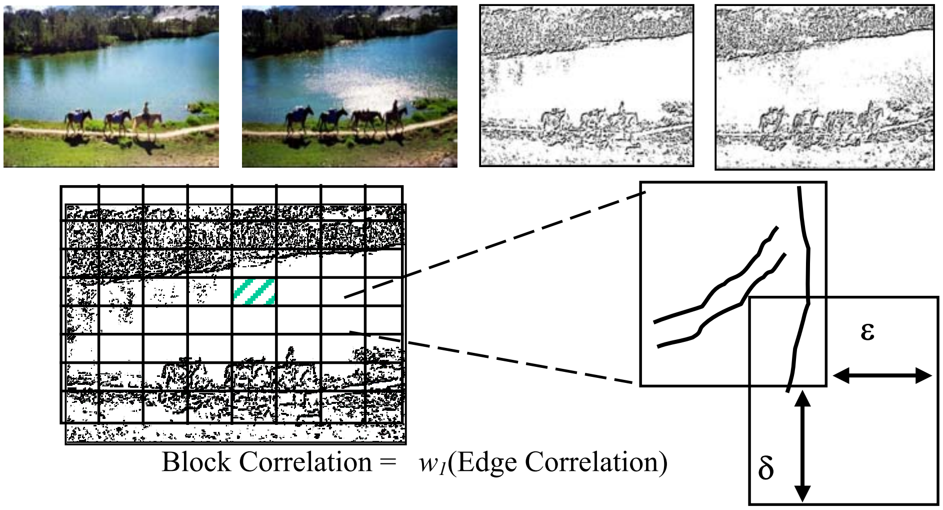
The block-based correlation we present in this section is used for three different purposes (Figure 64): (1) estimation of 2D translation, (2) detection of significant change area areas, (3) self-correlation.

As explained in the previous section, we use the technique we describe here to estimate 2D translations when the computation of a homography and the Fundamental matrix fail, and to detect the change areas once the images have been rectified (either by applying an inverse 2D translation operation, a homography, or using the Fundamental matrix). Before we describe the algorithm we should point out that many similar, more robust techniques for stereo correspondence [237], registration [85], and motion estimation [249] could be used here.

Since we wish to find differences between the images that result from object movements or replacements, it is reasonable to rely on a joint color-edge based correlation approach. We make the following assumptions about the invariance of edges.

- Strong edges (object boundaries/texture) remain approximately constant under lighting variations.
- Changes in exposure are approximately like changes in lighting.
- When subject movement occurs, similar edge patterns appear in different locations.
- Non-stationary subjects are likely to maintain similar edge structure.
- Camera location changes, if not too significant, preserve approximate edge structure.

We combine the approach of [130] with color average in *LUV* color space. The images I_1 and I_2 are divided into mxn blocks and their edge intensity images E_1 and E_2 are obtained using a Sobel operator (Figure 65). For each image pair, each block B_{1i} from the edge map E_1 is shifted over edge map E_2. For each block displacement δ, ε, we count the number of edge and space matches and mismatches between B_{1i} and the corresponding area in E_2. The edge correlation score for each block is the normalized maximum correlation value over all of the block's shifts. For each displacement we also compute the Euclidean distance between the *LUV* color averages of the B_i and the area in I_2. The joint correlation score for each block is a weighted linear combination of the edge correlation score and the *LUV* color correlation score.



Block Correlation = w_1(Edge Correlation)

+ w_2 (Color Correlation) (a)

Correlation = Max correlation for all ε, δ (b)

Self-similarity=Σ correlation scores for all blocks (c)

Figure 65. Block based correlation.

This produces an NxM matrix **Q** of values [0,1], where N and M are the width and height of the image for which the correlation is being performed. Lower values in **Q** represent areas in which changes may have occurred. The average of **Q,** determines the quality of an alignment (Figure 64). If the images are aligned well (e.g., the scene is approximately constant), the correlation score will be high.

It is also used, with a pre-determined threshold t_2 to obtain a *mask* of significant change areas (Figure 66, Figure 63). In these examples the threshold is low, therefore minor changes are selected. Note the minor rotation of the boat in Figure 66.

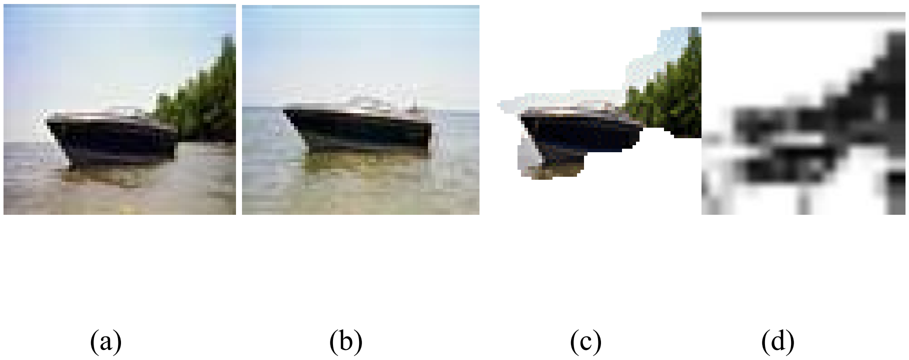
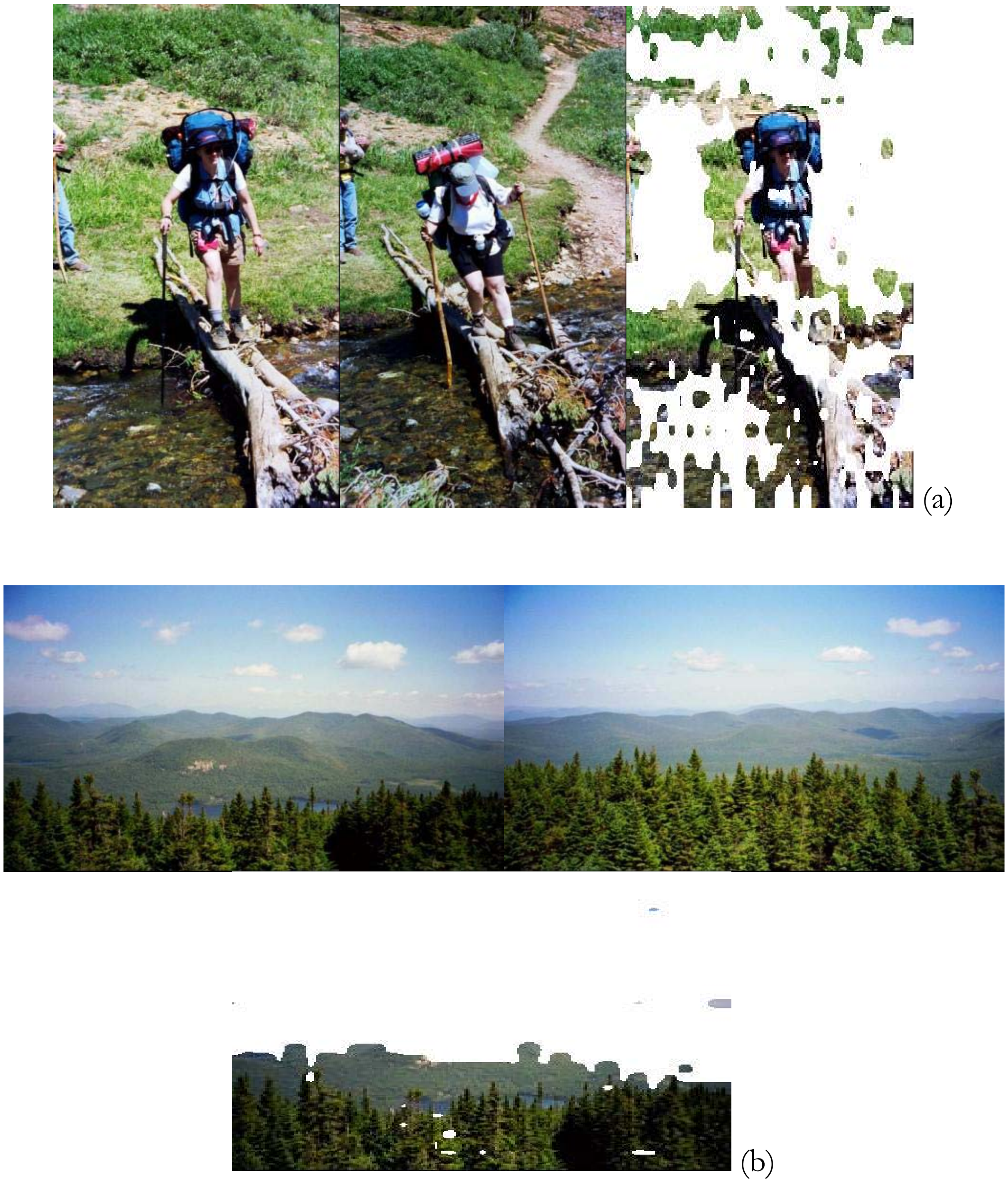


Figure 66. Two images, a change area mask (c), and a self-saliency map (d).

Using a slight modification of the approach to obtain **Q** we compute a *self-saliency map* for each image independently. A self-saliency map **S** is simply an NxM matrix whose entry values indicate how similar the particular pixel is to its neighbors. Areas that are less similar to their neighbors are more salient and therefore deemed to be more important. Similar ideas have been used for registration and tracking [94][100]. We use the block correlation approach to compute self-similarity for each block, but we don't select each block's maximum correlation score over the block's displacements. Instead, for each block, we accumulate all of the block's correlation scores over all of the block's displacements (Figure 65(c)). Note that in smooth areas (like the sky) all neighboring blocks are similar (Figure 66) and the approach is able to select the salient areas. In Figure 67 we show several examples of change areas detected between duplicate candidates. Note that the algorithm is capable of roughly detecting the appropriate change areas. In Figure 67(d), for instance, it is able to detect the difference in the location of the plane. Deciding whether those differences are important, however, is often difficult.



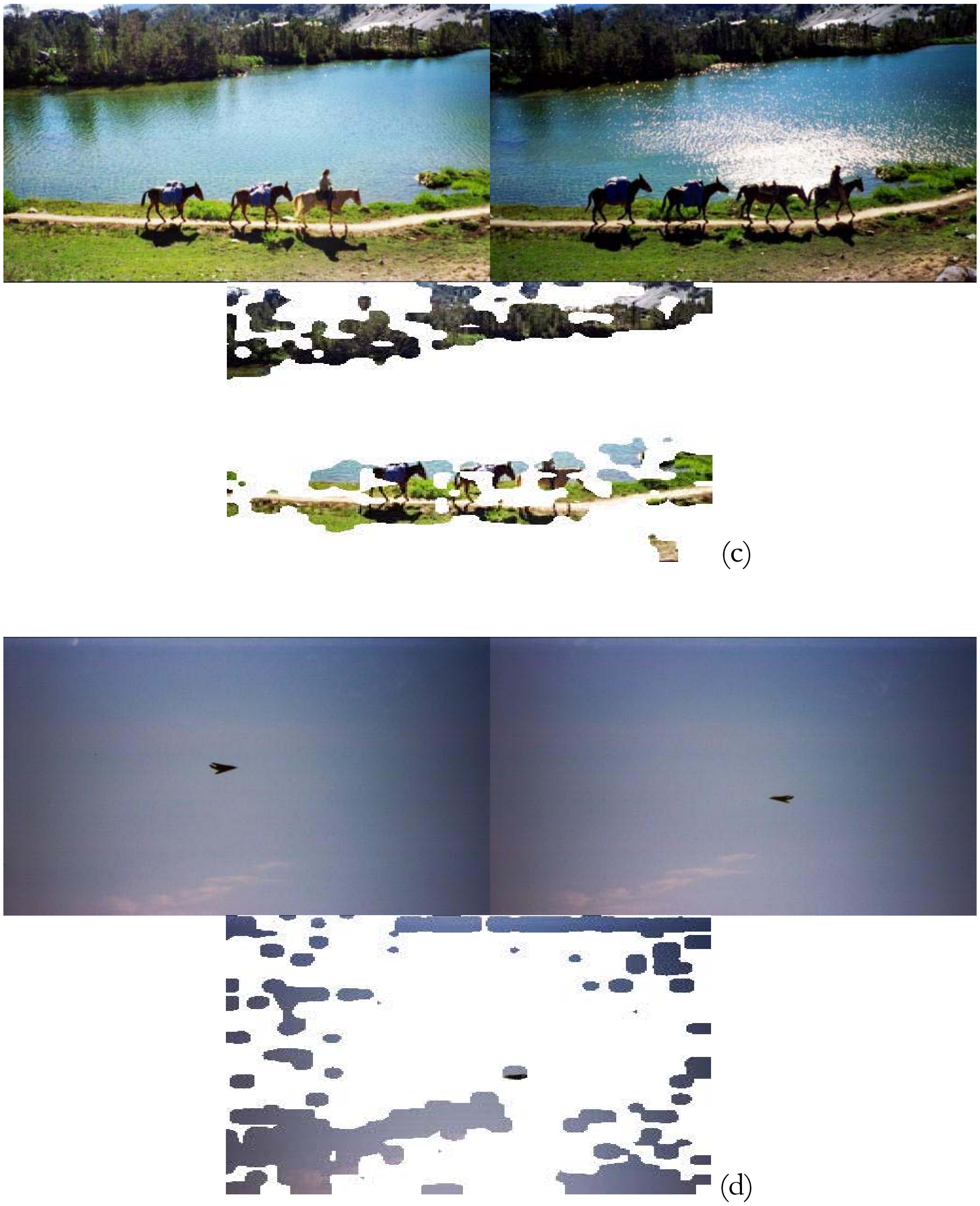


Figure 67. Example of change areas obtained by the block-based correlation algorithm.

5.3.5.3 Step III: Analysis of Change Areas

Recall from Figure 64 that after the global change areas are found (previous stage) we perform a local analysis of the images in the areas *inside* the change area mask computed in the previous section.

The local analysis process consists of two steps: (1) growing of feature point matches, and (2) area matching using visual detectors.

Growing Interest Points

The previous stage (section 5.3.5.2) produces a *binary* change area mask. Since the images have been aligned, the mask can be placed directly over each of the images to select those pixel areas that differ the most between the two images.

At the same time, within the change area mask we *might* have a number of interest point matches p_i and p_j, which were found in section 5.3.5.1.

The goal of this stage is to use the interest point matches *inside* the change area mask to *reduce* the areas that do not match inside it. In other words, we wish to find other areas inside the change area, which match between the two images, and we use the matching points already found to perform the search.

We search the neighborhood around the matches in both images. In particular, if the 2D Euclidean distance between the interest points that match is below a threshold t_3, it is assumed that they come from the same object and the object movement is negligible. Otherwise there are two possibilities: (1) the points correspond to the same object and the object has moved, or (2) the points correspond to a different object. We search the *neighborhood* of the points in each of the images.

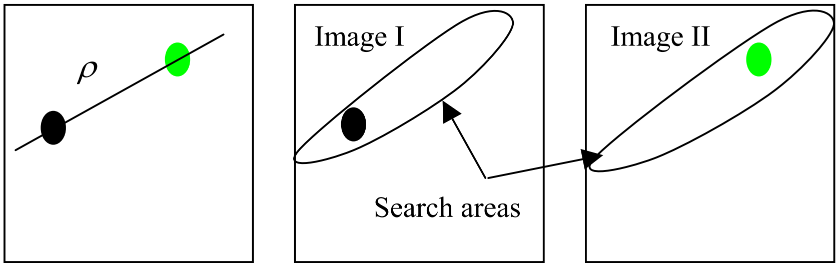


Figure 68. Processing of interest point matches.

Matching around interest point neighborhoods is computed using the block-based correlation method of section 5.3.5.2. Given two matching points, $p_1(x_{11}, y_{12})$ in image i_1 and $p_1(x_{21}, y_{22})$ in image i_2, the goal is to determine if the areas around those points also match.

The blocks are shifted along the direction of the 2D line that connects p_1 and p_2. Since we assume the feature points belong to the same object, there should not be discontinuities in the similarity scores of adjoining blocks in each of the images. Therefore, a search for correspondences is performed along the line connecting p_1 and p_2 and the search is stopped if the similarity value falls below a threshold t. The similarity is measured by both comparing the self-saliency maps of the images and by using the block-based similarity metric described in section 5.3.5.2.

In essence what this process does is attempt to match the *areas* that connect the matching interest points. The interest point for a given object, however, may be at the edge of the object or somewhere inside the object, as depicted in Figure 69. In the figure, the object has moved to the right and the two images, the duplicate candidates, have been overlaid. In both images the interest points detected correspond to the same point in 3D space of the object. In Figure 69(a) the interest point is on the edge of the object, so only the areas to the right of the interest points should match in the two images (the area connecting the two points, as was just described). In Figure 69(b) the interest points are in the middle of the object, so the neighborhoods to the right and left of the images will match. To handle the second case, the search extends to both sides of the corresponding interest points.

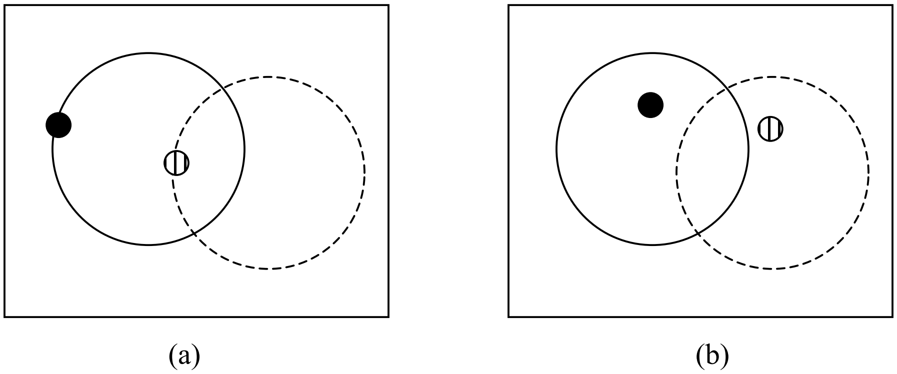


Figure 69. Possible locations of the matched interest points.

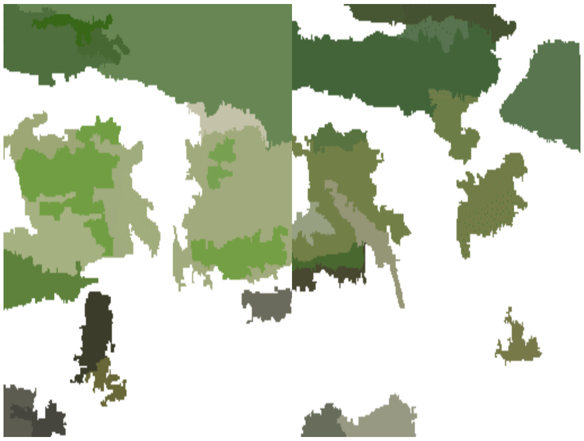
Rules and object detectors

After the previous stage there will still be areas in the two images that do not match well. Here we combine domain knowledge with object detectors to analyze those changes.

Object detectors are learned from a set of training examples using a collection of machine learning algorithms in the *Visual Apprentice* described in chapter 4. Images are automatically segmented and objects are detected in the following categories: *sky, vegetation,* and *face*. A sky object, for example, is detected by grouping several contiguous sky regions obtained from segmentation. Some examples are shown in Figure 70. Note that the algorithm is not always capable of correctly detecting the appropriate objects. Most of the faces are correctly detected in pair (a), as are the vegetation areas in pairs (b) and (c). Incorrect faces are detected in pair (c).

(a)

(b)



Vegetation

Faces

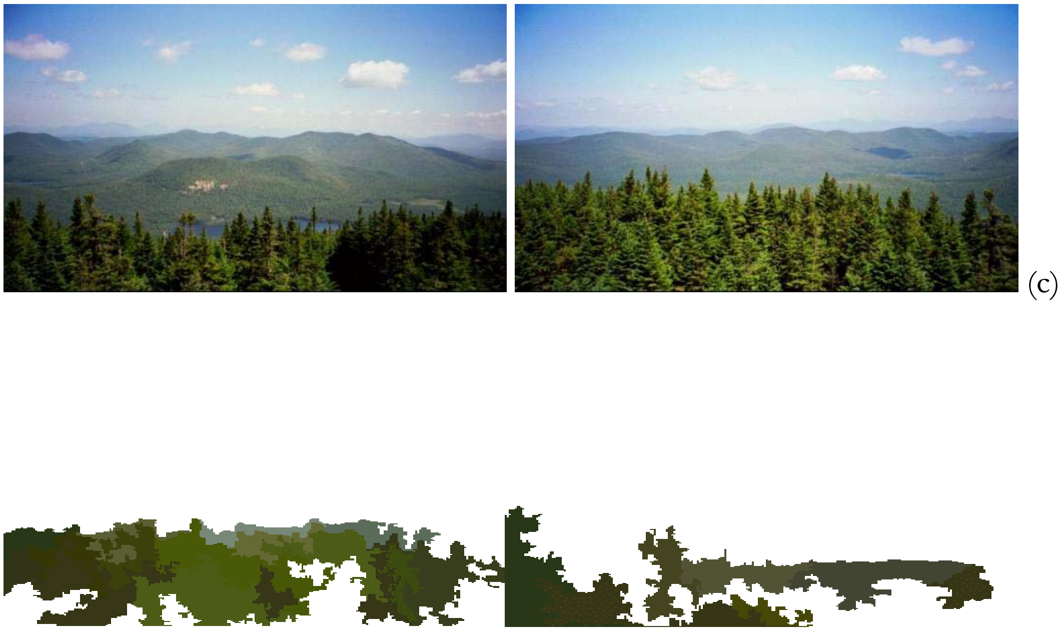


Figure 70. Examples of duplicate candidates and object areas (faces and vegetation) detected automatically.

Manually constructed domain-knowledge *heuristic* rules use the output of the classifiers to determine the significance of the change areas that remain after the previous stage. For example, if faces are found in the change areas, and faces appear in both images, the algorithm tries to match the areas of the two images below the faces. The matching is done again using the algorithm of section 5.3.5.2, but the weights for color and edges are set by the rules. For areas below faces the difference in color is more important, so more weight is given to that feature. A low matching score may indicate that a different person is in the scene. Figure 71 shows additional examples of detected areas. Note that the faces in Figure 71 and in Figure 70(b) are missed so the rules for clothing are not applied.

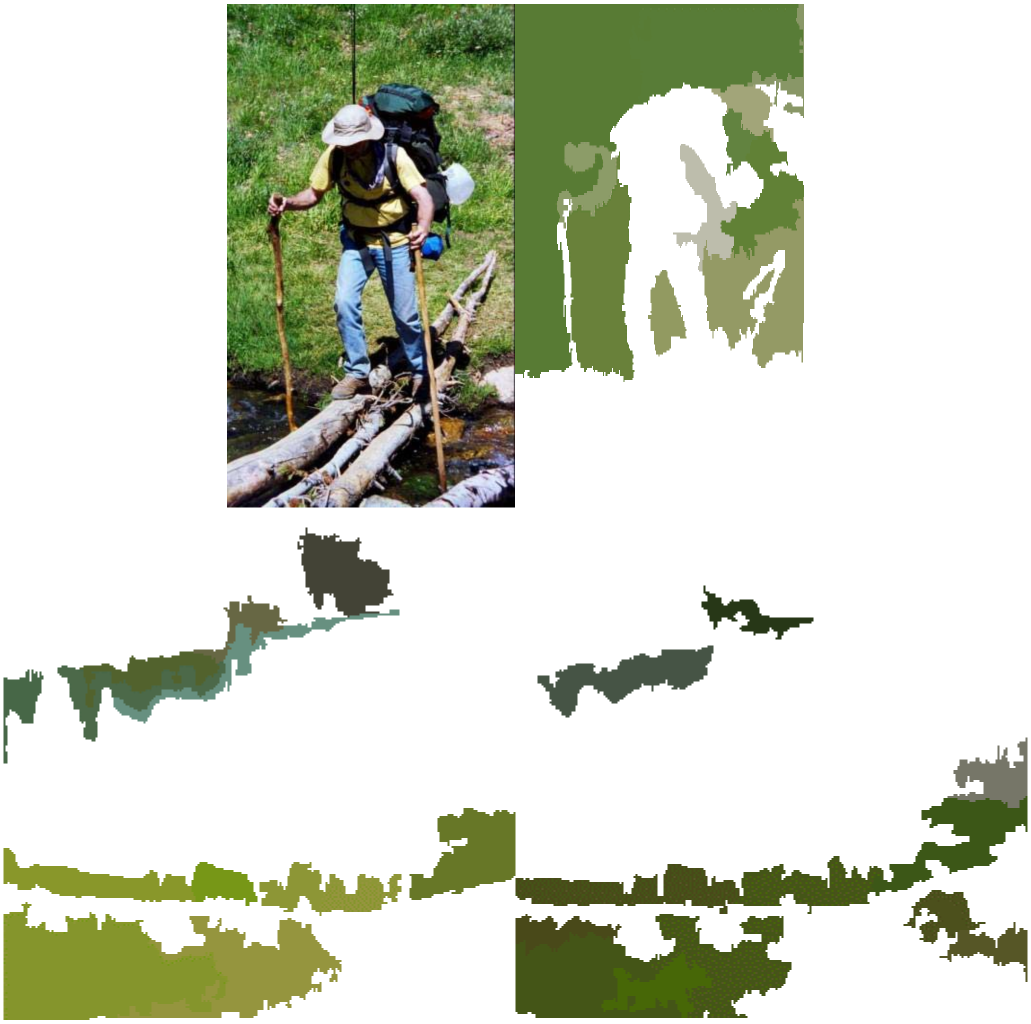


Figure 71. Detected areas.

We outline rules in our system in Table 20. Rules for sky, vegetation, and greenery relax the matching constraints, meaning that those areas do not need to match too closely. This is intuitive since such object areas often contain non-stationary elements such as clouds and trees that may not be *identical* in two images that are considered duplicates.

Table 20. Domain-specific rules used in the duplicate decision process.

Objects	Rules
Face	If *face_match*(face$_1$, face$_2$) then If *clothes_match*(face$_1$, face$_2$) then same person=1. *face_match:* two face areas match only if their similarity score is below a threshold *f.* Object similarity is based on location, area, aspect ratio, and color. In face_match all four have equal weights. *clothes_match:* matching of a rectangular area 3 times the width of the largest face width (of face$_1$ and face$_2$) and the same height as the largest face height. In the similarity score color is given a weight w_c and edges given a weight w_e, where w_c=0.9 and w_e=0.1. The clothes area is immediately below each face.
Sky	If *sky_match*(sky$_1$, sky$_2$) then Same_sky=1 The similarity between sky objects is analogous to face_match's. The following weights are used: color (0.5), area (0.4), location (0.3), aspect ration (0.2).
Vegetation	This rule is identical to the rule for sky.

The knowledge-based rules, then, are basically used to modify the similarity thresholds used to determine if two areas are from the same object (Figure 72). The self-saliency maps are also included in the similarity computation of the rules: if two image areas have very distinct saliency values they do not likely correspond to the same object.

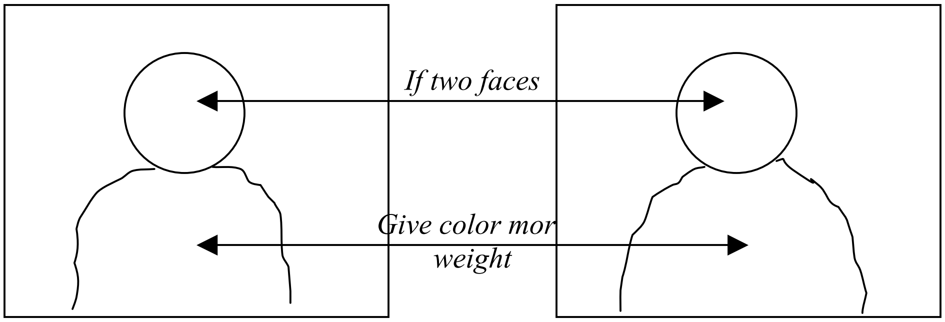


Figure 72. Schematic diagram of the way rules are used.

5.3.5.4 Final Step: Duplicate Decision

The duplicate similarity score is computed based on the change area mask of step I, and the rules that make use of object detectors. In particular, the final decision is based on the change area mask, the matched areas found using interest point growing and those found using the rules, and the mismatched areas.

In some cases, for example, objects will not be detected in the images. If that is the case the heuristic rules cannot be applied and therefore the duplicate decision is based on the mismatched areas and self-saliency maps. Mismatched areas are those that remain after the global change area mask is computed and interest point growing is performed.

When objects are detected and rules are applied, these may only cover a subset of the areas inside the change area mask. This is depicted in Figure 73. The final decision is made based on a weighted linear combination of the average correlation scores of the areas in Figure 73: those handled by the rules (objects and areas near objects), the mismatched areas, and the overall change area mask.

Note that these steps depend on the objects found and the rules, which determine the matching score between areas near those objects.

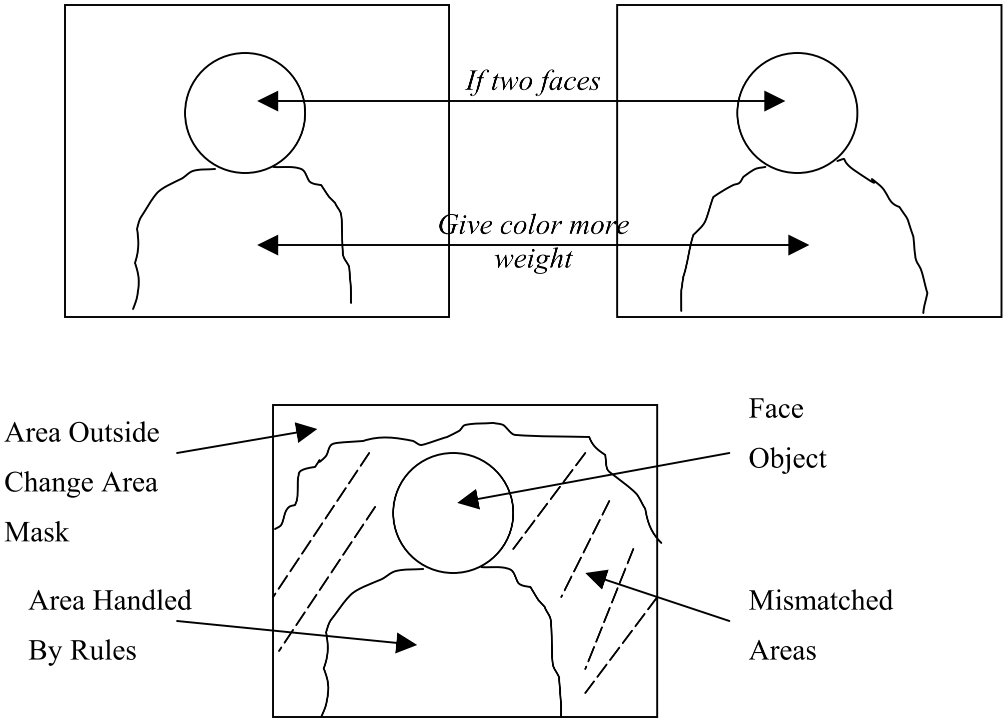


Figure 73. Areas used in final decision.

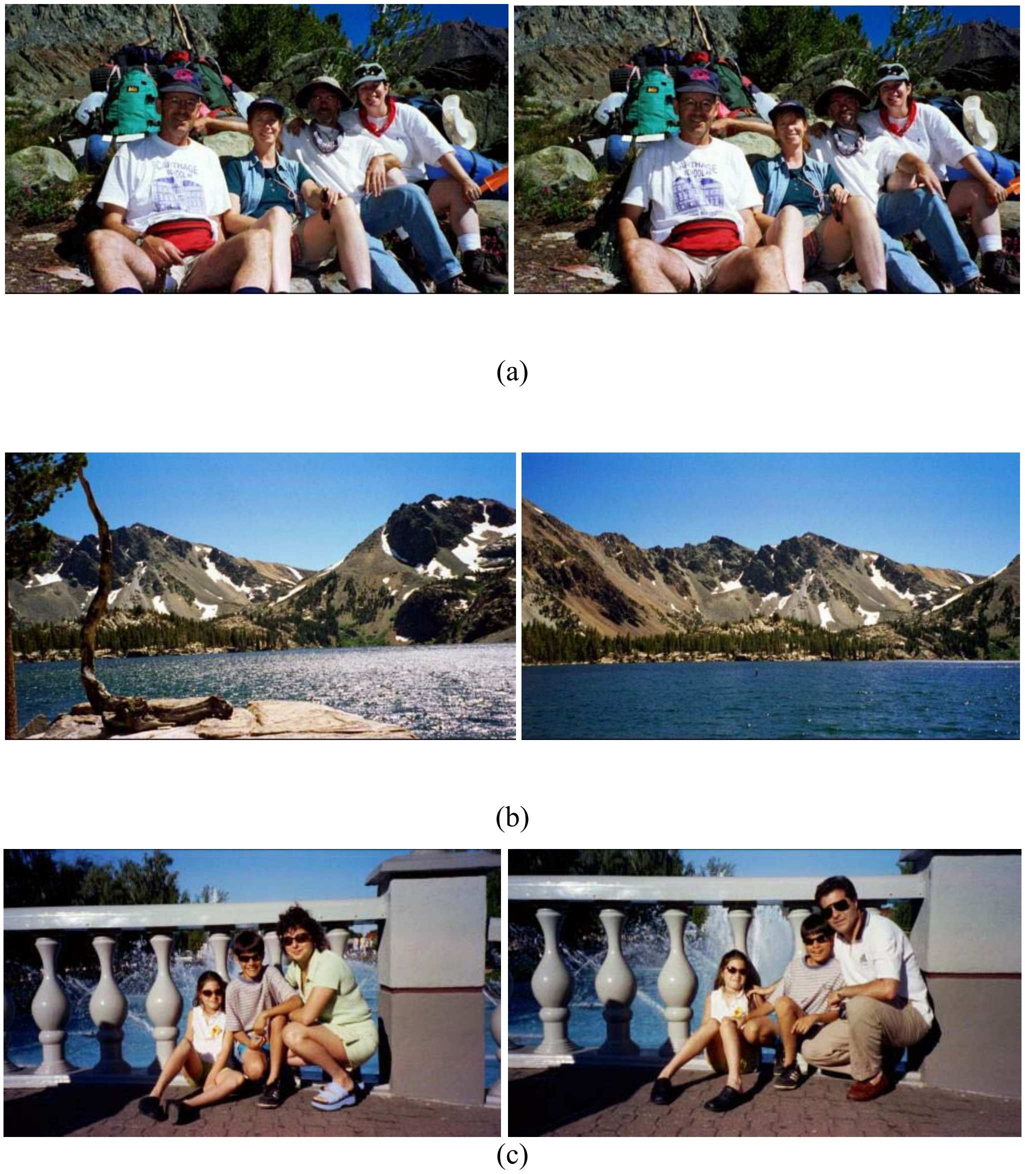
Matched areas and mismatched areas are treated separately because, as shown in Figure 73, they may not necessarily be covered by the rules or the global change area mask.

5.3.6 Ground Truth Construction And Analysis

A duplicate and similar non-duplicate database of consumer photographs was constructed and labeled by 10 people. Detailed analysis of this data is extremely important for understanding the duplicate problem, for testing the computational approach, and for identifying open issues.

We randomly chose 60 rolls of color APS and 35 mm film, from 54 real consumers[34]. Then, 255 image pairs (430 images) were manually selected from these rolls for inclusion in the database. We included only obvious duplicates, very similar non-duplicates, and pairs that could be labeled as either duplicates or non-duplicates. Image pairs that were clearly non-duplicates were excluded (Figure 74). All of the image pairs in our database, therefore, are visually very similar (please see other figures throughout the chapter).

[34] The database we described was constructed and labeled by researchers at Kodak. The analysis of the database was done by the author at Columbia.



(a)

(b)

(c)

Figure 74. An obvious duplicate (a), a "borderline" duplicate (b), a non-duplicate (c).

The initial selection of 255 duplicate candidate pairs was printed in color, each pair on a separate sheet, and given individually in random order to 10 people for labeling. Each individual was given the definition of duplicates at the beginning of section 5.3 and required to label each pair as either "duplicate" or "non-duplicate":

Duplicates: *two photographs are duplicates if they have the same content and composition, and were taken from the same angle and range.*

Consequently, each pair received 10 "votes." These results are summarized in Table 21. The number of pairs which all 10 subjects classified as "duplicate," for example, is 65, while only 43 of the pairs were classified as "non-duplicates" by all 10 subjects.

The table clearly indicates considerable subjectivity. Only in 43% of the pairs there was 100% agreement between all subjects (65 duplicates, 43 non-duplicates). In 4% of the cases there is no agreement at all on whether the pairs are duplicates or not. Where there is a non-unanimous majority agreement about the classification, in 23% of the cases the agreement is on duplicates, versus 43% of agreement for the non-duplicates. In other words, the number of overall agreements for duplicates and non-duplicates is roughly the same, but within the non-duplicate category there is less agreement.

Table 21. Database label distribution.

Positive duplicate votes	**No. of pairs**
10	65 (26%)
9	18 (7%)
8	14 (5%)
7	14 (5%)
6	13 (5%)
5	7 (4%)
4	15 (6%)
3	14 (5%)
2	26 (10%)
1	26 (10%)
0	43 (17%)
TOTAL	255 (100%)

Each candidate pair was manually classified by the author (independently of the ground truth labels) according to categories derived from Table 16. The name of each category indicates the most visible change in the image pair with the other elements approximately constant (*angle*: changes in angle only; *no change*: almost identical duplicates; *framing*: vertical vs. horizontal framing only, *horizontal shift:* camera rotation or translation, etc.). This is an approximate classification— it is often difficult to determine the type of change in viewpoint with the naked eye and several minor changes often occur at once. The first column in Table 22 shows the number of duplicates in the database that are in the respective category. The 100% positive and 100% negative columns show the number of image pairs within each category for which there was full agreement by the 10 subjects who labeled the ground truth. The percentage of pairs that received the same vote within each category are shown in parenthesis (e.g., 30% of the images with a 10 to 0 positive vote are in the "No change" category).

Image pairs for which there are no major visible changes in the parameters (*no change*) account for 30% of the duplicates in the ground truth. Using these statistics, the most important cases for duplicates are *no change* (30%), *horizontal/vertical shift* (22% combined)*, subject move* (17%), and *zoom* (15%). Solving the duplicate problem for only these types of cases accounts for 84% of the duplicates in the ground truth.

Images with "several changes" account for 44% of the non-duplicates. The most important non-duplicates are *several changes* (44%), and *subject change* (42%). Pairs in these categories account for 86% of the non-duplicates.

Table 22. Database categories.

Category	**No.** **In category**	**100% Positive**	**100% Negative**
Angle	9 (4%)	4 (6%)	0
Different background	3 (2%)	1 (2%)	0
No change	24 (9%)	20 (30%)	0
Framing	13 (5%)	2 (4%)	2 (5%)
Horizontal shift	18 (7%)	8 (12%)	1 (2%)
Vertical shift	8 (3%)	7 (10%)	0
Subject move	43 (17%)	11 (17%)	0
Subject change	37 (14%)	1 (2%)	18 (42%)
Several changes	64 (25%)	1 (2%)	19 (44%)
Zoom	36 (14%)	10 (15%)	3 (7%)
TOTAL	255	65	43

Deciding if two images are duplicates is highly subjective. In addition, when there is 100% agreement duplicate images can be visually different and non-duplicate images can be visually very similar. The pairs in Figure 75, for example, are visually dissimilar (note foreground/background parallax, which makes it impossible to align both simultaneously). The non-duplicate pairs in Figure 74 (c) and Figure 65, on the other hand, are visually very similar— high level semantic information is most likely used to decide that the images are not duplicates in those two cases. Figure 56 (b) also shows similar non-duplicates with 100% agreement. Very similar non-duplicates are more common than very dissimilar duplicates.

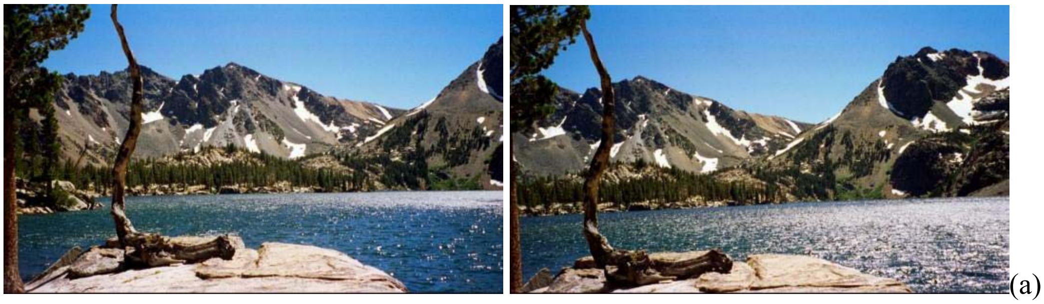


Figure 75. Non-identical duplicates.

Next we present results on this difficult database. Clearly, detection of duplicates with high accuracy is not attainable without solving several open issues in image understanding. The experiments, however, serve to investigate some of the limitations.

5.4 EXPERIMENTAL RESULTS

The database we chose for the experiments is a very difficult one because it contains only very similar photographs from real consumers. Therefore, high performance, even on the cases that may seem simple is unattainable. Consequently, we focus on a detailed analysis of the results in order to identify the open issues, strengths, and weaknesses of the approach we have presented. First we present the overall detection results. Then we show the performance of the object detectors.

5.4.1 Setup and Overall Performance

Each duplicate candidate *pair* was labeled by 10 subjects, therefore each duplicate candidate has 10 "votes" that determine its label. We used two thresholds, l_1 and l_2, to partition the database into *duplicates*, *non-duplicates*, and *unknowns*. First we tested our approach only on pairs for which there was 100% labeling agreement between all subjects. This subset of the database (set A) contains 108 images (65 duplicates and 43 non-duplicates). The framework was implemented using various available algorithms in [258][135].

For set A we achieved 64% precision and 97% recall.

In experiment B, we selected only 100 pairs of images from the duplicate database. The images in this set had different partition thresholds (t_1=3, t_2=7). If a pair of images had 7 or more "duplicate" labels, we labeled that pair as a duplicate (non-duplicate if it had 3 or less and unknown if it had 4, 5, or 6 "duplicate" votes). We obtained 52% precision and 70% recall.

Table 23 shows the number of false positives, misses, and hits in each of the categories for experiment B.

Table 23. Distribution of false positives, misses, and hits.

False Positives	
Category	**No.**
Several changes	12
Subject Replacement	5
No change	3
Zoom	1
Horizontal Shift	1
Subject Move	2

Misses	
Category	**No.**
Several changes	3
Zoom	1
Horizontal Shift	1
No change	2
Subject Move	1
Framing	1
Vertical Shift	1

Hits	
Category	**No.**
No Change	8
Subject Move	5
Several changes	3
Zoom	3
Different Background	1
Framing	1
Angle	3
Horizontal Shift	1
Vertical Shift	1

Most false positives are in the *several changes* and *subject replacement* categories. False positives occur mainly in the following scenarios: (1) cluttered scenes; (2) changes in a small part of the image; (3) changes not specific to an image area— the images are very similar but not quite the same. It is clear that most of the errors can be expected due to the subjectivity of the labeling.

False negatives, on the other hand, occur when there are semantically unimportant changes.

The classification results on this difficult database are not surprising. Achieving high performance requires solving several open issues in image understanding. In some of the non-duplicates the changes are very small, while in some of the duplicates the changes are large. Rejecting images with very small variations results in a higher number of misses. On the other hand, eliminating images that are very similar eliminates several true duplicates with small variations.

5.4.2 Object Detectors

We constructed several *Visual Detectors* using the framework of chapter 4. For each of the detectors we used the number of training examples indicated in Table 24. In total we obtained 2,270 positive examples at the region level and 686 positive examples of groups of regions.

We used different sets of images from training: news images, images from the MPGE-7 collection, and images from a personal collection. In other words, the images in the duplicate database were not used *at all* for the construction of the detectors. Table 24 shows the *baseline* performance for different classifiers using a nearest neighbor algorithm.

Table 24. Training sets for each of the detectors.

Class	**No. Positive Examples**	**No. Positive Images**	**No. Positive Group Examples**	**No. Positive Group Images**
Face	578	223	196	170
Sky	641	360	179	157
Vegetation	1051	155	311	136

Table 25. Performance of *region* level classifiers.

Class	Positive	Negative	Total	Precision	Recall
Face	578	578	1156	84%	94%
Vegetation	1051	1082	2133	80%	84%
Sky	641	672	1313	95%	96%

Table 26. Performance of groups of regions.

Class	Positive	Negative	Total	Precision	Recall
Face	196	196	392	90%	97%
Vegetation	311	311	622	81%	88%
Sky	179	179	358	93%	92%

5.4.3 Analysis of the Results

The duplicate image database that we used for our experiments contained only images from real consumers. The set included a wide variety of photographs: landscapes, group portraits, close-ups of objects, etc. There were slight differences between non-duplicates, and significant changes between duplicates. In addition, we found significant subjectivity in the labeling results.

On evaluating our computational framework, two obstacles played an important role: (1) subjectivity in the labeling results, and (2) use of semantic information in labeling.

We tested our framework on image pairs for which we found significant labeling agreement, and on image pairs for which we found full agreement. In other words, we focused on image pairs for which labeling results were more consistent. When the labels are consistent, however, we found that high level semantic information is used to decide if an image is a duplicate of another or not.

Solving the problem, however, even for the subset of images with consistent labels, is a formidable challenge. Within our framework, two immediate limitations can be highlighted: (1) number of object detectors, and (2) accuracy of object detectors. Increasing the number of detectors and their accuracy could be very helpful for particular types of duplicates. Having face recognition capability, for example, would certainly help eliminate many similar non-duplicates.

The biggest limitation, however, concerns the way in which similarity between images should be measured. In other words, even if we have all of the possible detectors available, we still need similarity models that help us decide when two objects are "duplicates" and when they are not. The duplicate decision, therefore, depends on the *semantic* content of the two images, particularly on the specific types of objects that appear in the images. Having detectors for an unlimited set of objects, therefore, is not sufficient: models of similarity between types of objects are needed. The application of these models depend on contextual information and domain knowledge.

5.5 DISCUSSION

In chapter 1 we argued that consumers are very likely to drive the most important multimedia applications of the future. Building such applications requires not only understanding visual information and its many levels of indexing (see chapters 2 and 3), but also understanding consumers, the types of images they produce, and ultimately how they are used.

In the particular problem studied in this chapter, we found that many factors play a role in organizing consumer photographs. Images can be grouped in multiple ways, according to any of the levels of the pyramid of chapter 3, based on *syntax* (e.g., colors, textures, etc.) or *semantics* (e.g., events, objects, etc.). Such grouping often depends on specific knowledge about the subjects, or events that were photographed. Consequently, providing tools for users to *subjectively* organize their own images remains an important research area.

Two of the biggest challenges in developing approaches to semi-automatically organize personal collections are the user's subjectivity and the use of high-level semantics in making decisions about how images should be organized. In addition, knowledge often goes beyond what is represented in the pictures alone and depends on contextual information. This information is often known only by the photographer, or by people who participated in the events photographed.

These issues were encountered both in trying to evaluate our clustering algorithm and in applying our framework to detect non-identical duplicate images.

In *STELLA*, future work includes performing a study of the way in which people organize their images (using the system's capability to track user operations), clustering images based on the output of visual detectors, and developing more complex composition features, among others.

In the duplicate framework, future work includes expanding the set of object detectors and using scene classifiers (e.g., applying face detection; landscape classifiers), incorporating more domain knowledge, and investigating ways to automatically choose the right transformation (e.g., Homography) depending on the type of image (e.g., portrait vs. landscape).

5.6 SUMMARY

In this chapter we presented *STELLA*, a system for semi-automatically organizing personal photography collections. We introduced a simple variation of Ward's hierarchical clustering algorithm that uses sequence information (e.g., the order in which photographs are made) to perform hierarchical clustering, and a new feature based on composition.

We also introduced the challenging problem of non-identical duplicate image detection in consumer photography. First, we modeled this difficult problem in terms of the components that cause changes between two images (*scene*, *camera*, and *image*) and presented a classification of duplicates based on the model. Then, we presented a novel framework for automatically detecting duplicate images. Our framework is based on the combination of low-level features, object detectors, and domain knowledge. Finally, we constructed a novel and extensive image duplicate database of 255 image pairs from real consumers labeled by 10 other people. We analyzed the labeling results in detail and presented experiments using our framework.

The main contributions of the chapter can be summarized as follows: (1) formulation of the duplicate problem using a new model of duplicates based on multiple view geometry (*scene*, *camera*, and *image*), (2) presentation of a novel framework to detect different types of duplicates (3) combination of low-level features (i.e., corners), object detection, and domain knowledge, and (4) construction and detailed analysis of a novel duplicate consumer image database.

6 CONCLUSIONS AND FUTURE DIRECTIONS

6.1 INTRODUCTION

In this chapter we summarize the work presented in this thesis and discuss extensions to our work.

We have addressed several challenging problems in automatic indexing and organization of visual information through user interaction at multiple levels. Our work is based on the hypothesis that the future of multimedia will bring about many novel and exciting applications in which consumers will be involved, much more than today, in organizing and using their images after they are created. This has required us to address three fundamental aspects of the problem of automatic indexing and organization of visual information. We focused on understanding visual information and the ways that people search it, on building computational frameworks that learn from user input, and on applying visual detectors in the domain of consumer photography.

6.1.1 Outline

In section 6.2 of we briefly summarize the content of the thesis and in section 6.3 we outline possible extensions to our work.

6.2 SUMMARY OF THESIS

In this thesis we addressed the problem of automatic indexing and organization of visual information through user interaction at multiple levels. We focused on the following three important areas: (1) understanding of visual content and the way users search and index it; (2) construction of flexible computational methods that learn how to automatically classify images and videos from user input at multiple levels; (3) integration of generic visual detectors in solving practical tasks in a specific domain (consumer photography).

In particular, we presented the following: (1) novel conceptual structures for classifying visual attributes (the *Multi-Level Indexing Pyramid*); (2) a novel framework for learning structured visual detectors from user input (the *Visual Apprentice*); (3) a new study of human eye movements in observing images of different visual categories; (4) a new framework for the detection of non-identical duplicate consumer photographs in an interactive consumer image organization system; (5) a detailed study of duplicate consumer photographs.

The *Multiple-Level Indexing Pyramid* we presented classifies visual attributes into ten levels. The first four levels are used for *syntactic* attributes and the remaining six for *semantic* attributes. The following ten levels are defined by the pyramid: *type/technique* (e.g., color, black and white), *global distribution* (e.g., color histogram), *local structure* (e.g., lines and circles), *composition* (e.g., leading angle), *generic object* (e.g., person), *generic scene* (e.g., indoors), *specific object* (e.g., Alex Jaimes), *specific scene* (e.g., central park in New York), *abstract object* (e.g., a dove represents peace), and *abstract scene* (e.g., paradise). We presented experiments to test the pyramid's ability to classify a full range of visual attributes generated in different tasks, to guide the image indexing process (i.e., manually describe images for future retrieval), and to improve retrieval in an image database (i.e., using keyword search). Our experiments showed that the pyramid is complete (classifies the full range of attributes generated), is useful in guiding the indexing task (more attributes were generated when individuals used the pyramid to index images) and improves retrieval (we found up to 80% improvement in precision when images are retrieved using keywords and a pyramid level is specified when compared against keyword retrieval alone). The pyramid has also been found to be complete by other researchers [37] in the sense that it is able to classify all of the visual attributes generated in manual image indexing tasks.

In the *Visual Apprentice* (VA), first a user defines a model via a multiple-level *definition hierarchy* (a scene consists of *objects*, *object-parts*, *perceptual areas*, and *regions*). Then, the user labels example images or videos based on the hierarchy (a handshake image contains two faces and a handshake) and visual features are extracted from each example. Finally, several machine learning algorithms are used to learn classifiers for different nodes of the hierarchy. The best classifiers and features are automatically selected to produce a *Visual Detector* (e.g., for a handshake), which is applied to new images or videos. We tested the *Visual Apprentice* framework in the construction of several detectors (handshake, baseball batting scene, face, sky, greenery, etc.) and applied the resulting detectors to different sets of images and videos (baseball video, news images, and consumer photographs in chapter 5). Our experiments demonstrate that the framework is flexible (can be easily applied to construct detectors with different hierarchies in different domains), and can produce reasonably good performance (92% accuracy in baseball batting scenes, 94% in handshake images in a news data set, 94% accuracy in detecting skies, etc.). Furthermore, the classifiers produced by the framework can be easily used by experts to construct highly accurate detectors for specific applications (e.g., in [275] rules were manually constructed based on detectors constructed using the *Visual Apprentice*).

In the human eye tracking experiments we examined variations in the way people look at images within and across different visual categories. We found consistency in the way people view images of *handshakes* and *centered objects*, and we found inconsistent patterns in the way that people view images of *crowds, landscapes* and images in a *miscellaneous* category. This finding is extremely important because it suggests that computational approaches could make use of eye tracking results within specific image categories. Although many eye tracking experiments have been performed in the past, to our knowledge, this is the first study that specifically compares eye movements across categories.

Finally, we presented *STELLA*, a system we have developed to help users semi-automatically organize their collections. In STELLA we introduced a simple composition-based feature and a hierarchical clustering algorithm (a simple variation of Ward's algorithm) that uses image sequence information to group similar images. Within STELLA we addressed the problem of automatic detection of non-identical consumer images. In particular, we presented a model of changes between non-identical duplicate photographs, a novel framework for the detection of non-identical duplicate consumer photographs, and a detailed analysis of a duplicate image database of consumer photographs (255 image pairs from 60 rolls of film from 54 real consumers). Our approach to automatically detect non-identical duplicate images is based on a multiple strategy that combines knowledge about the geometry of multiple views of the same scene, the extraction of low-level features, the detection of objects using the VA and domain knowledge about the duplicate problem in the consumer domain. We discussed different strategies to evaluate the clustering of consumer photographs and presented experiments using our framework to detect non-identical duplicate images. Although high performance in the duplicate database that we used is not achievable with the current state of the art, our analysis suggests that the framework presented is suitable for the non-identical duplicate detection problem. Furthermore, we were able to identify the main open issues in detecting non-identical duplicates, namely subjectivity and the use of high-level semantic analysis when decisions are made.

6.3 EXTENSIONS TO OUR WORK

In this section we provide a brief summary of open issues and future research directions. Further details can be found in the discussion sections of the corresponding chapters.

Multi-level Indexing Pyramid

We argued that the pyramid could be easily applied to video in a way that is similar to how it is applied to images. One of the differences brought about by the temporal nature of video is that descriptions change over time. A camera zoom, for example, can close in on an object so that the attributes that describe it are no longer local attributes but rather global ones. Such changes could be considered in extending the framework. How can we define time-varying structures to handle these cases?

In [142] we proposed applying the *Multi-level Indexing Pyramid* for classifying audio descriptors. Audio, however, can present additional challenges in particular for dividing sound sources into different levels. It is possible, for example, to describe sounds using abstract attributes (e.g., emotive attributes such as sad melody) to entire pieces. But separating the sounds, for instance, of particular musical performances might be difficult. We believe that future research on applying the pyramid to audio signals can be beneficial in helping us identify the structure of the signals for audio information retrieval.

The Visual Apprentice

There are many interesting ways in which the *VA* framework can be expanded. Active learning could be implemented so that, as the system learns, it helps the user find and label new examples. Applying the VA to video has additional implications that we have not explored. For example, dynamic definition hierarchies that change over time could be formulated. Another interesting possibility is to include time information in the construction of the models. Certain nodes, for example, could only appear after other nodes or events.

Eye Tracking

Experiments on eye tracking could be clearly expanded to many more categories. In addition, we could conduct similar experiments using video and other types of images. Construction of automatic classifiers from eye tracking input is an interesting possibility. A system like the VA could potentially be integrated with an eye tracker so that, for example, it would automatically learn to detect relevant objects and scenes as a user passively watches TV programs. Further research could investigate dependency of viewing patterns on age, gender, and nationality, among others. The effect of familiarity with the elements depicted could also be investigated, as well as the effect of lighting and location of the objects in the scene (e.g., foreground vs. background).

Duplicate Framework

Automatic detection of non-identical duplicate consumer photographs is a very challenging problem and the framework we proposed can be extended in many ways to address it. Clearly, constructing additional object and scene detectors would be very valuable. New similarity metrics based on "families" of objects could have a potentially important impact on duplicate detection. The way that we judge similarity between two people, for example, is quite different from the way we judge similarities between trees.

7 REFERENCES

Art

[1] J. Albers. Interaction of Color. Yale University Press. New Haven, 1963.

[2] R. Arnheim. *Art and Visual Perception: A psychology of the Creative Eye.* University of California Press. Berkeley, California, 1984.

[3] R. Arnheim. *New Essays on the Psychology of Art.* University of California Press, New York, 1986.

[4] S. Barnet. *A Short Guide to Writing About Art.* 5th Edition, Logman, New York, 1997.

[5] J. Barzun. The Use and Abuse of Art. Princeton University Press, Princeton, 1974.

[6] W. Benjamin, "The Work of Art in the Age of Mechanical Reproduction," 1935.

[7] J. Berger, *About Looking.* Pantheon Books, New York, 1980.

[8] J. Burnham, *The Structure of Art*, George Braziller Inc., New York, NY, 1971.

[9] G.T. Buswell. *How People Look at Pictures: A Study of the Psychology of Perception in Art.* University of Chicago press, Chicago, 1935.

[10] V.D. Coke. The Painter and the Photograph: from Delacroix to Warhol. University of New Mexico Press, Albuquerque, NM, 1964.

[11] B. Cole. The Informed Eye: Understanding Masterpieces of Western Art. Ivan R. Dee, Chicago, 1999.

[12] D.A. Dondis. *A primer of visual literacy.* MIT Press, Cambridge, Mass., 1973.

[13] H. Foster, editor, *Vision and Visuality: Discussions in Contemporary Culture*, No. 2, Dia Art Foundation, New York, 1988.

[14] H. Kreitler and S. Kreitler, *Psychology of the Arts*. Duke University Press, 1972.

[15] S. Lalvani. Photography, Vision, and the Production of Modern Bodies. State University of New York Press, Albany, 1996.

[16] E. Panofski. Studies in Iconology. Harper & Row, New York, 1962.

[17] F. Ritchin. In Our Own Image, the Coming Revolution in Photography: How Computer Technology is Changing Our View of the World. Aperture Foundation, New York, 1999.

[18] A. Scharf. Art and Photography. Pelican Books, Maryland, 1968.

[19] S. Sontang, On Photography. Dell Publishing, New York, 1973.

[20] N. Stangos, editor. Concepts of Modern Art. Thames and Hudson, New York, 1981.

[21] J. Szarkowski, Looking at Photographs. The Museum of Modern Art, New York, 1973.

[22] P.C. Vitz and A.B. Glimcher. Modern Art and Modern Science: the Parallel Analysis of Vision. Praeger Publishers, New York, 1984.

Cognitive Psychology

[23] S.L. Armstrong, L.R. Gleitman and H. Gleitman, "What Some Concepts Might Not Be," *Cognition,* 13:263-308, 1983.

[24] B. Burns, editor. Percepts, Concepts and Categories: the Representation and Processing of Information. Elsvier Academic Publishers, New York, 1992.

[25] B. Burns, "Perceived Similarity in Perceptual and Conceptual Development: The Influence of Category Information on Perceptual Organization," in B. Burns, editor, *Percepts, Concepts and categories*, pages 175-231, Elsvier Academic Publishers, New York, 1992.

[26] S. Harnad, editor. *Categorical Perception: the Groundwork of Cognition.* Cambridge University Press, New York, 1987.

[27] W.R. Hendee, P. N.T. Wells, editors. *The Perception of Visual Information.* Second Edition. Springer Verlag, New York, 1997.

[28] F.C. Keil, and M.H. Kelly, "Developmental Changes in Category Structure," in S. Hanard, editor, *Categorical Perception: the Groundwork of eCognition,* pages 491-510*,* Cambridge University Press, New York, 1987.

[29] M.W. Morris and G.L. Murphy, "Converging Operations on a Basic Level in Event Taxonomies," *Memory and Cognition,* 18(4): 407-418*,* 1990.

[30] A. Rifkin, "Evidence for a Basic Level in Event Taxonomies," *Memory and Cognition,* 13(6):538-556, 1985.

[31] E. Rosch and C.B. Mervis, "Family Resemblances: Studies in the Internal Structure of Categories," *Cognitive Psychology,* 7:573-605, 1975.

[32] E. Rosch, C. B. Mervis, W.D. Gray, D.M. Johnson and P. Boyes-Braem, "Basic Objects in Natural Categories," *Cognitive Psychology,* 8:382-439, 1976.

[33] L.B. Smith and D. Heise. "Perceptual Similarity and Conceptual Structure," in B. Burns, editor, *Percepts, Concepts and Categories,* pages 233-272, Elsvier Academic Publishers, New York, 1992.

[34] E.R. Tufte. Visual Explanations: Images and Qualities, Evidence and Narrative. Graphics press, Cheshire, Connecticut, 1997.

[35] B. Tversky and K. Hemenway, "Categories of Environmental Scenes," Cognitive Psychology, 15:121-149, 1983.

[36] A. Tversky, "Features of Similarity," Psychological Review, 84(4):327-352, July 1977.

Information Sciences

[37] M.A. Burke, "Personal Construct Theory as a Research Tool for Analsing User Perceptions of Photographs," in proceedings of *International Conference on Image and Video Retrieval (CIVR 2002)*, London, UK, July, 2002.

[38] E.T. Davis, "A Prototype item-level index to the civil war photographic collection of the Ohio Historical Society," *Master of Library Science thesis*, Kent State University, August, 1997.

[39] Dublin Core v. 1.1 (*http://purl.oclc.org/metadata/dublin_core*), February 2000.

[40] J.P. Eakins, "Design Criteria for a Shape Retrieval System," *Computers in Industry,* 21(2):167-184, 1993.

[41] P.G.B. Enser, "Query Analysis in a Visual Information Retrieval Context," *Journal of Document and Text Management*, 1(1):25-39, 1993.

[42] R. Fidel, T.B. Hahn, E.M. Rasmussen, and P.J. Smith, editors. *Challenges in indexing electronic text and images*. *ASIS Monograph series*. Learned Information Inc., 1994.

[43] Getty Information Institute. Categories for the Description of Works of Art (*http://www.getty.edu/gri/standard/cdwa/*), December 1999.

[44] Getty Research Institute. Getty Vocabulary Program: Art & Architecture Thesaurus (*http://shiva.pub.getty.edu/aat_browser/*), February 2000.

[45] A. Jaimes, C. Jorgensen, A.B. Benitez, and S.-F. Chang, "Experiments in Indexing Multimedia Data at Multiple Levels", in *ASIS&T Advances in Classification Research*, vol. 11, 2001.

[46] B.J., Jones, "Variability and Universality in Human Image Processing," in Francis T. Marchese, editor, *Understanding Images: Finding Meaning in Digital Imagery*, TELOS, Santa Clara, CA, 1995.

[47] C. Jorgensen, "Image Attributes," *Ph.D. thesis*, Syracuse University, 1995.

[48] C. Jorgensen, "Image attributes in describing tasks: an investigation," *Information Processing & Management,* 34(2/3):161-174, 1998.

[49] C. Jorgensen, "Classifying Images: Criteria for Grouping as Revealed in a Sorting Task," *Advances in Classification Research*, vol. 6:45-64, 1995.

[50] C. Jörgensen, "Testing an image description template," in Steve Hardin (Ed.), Proceedings of the 59th Annual Meeting of the American Society for Information Science ASIS '96, pp. 209-213, Medford NJ: Information Today, Oct. 1996.

[51] C. Jörgensen, "Retrieving the unretrievable: art, aesthetics, and emotion in image retrieval systems," in Bernice E. Rogowitz, Thrasyvoulos N. Pappas (Eds.), Proceedings SPIE Vol. 3644 Human Vision and Electronic Imaging IV, pp. 348-355, San Jose, CA, 1999.

[52] C. Jörgensen, "Image indexing: an analysis of selected classification systems in relation to image attributes named by naive users," Final Report to the Office of Sponsored Research, Online Computer Library Center, 2000.

[53] C. Jörgensen, and R. Srihari, "Creating a web-based image database for benchmarking image retrieval systems: a progress report." In Bernice E. Rogowitz, Thrasyvoulos N. Pappas (Eds), Proceedings SPIE Vol. 3644 Human Vision and Electronic Imaging IV, pp. 534-541, San Jose, CA, 1999.

[54] Library of Congress. *Thesaurus for Graphic Materials I: Subject Terms*, February 2000. *http://www.loc.gov/rr/print/tgm1*

[55] K. Markey, "Computer Assisted Construction of a Thematic Catalog of Primary and Secondary Subject Matter," *Visual Resources*, vol. 3:16-49, Gordon and Breach, Science Publishers, 1983.

[56] B. Orbach, "So That Others May See: Tools for Cataloguing Still Images," in *Describing Archival Materials: the Use of the MARC AMC Format,* pages 163-191, Haworth Press, 1990.

[57] E.B. Parker. *LC Thesaurus for Graphic Materials: Topical Terms for Subject Access*. Library of Congress, Washington DC, 1987.

[58] E.M. Rasmussen, "Indexing Images," *Annual Review of Information Science and Technology (ARIST)*, vol. 32:169-196, 1997.

[59] H. Roberts, "Do You Have Any Pictures Of...? Subject Access to Works of Art In Visual Collections and Book Reproductions," *Art Documentation*, pages 87-90, Fall, 1988.

[60] S.S. Layne, "Some Issues in the Indexing of Images," *Journal of the American Society for Information Science,* 45(8):583-588, 1994.

[61] S.S. Layne, "Analyzing the Subject of a Picture: A Theoretical Approach," *Cataloguing and Classification Quarterly*, 6(3):39-62, The Haworth Press, 1986.

[62] J. Turner, "Determining the Subject content of still and moving image documents for storage and retrieval: an experimental investigation," *Ph.D. thesis*, University of Toronto, 1994.

[63] J. Turner, "Cross-Language Transfer of Indexing Concepts for Storage and Retrieval of Moving Images: Preliminary Results," in *proceedings of ASIS 1996 Annual Conference*, October, 1996.

[64] VRA Data Standards Committee. Visual Resources Association. (January 2000). The VRA core categories for visual resources, version 2.0. Available: *http://www.vra.oberlin.edu/vc.html*

Visual Information Retrieval/Computer Vision/Other

[65] D. Aha, Editor. *Lazy Learning*. Kluwer Academic Publishers, The Netherlands, 1997.

[66] A.V. Aho, S.-F. Chang, K.R. McKeown, D. Radev, J.R. Smith and K. Zaman, "Columbia Digital News System. An Environment for Briefing and Search over Multimedia Information," in *proceedings IEEE International Conference on the Advances in Digital Libraries (ADL) International Forum on Advances in Research and Technology*, pages 82-94, Washington, D.C. May, 1997.

[67] A. Aner and J.R. Kender, "Video Summaries Through Mosaic-Based Shot and Scene Clustering," in proc. *European Conference on Computer Vision*, Denmark, May 2002.

[68] M. Arani, et al., "Mosaic Based Representations of Video Sequences and Their Applications," in Proc. 5th *International Conference on Computer Vision*, pp. 605-611, 1995.

[69] *Art and Optics Conference: Toward An Evaluation of David Hockney's New Theories Regarding Opticality in Western Painting of the Past 600 Years*, New York, NY, December 2001 (*http://www.artandoptics.com*)

[70] A. Amir, and M. Lindenbaum, "A generic grouping algorithm and its quantitative analysis," *IEEE Transactions on Pattern Recognition and Machine Intelligence (PAMI)*, 20(2):186-192, February 1998.

[71] K. M. Andress, and A.C. Kak, "Evidence Accumulation and Flow of Control in a Hierarchical Reasoning System," *Artificial Intelligence Magazine*, 9(2):75-94, 1988.

[72] J.R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R.C. Jain, and C. Shu, "The VIRAGE Image Search Engine: An Open Framework for Image Management," in *proceedings of SPIE Storage and Retrieval for Still Image and Video Databases IV*, vol. 2670:76-87, February 1996.

[73] S. Baker and I. Matthews, "Equivalence and Efficiency of Image Alignment Algorithms," *Proceedings of the 2001 IEEE Conference on Computer Vision and Pattern Recognition*, December, 2001.

[74] D.M. Bates and D.G. Watts. *Nonlinear Regression Analysis and its Applications.* John Wiley & Sons, New York, 1988.

[75] A. B. Benitez and S.-F. Chang, "Multimedia Knowledge Integration, Summarization and Evaluation," in *proceedings of the 2002 International Workshop On Multimedia Data Mining in conjunction with the International Conference on Knowledge Discovery & Data Mining (MDM/KDD-2002)*, Edmonton, Alberta, Canada, July 23-26, 2002

[76] A.B. Benitez, A. Jaimes, S.-F. Chang, J.R. Smith, and C.-S. Li, "Fundamental Entity-Relationship Models for the Generic Audio Visual DS," Contribution to *ISO/IEC JTC1/SC29/WG11 MPEG99/M4754*, Vancouver, Canada, July 1999.

[77] A.B. Benitez, A. Jaimes, S. Paek, S.-F. Chang, **"**Fundamental Entity-Relation Models for a Multimedia Archive DS**,"** Contribution to *ISO/IEC JTC1/SC29/WG11 MPEG99/M4755*, Vancouver, Canada, July 1999

[78] A.B. Benitez, S. Paek, S.-F. Chang, A. Huang, A. Puri, C.-S. Li, J.R. Smith, L. D. Bergman, C. Judice, "Object-Based Multimedia Description Schemes and Applications for MPEG-7," *Image Communications Journal*, invited paper on *Special Issue on MPEG-7*, vol. 16:235-269, 1999.

[79] A. Benitez, A. Jaimes, C. Jorgensen, and S.-F. Chang, "Report of CE on Multilevel Indexing Pyramid," contribution to *ISO/IEC JTC1/SC29/WG11 MPEG00/M6495*, La Baule, France, October 2000.

[80] L.D. Bergman, and V. Castelli, editors. *Image Databases, Search and Retrieval of Digital Imagery.* John Wiley & Sons, New York (forthcoming).

[81] L.D. Bergman, V. Castelli, C.-S. Li and J.R. Smith, "SPIRE, a Digital Library for Scientific Information," *International Journal of Digital Libraries, Special Issue* "In the Tradition of the Alexandrian Scholars", 3(1):85-99, July 2000.

[82] T.O. Binford, "Survery of Model-Based Image Analysis Systems," *International Journal of Robotics Research*, 1(1), Spring 1992.

[83] G. Briscoe and T. Caelli, editors. A *Compendium of Machine Learning,* Ablex series in Artificial Intelligence. Norwood, NJ, 1996.

[84] R. Brooks, "Symbolic Reasoning Among 3-dimensional Models and 2-Dimensional Images," *Artificial Intelligence,* 17:285-349, 1981.

[85] L.G. Brown, "A Survey of Image Registration Techniques," *ACM Computing Surveys*, 24(4), pp. 325-376, 1992.

[86] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Region-Based Image Querying," in *proceedings IEEE Workshop on Content-Based Access of Image and Video Libraries*, pages 42-49, 1997.

[87] V. Castelli L.D. Bergman, eds. *Image Databases: Search and Retrieval of Digital Imagery*. John Wiley and Sons, New York, 2002.

[88] E.Y. Chang, et al., "RIME: A Replicated Image Detector for the World-Wide Web", *SPIE Vol. 3527*, pp. 68-67, 1998.

[89] S.-F. Chang, W. Chen, H.J. Meng, H. Sundaram, and D. Zhong, "A Fully Automatic Content-Based Video Search Engine Supporting Multi-Object Spatio-temporal Queries," *IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on Image and Video Processing for Interactive Multimedia*, 8(5):602-615, September 1998.

[90] S.-F. Chang, Q. Huang, T.S. Huang, A. Puri, and B. Shahraray., "Multimedia Search and Retrieval," in A. Puri and T. Chen, eds., *Advances in Multimedia: Systems, Standards, and Networks*, Marcel Dekker, 1999.

[91] S.-F. Chang, T. Sikora, and A. Puri, "Overview of the MPEG-7 Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, special issue on MPEG-7, June 2001 (to appear).

[92] S.-F. Chang, J.R. Smith, M. Beigi and A. Benitez, "Visual Information Retrieval from Large Distributed On-line Repositories," *Communications of the ACM*, 40(12):63-71, December, 1997.

[93] S.-F. Chang, B. Chen and H. Sundaram, "Semantic Visual Templates: Linking Visual Features to Semantics," *in proceedings of International Conference on Image Processing (ICIP '98), Workshop on Content Based Video Search and Retrieval*, pages 531-535, Chicago IL, October, 1998.

[94] G.E. Christensen et al., "Consistent Image Registration", *IEEE Trans on Medical Imaging*, Vol. 20 No. 7, July 2001.

[95] T.-S. Chua, W.-C. Low, and C.-X. Chu, "Relevance Feedback Techniques for Color-based Image Retrieval," in *proceedings of IEEE Multimedia Modeling(MMM '98),* pages 24-31, 1998.

[96] Clarinet newsgroup (*http://www.clarinet.com*)

[97] W. Cohen, "Learning Trees and Rules with Set-valued Features," in *proceedings of Thirteenth National Conference on Artificial Intelligence (AAAI 1996)*, Portland, Oregon, August 1996.

[98] J.M. Corridoni, A. Del Bimbo, and P. Pala, "Retrieval of Paintings using Effects Induced by Color Features," in *proceedings of IEEE Workshop on Content-Based Access of Image and Video Databases*, pages 2-11, Bombay, India, January 1998.

[99] I.J. Cox, M.L. Mller, T.P. Minka, and P.N. Yianilos, "An Optimized Interaction Strategy for Bayesian Relevance Feedback," in *proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR '98),* pages 553-558, 1998.

[100] T. Darell and M. Covell, "Correspondence with Cumulative Similarity Transforms," *IEEE PAMI*, Vol. 23, No. 2, pp. 222-227, Feb. 2001.

[101] M. Das and E.M. Riseman, "Feature Selection for Robust Color Image Retrieval," in *proceedings of DARPA IUW,* New Orleans, LA, 1997.

[102] A. Del Bimbo. *Visual Information Retrieval*. Morgan Kaufmann Publishers, San Francisco, USA, 1999.

[103] T.G. Dietterich, "Proper Statistical Tests for Comparing Supervised Classification Learning Algorithms," *Technical Report*, Department of Computer Science, Oregon State University, 1996.

[104] N. Dimitrova, Y. Chen, and L. Nikolovska, "Visual Associations in DejaVideo," in *proceedings of the Fourth Asian Conference on Computer Vision (ACCV 2000)*, Vol. 1, pp. 353-362, Taipei, Taiwan, January 2000.

[105] N. Dimitrova. *Content Classification and Retrieval of Digital Video on Motion Recovery*. Ph.D. thesis, Arizona State University, Tempe, AZ, 1995.

[106] D. Doermann, H. Li, and O. Kia, "The Detection of Duplicates in Document Image Databases," in Proc. *4th International Conference on Document Analysis and Recognition*, Vol. 1, pp. 314-318, 1997.

[107] D. Dori and H. Hel-Or, "Semantic Content Based Image Retrieval Using Object-Process Diagrams," in *proceedings 7th International Workshop on Structural and Syntactic Pattern Recognition (SSPR 98)*, Sydney, Australia, 1998.

[108] R.O. Duda, P.E. Hart, D.G. Stork, *Pattern Classification*, John Wiley & Sons, New York, 2001.

[109] B.A. Draper, “Learning Object Recognition Strategies,” *Ph.D. thesis*, Computer Science Department, University of Massachusetts, MA, 1993.

[110] G. Durand, C. Thienot, and P. Faudemay, “Extraction of Composite Visual Objects from Audiovisual Materials,” in *proceedings of SPIE Multimedia and Archiving Systems IV*, vol. 3846:194-203, Boston, MA, 1999.

[111] J. Durkin. *Expert Systems: Design and Development.* Prentice Hall, NJ, 1994.

[112] O. Faugeras, Q.T. Luong, and T. Papadopoulo. *The Geometry of Multiple Images*. MIT Press, 2001.

[113] D.A. Forsyth and M. Fleck, “Body Plans,” in proceedings of *IEEE Computer Vision and Pattern Recognition (CVPR '97)*, pages 678-683, San Juan, Puerto Rico, 1997.

[114] C. Frankel, M.J. Swain and V. Athitsos, “WebSeer: An Image Search Engine for the World Wide Web,” *University of Chicago Technical Report TR-96-14*, July 31, 1996.

[115] S. Geman, D.F. Potter, and Z. Chi, “Composition Systems,” *Technical Report Division of Applied Mathematics*, Brown University, 1998.

[116] J. Gluckman and S.K. Nayar, “Rectifying Transformations that Minimize Resampling Effects,” in *Proc. IEEE CVPR* ’02, Hawaii, Dec. 2002.

[117] M. Gorkani, and R.W. Picard, “Texture Orientation for Sorting Photos at a Glance,” in proceedings of *IEEE International Conference on Pattern Recognition (ICPR '94)*, vol. 1:459-464, Jerusalem, Israel, October 1994.

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

[119] W.I. Grosky, and R. Mehrotra, “Index-Based Object Recognition in Pictoral Data Management,” *Computer Vision, Graphics, and Image Processing (CVGIP)* vol. 52:416-436, 1990.

[120] B. Gunsel, A.M. Ferman, and A.M. Tekalp, “Temporal video segmentation using unsupervised clustering and semantic object tracking,” *IS&T/SPIE Journal of Electronic Imaging,* 7(3):592-604, July 1998.

[121] A. Hampapur and R. Bolle “Comparison of Distance Measures for Video Copy Detection,” *IEEE ICME 2001.*

[122] A.R. Hanson, and E.M. Riseman, "VISIONS: A computer System for Interpreting Scenes," in Hanson and Riseman, editors, *Computer Vision Systems*, pages 303-333, Academic Press, New York, 1978.

[123] C.J. Harris and M. Stephens, "A Combined Corner and Edge Detector," in proceedings of the 4th Alvey Vision Conference, pp. 147-151, Manchester, England, 1988.

[124] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, UK, 2000.

[125] T. Hastie and D. Pregibon, "Shrinking Trees," *AT&T Bell Laboratories Technical Report*, 1990.

[126] D. Healey, "Preattentive Processing in Visualization," *http://www.cs.berkeley.edu/~healey/PP/PP.shtml*

[127] R.S. Heller and C.D. Martin, "A Media Taxonomy," *IEEE Multimedia Magazine*, 2(4):36-45, Winter 1995.

[128] D. Hernandez, "Qualitative Representation of Spatial Knowledge," *Lecture Notes in Artificial Intelligence*, 804, Springer-Verlag, Berlin, 1994.

[129] C.R. Hick. *Fundamental Concepts in the Design of Experiments*. Holt, Rinehart, and Wilson, New York, 1982.

[130] K. Hirata and T. Kato, "Query by Visual Example", Proc. *Intl. Conf. On Extending Database Technology EDBT92*, pp. 56-71, March, 1992.

[131] N. Hirzalla, B. Falchuk, and A. Karmouch, "A Temporal Model for Interactive Multimedia Scenarios," *IEEE Multimedia Magazine*, 2(3):24-31 Fall 1995.

[132] Y.S. Ho and A. Gersho, "Classified Transform Coding of Images Using Vector Quantization," *IEEE International Conference on Acoustics and Signal Processing (ICASP '89)*, vol. 3:1890-1893, May 1989.

[133] D. Hockney, *Secret Knowledge: Rediscovering the Lost Techniques of the Old Masters,* Viking Studio, New York, 2001.

[134] K. Ikeuchi, and M. Veloso, editors. *Symbolic Visual Learning*. Oxford University Press, New York, 1997.

[135] Intel OpenCV Open Source Computer Vision Library *(http://www.intel.com/research/mrl/research/opencv)*

[136] C.E. Jacobs, A. Finkelstein, D.H. Salesin, "Fast Multiresolution Image Querying," in *proceedings of ACM Conference on Computer Graphics (SIGGRAPH)*, pages 277-286, August 1995.

[137] A. Jaimes, M. Naphade, H. Nock, J.R. Smith, and B. Tseng, "Context Enhanced Video Understanding," in *proceedings of SPIE Storage and Media Databases 2003*, Santa Clara, CA, 2003.

[138] A. Jaimes, and S.-F. Chang, "Concepts and Techniques for Indexing Visual Semantics", in L.D. Bergman, and V. Castelli, editors. *Image Databases, Search and Retrieval of Digital Imagery.* John Wiley & Sons, New York, 2002.

[139] A. Jaimes, and S.-F. Chang, "Learning Visual Object Detectors from User Input", invited paper, *International Journal of Image and Graphics (IJIG), special issue on Image and Video Databases*, August, 2001.

[140] A. Jaimes, J.B. Pelz, T. Grabowski, J. Babcock, and S.-F. Chang, "Using Human Observers' Eye Movements in Automatic Image Classifiers" in *proceedings of SPIE Human Vision and Electronic Imaging VI*, San Jose, CA, 2001.

[141] A. Jaimes, A.B. Benitez, S.-F. Chang, and A.C. Loui, "Discovering Recurrent Visual Semantics in Consumer Photographs," invited paper, *International Conference on Image Processing (ICIP 2000), Special Session on Semantic Feature Extraction in Consumer Contents*, Vancouver, Canada, September 10-13, 2000.

[142] A. Jaimes, A. B. Benitez, S.-F. Chang, "Multiple Level Classification of Audio Descriptors", *ISO/IEC JTC1/SC29/WG11 MPEG00/M6114*, Geneva, Switzerland, May/June 2000.

[143] A. Jaimes, C. Jorgensen, A.B. Benitez, and S.-F. Chang, "Multiple Level Classification of Visual Descriptors in the Generic AV DS," Contribution to *ISO/IEC JTC1/SC29/WG11 MPEG99/M5251*, Melbourne, Australia, October 1999.

[144] A. Jaimes and S.-F. Chang, "Learning Visual Object Filters and Agents for on-line Media," *ADVENT Project Technical Report*, Columbia University, June, 1999.

[145] A. Jaimes and S.-F. Chang, "Model-Based Classification of Visual Information for Content-Based Retrieval," in *proceedings of SPIE Storage and Retrieval for Image and Video Databases VII*, vol. 3656:402-414, San Jose, CA, January 1999.

[146] A. Jaimes and S.-F. Chang, "Integrating Multiple Classifiers in Visual Object Detectors Learned from User Input," Invited paper, session on Image and Video Databases*, 4th Asian Conference on Computer Vision (ACCV 2000)*, vol. 1:376-381, Taipei, Taiwan, January 8-11, 2000.

[147] A. Jaimes and S.-F. Chang, "Automatic Selection of Visual Features and Classifiers," in *proceedings of SPIE Storage and Retrieval for Media Databases 2000*, vol. 3972:346-358, San Jose, CA, January 2000.

[148] A. Jaimes and S.-F. Chang, "A Conceptual Framework for Indexing Visual Information at Multiple Levels," in *proceedings of SPIE Internet Imaging 2000*, vol. 3964:2-15. San Jose, CA, January 2000.

[149] A. Jaimes, S.-F. Chang, and A.C. Loui, "Duplicate Detection in Consumer Photography," in *proceedings of ACM Multimedia 2002*, Juan Les Pines, France, December 2002.

[150] A. Jain and R. Dubes, *Algorithms for Clustering Data*, Prentice-Hall, 1988.

[151] R. Jain and A. Hampapur, "Metadata in Video Databases," *SIGMOD Record, Special Issue on Metadata for Digital Media*, December, 1994.

[152] A. Jain and D. Zongker, "Feature Selection: Evaluation, Application, and Small Sample Performance," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 19(2):153-158, February 1997.

[153] R.A. Jarvis and E.A. Patrick, "Clustering Using similarity measure based on Shared Near Neighbors," *IEEE Transactions on Computers*, 22(11), November, 1973.

[154] G.H. John, R. Kohavi, and K. Pfleger, "Irrelevant Features and the Subset Selection Problem," in *proceedings of 11th International Conference on Machine Learning (ICML '94)*, pages 121-129, 1994.

[155] C. Jorgensen, A. Jaimes, A. B. Benitez, and S.-F. Chang, "A Conceptual Framework and Research for Classifying Visual Descriptors," invited paper, *Journal of the American Society for Information Science (JASIS), special issue on "Image Access: Bridging Multiple Needs and Multiple Perspectives,"* Spring 2001.

[156] R. Kasturi and R.C. Jain, editors. *Computer Vision: Principles*. IEEE Computer Society Press, 1991.

[157] J.M. Keller, M.R. Gray and J.A. Givens Jr., "A fuzzy k-nearest neighbor algorithm," *IEEE Transactions on Systems Man, and Cybernetics,* 15(4):580-585, July/August 1985.

[158] D. Kirsh, “Interactivity and Multimedia Interface,” *Instructional Science,* 25(2):79-96. 1997.

[159] R. Kohavi, “Feature Subset Selection Using the Wrapper Model: Overfitting and Dynamic Search Space Topology,” in *proceedings of First International Conference on Knowledge Discovery and Data Mining*, pages 192-197, 1995.

[160] R. Kohavi, “A Study of Cross-validation and Bootstrap for Accuracy Estimation and Model Selection,” in *proceedings of the 14th International Joint Conference on Artificial Intelligence (JCAI '95)*, pages 1137-1143, Morgan Kaufmann Publishers, Inc., 1995.

[161] R. Kohavi, G. John, R. Long, D. Manley, and K. Pfleger, “MLC++: A Machine Learning Library in C++,” in *proceedings of Conference on Tools with Artificial Intelligence 94*, 1994.

[162] D. Koller and M. Sahami, “Toward Optimal Feature Selection,” in *proceedings of 13th International Conference on Machine Learning (ICML)*, Bary, Italy, July 1996.

[163] M. Koster, “Robots in the Web: Threat or Treat?,” *Connexions*, 9(4), April 1995.

[164] A. Kuchinsky, et al., “FotoFile: A Consumer Multimedia Organization and Retrieval System”, *ACM Conf. On Computer and Human Interaction*, Pittsburg, PA 1999.

[165] E. Lang, K.-U. Carstensen, and G. Simmons. Modelling Spatial Knowledge on a Linguistic Basis, Lecture Notes in Artificial Intelligence, 481, Springer-Verlag, Berlin, 1991.

[166] C.S. Lee, W.-Y. Ma, and H.J. Zhang, “Information Embedding Based on User’s Relevance Feedback for Image Retrieval,” in *proceedings of SPIE Multimedia Storage and Archiving Systems IV*, vol. 3846:294-304, Boston, MA, September 1999.

[167] W.-H. Lin and A. Haupman, “A Wearable Digital Library of Personal Conversations,” in proc. *Joint Confernce on Digital Libraries (JCDL ’02)*, Portland, Oregon, July, 2002.

[168] A. Loui and A. E. Savakis, "Automatic Image Event Segmentation and Quality Screening for Albuming Applications*", ICME 2000*, New York City, July 2000.

[169] A. Loui, and M. Wood, "A software system for automatic albuming of consumer pictures," *Proc. ACM Multimedia 99*, pp. 159-162, Orlando, Fl., Oct. 30-Nov. 5, 1999.

[170] L. Li and M.K.H. Leung, "Integrating Intensity and Texture Differences for Robust Change Detection," *IEEE Trans. On Image Processing*, Vol. 11, No. 2, pp. 105-112, Feb. 2002.

[171] P. Lipson, "Context and Configuration Based Scene Classification," *Ph.D. thesis,* MIT Electrical and Computer Science Department, September 1996.

[172] F. Liu and R.W. Picard, "Periodicity, directionality and randomness: Wold features for image modeling and retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 18(7):722-733, July 1996.

[173] G.L. Lohse, K. Biolsi, N. Walker and H.H. Rueter, "A Classification of Visual Representation," *Communications of the ACM*, 37(12):36-49, December 1994.

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

[175] J.B.A. Maintz and M.A. Viergever, "A Survey of Medical Image Registration", Medical Image Analysis, Vol. 2 No. 1, pp.1-36, 1998.

[176] B.S. Manjunath, P. Salembier, and T. Sikora, eds., *Introduction to MPEG-7: Multimedia Content Description Interface*, John Wiley & Sons, New York, 2002.

[177] J. Mao, and A.K. Jain, "Texture Classification and Segmentation Using Multiresolution Simultaneous Autoregressive Models," *Pattern Recognition*, 25(2):173-188, 1992.

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

[179] D.McG. Squire, T. Pun, "A Comparison of Human and Machine Assessments of Image Similarity for the Organization of Image Databases," in *proceedings Scandinavian conference on Image Analysis*, June 9-11, Lappeenranta, Finland, 1997.

[180] D.M. McKeown Jr., W.A. Harvey Jr., and J. McDermott, "Rule-Based Interpretation of Aerial Imagery," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 7(5), September 1985.

[181] D.M. McKeown Jr., W.A. Harvey Jr., and L.E. Wixson, "Automatic Knowledge Acquisition for Aerial Image Interpretation," *Computer Vision, Graphics, and Image Processing (CVGIP),* 46:37-81, 1989.

[182] C. Meilhac and C. Nastar, "Relevance Feedback and Category Search in Image Databases," in *proceedings IEEE International Conference on Multimedia Computing and Systems*, vol. 1:512-517, June 7-11, Florence, Italy, 1999.

[183] J. Meng and S.-F. Chang, "Tools for Compressed-Domain Video Indexing and Editing," in *proceedings of SPIE Conference on Storage and Retrieval for Image and Video Database*, vol. 2670:180-191, San Jose, February 1996.

[184] Z. Michalewicz. Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag, New York, 1992.

[185] T. Minka, and R. Picard, "Interactive Learning Using a Society of Models," *Pattern Recognition,* 30(4), 1997.

[186] T. Mitchell. *Machine Learning.* McGraw-Hill, New York, 1997.

[187] A. Mojsilovic, J. Gomes, and B. Rogowitz, "Isee: Perceptual Features for Image Library Navigation," in *proceedings of SPIE, Human Vision and Electronic Imaging*, San Jose, CA, January 2002.

[188] A. Mojsilovic and B. Rogowitz, "Capturing Image Semantics with Low-Level Descriptors," in *proceedings IEEE International Conference on Image Processing (ICIP 2001)*, Thessaloniki, Greece, October 2001.

[189] A.W. Moore and M.S. Lee, "Efficient Algorithms for Minimizing Cross Validation Error," in *proceedings of the 11th International Conference on Machine Learning*, Morgan Kaufmann, 1994.

[190] MPEG-7, MMDS Group, "MPEG-7 Multimedia Description Schemes (V6.0)," *Doc. ISO/IEC JTC1/SC29/WG11 MPEG01/N3815*, Pisa, Italy, January, 2001.

[191] MPEG-7, Requirements Group, "MPEG-7 Overview (V8.0)," *ISO/IEC JTC1/SC29/WG11*, Klangenfurt, July, 2002.

[192] MPEG Multimedia Description Scheme Group, "Text of ISO/IEC CD 15938-5 Information technology - Multimedia content description interface: Multimedia description schemes", *ISO/IEC JTC1/SC29/WG11 MPEG00/N3705*, La Baule, France, October 2000 (see also MPEG-7 website: *http://mpeg.telecomitalialab.com*)

[193] MPEG Requirements Group, "Description of MPEG-7 Content Set," *ISO/IEC JTC1/SC29/WG11 MPEG98/N2467*, Atlantic City, USA, 1998.

[194] H. Murase, and S.K. Nayar, "Visual Learning and Recognition of 3D Objects from Appearance," *International Journal of Computer Vision*, 14(1):5-24, 1995.

[195] P.M. Murphy, *UCI Repository of Machine Learning Databases- a machine-readable data repository*, Maintained at the department of Information and Computer Science, University of California, Irvine. Anonymous FTP from ics.uci.edu in the directory pub/machine-learning-databases, 1995.

[196] M. Nagao, and T. Matsuyama. *A Structural Analysis of Complex Aerial Photographs.* Plenum Press, New York, 1980.

[197] M.R. Napahde, and T.S. Huang, "A Probabilistic Framework for Semantic Video Indexing, Filtering, and Retrieval," *IEEE Transactions on Multimedia,* 3(1): 141-551, March 2001.

[198] S. Nayar and T. Poggio, editors. *Early Visual Learning.* Oxford University Press, New York, 1996.

[199] A.Y. Ng, "On Feature Selection: Learning with Exponentially many Irrelevant Features as Training Examples," in *proceedings of International Conference on Machine Learning (ICML)*, 1998.

[200] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos and G. Taubin, "The QBIC Project: Quering Images by Content Using Color, Texture, and Shape," in *proceedings* of *SPIE Storage and Retrieval for Image and Video Databases*, vol. 1908:173-187, February 1993.

[201] D.A. Norman and S.D., editors. *User Centered System Design: new perspectives on human-computer interaction*. L. Erlbaum Associates, Hillsdale, N.J., 1986.

[202] J. Novovicová, P. Pudil, and J. Kittler, "Divergence Based Feature Selection for Multimodal Class Densities," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 18(2):218-223, February 1996.

[203] V.E. Ogle and M. Stonebraker, "Chabot: Retrieval from a Relational Database of Images," *IEEE Computer Magazine*, 28(9):40-48, September 1995.

[204] Y.A. Otha, "A Region-oriented Image-Analysis System by Computer," *Ph.D. thesis*, Kyoto University, March 1980.

[205] S. Paek, and S.-F. Chang, "The case for Image Classification Systems Based on Probabilistic Reasoning," in *proceedings, IEEE International Conference on Multimedia and Expo (ICME 2000)*, New York City, July, 2000.

[206] S. Paek, C. L. Sable, V. Hatzivassiloglou, A. Jaimes, B. H. Schiffman, S.-F. Chang, K. R. McKeown, "Integration of Visual and Text based Approaches for the Content Labeling and Classification of Photographs," in *proceedings of ACM SIGIR '99 Workshop on Multimedia Indexing and Retrieval*, Berkeley, CA. August, 1999.

[207] S. Paek, A.B. Benitez and S.-F. Chang, "Self-Describing Schemes for Interoperable MPEG-7 Content Descriptions," in *proceedings of SPIE Visual Communications and Image Processing,* vol. 3653:1518-1530, San Jose, CA, January 1999.

[208] A. Pentland, R.W. Picard, and S. Sclaroff, "Photobook: Tools for Content-Based Manipulation of Image Databases," in *proceedings of SPIE Storage and Retrieval for Image and Video Databases II*, vol. 2187:34-47, 1994.

[209] E.G.M. Petrakis and C. Faloutsos, "Similarity searching in large image databases," *University of Maryland Department of Computer Science, Technical Report,* No. 3388, 1995.

[210] Photonet (*http://www.photo.net*).

[211] R.W. Picard, "Computer Learning of Subjectivity," *ACM Computing surveys*, 27(4):621-623, December 1995.

[212] R.W. Picard, "A Society of Models for Video and Image Libraries," *MIT Media Laboratory Perceptual Computing Section Technical Report,* No. 360, Cambridge, Massachusetts, 1996.

[213] J. Platt, "Auto Album: Clustering Digital Photographs Using Probabilistic Model Merging," in Proc. *IEEE CBAIVL Workshop in conj. With CVPR*, pp. 96-100, 2000.

[214] C.M. Privitera, and L.W. Stark, "Algorithms for Defining Visual Regions-of-Interest: Comparison with Eye Fixations," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22(9):970-981, September 2000.

[215] H. Purchase, "Defining Multimedia," *IEEE Multimedia Magazine*, 5(1):8-15, January-March 1998.

[216] J.R. Quinlan, "Induction of Decision Trees," *Machine Learning*, 1(1), 1986.

[217] R.K. Rajendran and S.-F. Chang, "Visual Search Tools and their Application in K-12 Education," *ADVENT project technical report*, Columbia University, May 1999.

[218] T.R. Reed and J.M. Hans Du Buf, "A Review of Recent Texture Segmentation and Feature Extraction Techniques," *Computer Vision, Graphics, and Image Processing (CVGIP): Image Understanding*, 57(3), May 1993.

[219] J.J. Rocchio Jr., "Relevance Feedback in Information Retrieval," In Gerard Slaton, editor, *The Smart Retrieval System: Experiments in Automatic Document Processing*, pages 313-323. Prentice-Hall, Englewood Cliffs, NJ, 1971.

[220] H.A. Rowley, S. Baluja, and T. Kanade, "Human Face Detection in Visual Scenes," *Carnigie Mellon University Technical Report CMU-CS-95, 158,* 1995.

[221] H.-Y. Shum and R. Szeliski. Construction of panoramic mosaics with global and local alignment. *International Journal of Computer Vision*, 36(2):101-130, February 2000.

[222] A. Rosenfeild, "Survey, Image Analysis and Computer Vision: 1992," *Computer Vision, Graphics, and Image Processing (CVGIP): Image Understanding*, 58(1):85-135, July 1993.

[223] Y. Rui, T.S. Huang, and S.-F. Chang, "Image Retrieval: Current Directions, Promising Techniques, and Open Issues," *Journal of Visual Communication and Image Representation*, No. 10:1-23, 1999.

[224] Y. Rui, T.S. Huang, M. Ortega, and S. Mehrotra, "Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval," *IEEE Transactions on Circuits and Video Technology*, 8(5):644-655, September 1998.

[225] J. Russ. *The Image Processing Handbook*. 3rd edition, CRC Press, Boca Raton, Fl, 1999.

[226] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall series in artificial intelligence, Prentice Hall, Englewood Cliffs, N.J., 1995.

[227] G. Salton. Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Addison-Wesley, MA, 1989.

[228] G. Salton and C. Buckley, "Term Weighting Approaches in Automatic Text Retrieval," *Information Processing and Management,* 25(5), 1988

[229] S.L. Salzberg, "On Comparing Classifiers: A Critique of Current Research and Methods," *Data Mining and Knowledge Discovery*, 1:1-12, 1999.

[230] S. Santini, "Explorations in Image Databases," *Ph.D. thesis*, University of California, San Diego, January 1998.

[231] S. Santini, A. Gupta, and R. Jain, "A User Interface for Emergent Semantics in Image Databases," in *proceedings of 8th IFIP Workshop on Database Semantics* (DS-8), Rotuva, New Zealand, January 1999.

[232] S. Santini, R. Jain, "Beyond Query by Example," in *proceedings of the ACM International Conference on Multimedia 98*, pages 345-350, Bristol, England, September 1998.

[233] S. Santini and R. Jain, "Gabor Space and the Development of Preattentive Similarity," in *proceedings of International Conference on Pattern Recognition (ICPR '96)*, vol. 1:40-44, Vienna, Autum 1996.

[234] T.J. Santner, and D.E. Duffy. *The Statistical Analysis of Discrete Data.* Springer-Verlag, New York, 1989.

[235] F. Schaffalitzky and A. Zisserman, "Multi-view Matching for Unordered Image Sets, or "How Do I Organize My Holiday Snaps?"," in proceedings of *7th European Conference on Computer Vision (EECCV 2002),* Vol. 1, pp. 414-431, Copenhagen, Denmark, 2002.

[236] F. Schaffalitzky and A. Zisserman, "Automated Scene Matching in Movies," in proceedings of *International Conference in Image and Video Retrieval (CIVR 2002)*, p. 186, London, UK, July 2002.

[237] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, 47(1):7-42, May 2002.

[238] D.W. Scott. Multivariate Density Estimation: Theory, Practice, and Visualization. John Wiley and Sons, New York, 1992.

[239] M. Shibata, A. Tam, C. Leung, K. Hasida, A.B. Benitez, A. Jaimes, S.-F. Chang, C. Jorgensen, and E. Oltmans, "Report of CE on Structured Textual Description", *ISO/IEC JTC1/SC29/WG11 MPEG00/M6240*, Beijin, China, July 2000.

[240] B. Shneiderman, "Meeting Human Needs with New Digital Imaging Technologies," *IEEE Multimedia,* Vol.9, No.4, pp. 8-14, Oct.-Dec., 2002.

[241] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-Based Image Retrieval at the End of the Early Years", in *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22(12):1349-1380, December 2000.

[242] J.R. Smith, *Integrated Spatial and Feature Image Systems: Retrieval, Analysis, and Compression*, Ph.D. thesis, Electrical Engineering Department, Columbia University, New York, New York, 1997.

[243] J.R. Smith, and S.-F. Chang, "Multi-stage Classification of Images from Features and Related Text," in *proceedings Fourth DELOS workshop*, Pisa, Italy, August, 1997.

[244] J.R. Smith and S.-F. Chang, "An Image and Video Search Engine for the World-Wide Web," in *proceedings of SPIE Storage & Retrieval for Image and Video Databases V*, vol. 3022:84-95, San Jose, CA, February 1997.

[245] J.R. Smith and S.-F. Chang, "Transform Features for Texture Classification and Discrimination in Large Image Databases," *IEEE International Conference on Image Processing*, vol. 3:407-411, Austin, TX, November, 1994.

[246] J.R. Smith and S.-F. Chang. "VisualSEEk: a fully automated content-based image query system," in *proceedings of the ACM Conference on Multimedia (ACM MM '96)*, pages 87-98, November, 1996.

[247] R.K. Srihari, "Automatic Indexing and Content-Based Retrieval of Captioned Images," *IEEE Computer Magazine*, 28(9):49-56, September 1995.

[248] U. Srinivasan, C. Lindley, and B. Simpson-Young, "A Multi-Model Framework for Video Information Systems," *in Database Semantics: Issues in Multimedia Systems*, pages 85-108, Kluwer Academic Publishers, January 1999.

[249] C. Stiller and J. Konrad, "Estimating Motion in Image Sequences: A Tutorial on Modeling and Computation of 2D Motion," *IEEE Signal Processing Magazine*, Vol. 16, Issue 4, pp. 70-91, July 1999.

[250] Y. Sun, H.J. Zhang, L. Zhang, and M. Li, "MyPhotos: A System for Home Photo Management and Processing," in proc. ACM Multimedia 2002, Juan Les Pines, France, December 2002.

[251] M.J. Swain and D.H. Ballard, "Color Indexing," *International Journal of Computer Vision,* 7(1):11-2, 1991.

[252] D.L. Swets and J.J. Weng, "Efficient Content-Based Image Retrieval using Automatic Feature Selection," in *proceedings of International Conference on Computer Vision (ICCV '95)*, Coral Gables, Florida, November 1995.

[253] T.F. Syeda-Mahmood, "Data and Model-Driven Selection Using Color Regions," *International Journal of Computer Vision*, 21(1/2):9-36, 1997.

[254] M. Szummer and R.W. Picard, "Indoor-Outdoor Image Classification," in *proceedings of IEEE International Workshop on Content-based Access of Image and Video Databases*, pages 42-51, Bombay, India, 1998.

[255] H. Tamura, S. Mori, and T. Yamawaki, "Textural Features Corresponding to Visual Perception," *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-8:6, June 1978.

[256] Tonomura, Y., A. Akutsu, Y. Taniguchi and G. Suzuki, "Structured Video Computing," *IEEE Multimedia Magazine*, 1(3): 34-43, Fall 1994.

[257] TiVO Official Homepage, *http://www.tivo.com*

[258] P.H.S. Torr, "A Structure and Motion Toolkit in Matlab," *Microsoft Research Technical Report No. MSR-TR-2002-56*, Redmond, WA, May 2002.

[259] P.H.S. Torr, "Model Selection for Structure and Motion Recovery from Multiple Images," *Microsoft Technical Report No. MSR-TR-99-16*, Redmond, WA, March 1999.

[260] P.H.S. Torr, "The Development and Comparison of Robust Methods for Estimating the Fundamental Matrix," *International Journal of Computer Vision*, Vol. 34, No. 3, pp. 271-300, 1997.

[261] A. Triesman, "Preattentive Processing in Vision," *Computer Vision, Graphics, and Image Processing (CGVIP),* 31:156-177, 1985.

[262] M. Turk, and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, 3(1), 1991.

[263] TV Anytime Forum, *http://www.tv-anytime.org*

[264] A. Vailaya, M. Figueiredo, A. Jain, and H.J. Zhang, "Image Classification for Content-Based Indexing," *IEEE Transactions on Image Processing*, 10(1): 117-130, January, 2001.

[265] A. Vailaya, A. Jain and H.J. Zhang, "On Image Classification: City vs. Landscape," in *proceedings of IEEE Workshop on Content-Based Access of Image and Video Libraries*, pages 3-8, Santa Barbara, California, June 21, 1998.

[266] A. Vailaya, M. Figueiredo, A. Jain, and H.J. Zhang, "Content-Based Hierarchical Classification of Vacation Images," in *proceedings of IEEE Multimedia Computing and Systems*, vol. 1:518-523, Florence, Italy, June 1999.

[267] A. Vailaya, M. Figueiredo, A. Jain, and H.J. Zhang, "Automatic Orientation Detection," in *proceedings of International Conference on Image Processing (ICIP '99)*, vol. 2:600-604, Kobe, Japan, 1999.

[268] N. Wacholder, Y. Ravin, and M. Choi, "Disambiguation of Proper Names in Text," in *proceedings of 5th Conference on Applied Natural Language Processing*, Washington D.C., April 1997.

[269] H. Wang and S.-F. Chang, "A Highly Efficient System for Automatic Face Region Detection in MPEG Video," *IEEE Transactions on Circuits and Systems for Video Technology, Special issue on Multimedia Systems and Technologies*, 7(4):615-628, August 1997.

[270] WEKA Machine Learning Software (available at: *http://www.cs.waikato.ac.nz/ml/weka/*)

[271] I.H.Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.* Morgan Kaufman Publishers, New York, 1999.

[272] M. M. Yeung, B.-L. Yeo and B. Liu, "Segmentation of Video by Clustering and Graph Analysis", *Computer Vision and Image Understanding*, V. 71, No. 1, July 1998.

[273] A. Yoshitaka, and T. Ichikawa, "A Survey on Content-Based Retrieval for Multimedia Databases," *IEEE Transactions on Knowledge and Data Engineering*, 11(1):81-93, January/February 1999.

[274] D. Zhong and S.-F. Chang, "Video Object Model and Segmentation for Content-Based Video Indexing," in proceedings of IEEE International Conference on Circuits and Systems, Special session on Networked Multimedia Technology & Applications, vol. 2:1492-1495, Hong Kong, June, 1997.

[275] D. Zhong and S.-F. Chang, "AMOS: An Active System for Mpeg-4 Video Object Segmentation," in proceedings of International Conference on

Image Processing (ICIP '98), vol. 2:647-651, Chicago, Illinois, USA, October 4-7, 1998.

[276] D. Zhong and S.-F. Chang, “Structure Analysis of Sports Video UsingDomain Models”, IEEE International Conference on Multimedia and Expo(ICME 2001), Tokyo, Japan, 2001.

Eye Tracking

[277] Becker, W. "Metrics," In *The Neurobiology of Saccadic Eye Movements*. Goldburg, M.E. & Wurtz, R.H., Eds. Elsevier Science Publishers, 1989.

[278] Buswell, G.T., *How People Look at Pictures*, University of Chicago Press, Chicago, 1920.

[279] Collewijn, H., Steinman, R.M., Erkelens, C.J., Pizlo, Z., & van der Steen, J., "Effect of freeing the head on eye movement characteristics during three-dimensional shifts of gaze and tracking," Chapter 64 in *The Head-Neck Sensory Motor System*, Berthoz, A., Graf, W., & Vidal, P.P., Eds. Oxford University Press, 1992.

[280] Cornsweet; Tom N. ,Crane; Hewitt D., *US Patent US3724932*, 1973.

[281] Epelboim J., Steinman R.M., Kowler E., Pizlo Z., Erkelens C.J., Collewijn H., "Gaze-shift dynamics in two kinds of sequential looking tasks", *Vision Res.* 37: 2597-2607, 1997.

[282] Gaarder, K. R., *Eye Movements, Vision, and Behavior*, John Wiley & Sons, New York, 1975.

[283] Gould, J. D., "Looking at Pictures", in *Eye Movements and Psychological Processes*, edited by R. A. Monty and J. W. Senders, John Wiley & Sons, New York, 1976.

[284] Guedry FE, Benson AJ., "Tracking performance during sinusoidal stimulation of the vertical and horizontal semicircular canals," In: Recent Advances in Aerospace Medicine, Busby, D E (Ed.). D. Reidel Publ. Co., Dordrecht, Netherlands, 1970.

[285] Land M.F., Furneaux, S. *The knowledge base of the oculomotor system.* Phil Trans R Soc Lond B 352: 1231-1239, 1997.

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

[287] Noton, D. and Stark, L.W. , “Scanpaths in Saccadic Eye Movements while Viewing and Recognizing Patterns,” *Vision Research* 11 (9), 929-42, 1971

[288] Pelz, J.B., Canosa, R., Babcock, J., Kucharczyk, D., Silver, A., and Konno, D., "Portable Eyetracking: A Study of Natural Eye Movements," in *proceedings of SPIE, Human Vision and Electronic Imaging* , San Jose, CA, 2000.

[289] Privitera C.M., and Stark, L.W., "Evaluating Image Processing Algorithms that Predict Regions of Interest", in *Pattern Recognition Letters 19*, pp. 1037-1043, 1998.

[290] Privitera, C.M., and Stark, L.W., “Algorithms for Defining Visual Regions-of-Interest: Comparison with Eye Fixations,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22(9):970-981, Sept. 2000.

[291] Robinson, D.A., "A method for measuring eye movements using a scleral search coil in a magnetic field," *IEEE Trans Bio-Med Electron,* BME-10, 137-145, 1963.

[292] Rybak, I.A., Gusakova, V.I, Golovan, A.V., Podladchikova, L.N., and Shevtsova, N.A., "A model of attention-guided visual perception and recognition," *Vision Research.* 38:2387-2400, 1998.

[293] Schill K., Umkehrer E., Beinlich S., Zetzsche C, Deubel H, Pöppel E., "A hybrid system for scene analysis with saccadic eye movements: learning of feature relations", *European Conf. on Visual Perception*, Oxford, England, 1998.

[294] Solso, R.L., *Cognition and the visual arts*, MIT Press, Cambridge, Mass., 1994.

[295] L. Wenyin, et al., “MiAlbum-A System for Home Photo Management Using the Semi-Automatic Image Annotation Approach,” proc. *ACM Multimedia*, Los Angeles, CA, 2000.

[296] Y. Yagawa, et al., “The Digital Album: A Personal File-tainment System”, Proc. *3rd International Conference on Multimedia Computing Systems*, pp. 433-439, 1996.

[297] Yarbus, A.F. *Eye Movements and Vision*, New York, Plenum Press. 1967.