ORGANISING AND STRUCTURING A VISUAL DIARY USING VISUAL INTEREST POINT DETECTORS

by

Michael Blighe, B.Sc. (Hons), M.Sc.

Submitted in partial fulfilment of the requirements

for the Degree of Doctor of Philosophy

Dublin City University

School of Electronic Engineering

Supervisor: Prof. Noel E. O'Connor

December, 2008

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: _____
Michael Blighe (Candidate)

ID: _____

Date: _____

Acknowledgements

I wish to extend my heartfelt gratitude to the many people who have enabled and supported the work behind this thesis. Special thanks go to my supervisor Prof. Noel E. O'Connor for valuable guidance and expertise as well as his constant support, advice and encouragement. Similar thanks are due to Professor Alan Smeaton and to my colleagues in the Centre for Digital Video Processing for their advice, support and suggestions at different stages of this work. Finally, this work would not have been possible without the financial support provided by the European Commission under contract FP6-027026 (K-Space), the aceMedia Project under contract FP6-001765, and Microsoft Research and Science Foundation Ireland under grant number 03/IN.3/I361.

On a personal note, I would like to thank my parents, friends and family, for unquestioningly supporting and encouraging me for as long as I can remember, in whatever I chose to do. Finally, to Neasa, for helping maintain my relative sanity, especially in the final months of this work.

Abstract

As wearable cameras become more popular, researchers are increasingly focusing on novel applications to manage the large volume of data these devices produce. One such application is the construction of a Visual Diary from an individual's photographs. Microsoft's SenseCam, a device designed to passively record a Visual Diary and cover a typical day of the user wearing the camera, is an example of one such device. The vast quantity of images generated by these devices means that the management and organisation of these collections is not a trivial matter. We believe wearable cameras, such as SenseCam, will become more popular in the future and the management of the volume of data generated by these devices is a key issue.

Although there is a significant volume of work in the literature in the object detection & recognition and scene classification fields, there is little work in the area of setting detection. Furthermore, few authors have examined the issues involved in analysing extremely large image collections (like a Visual Diary) gathered over a long period of time. An algorithm developed for setting detection should be capable of clustering images captured at the same real world locations (e.g. in the dining room at home, in front of the computer in the office, in the park, etc.). This requires the selection and implementation of suitable methods to identify visually similar backgrounds in images using their visual features. We present a number of approaches to setting detection based on the extraction of visual interest point detectors from the images. We also analyse the performance of two of the most popular descriptors - Scale Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF). We present an implementation of a Visual Diary application and evaluate its performance via a series of user experiments. Finally, we also outline some techniques to allow the Visual Diary to automatically detect new settings, to scale as the image collection continues to grow substantially over time, and to allow the user to generate a personalised summary of their data.

Table of Contents

List of Figures

List of Tables

List of Peer-Reviewed Publications

- C. O'Conaire, **M. Blighe** and N.E. O'Connor. "SenseCam Image Localisation using Hierarchical SURF Trees" in *15th International Multimedia Modeling Conference (MMM)*, Sophia-Antipolis, France, 7-9 January 2009.

- A. Doherty, C. O'Conaire, **M. Blighe**, A.F. Smeaton and N.E. O'Connor. "Combining Image Descriptors to Effectively Retrieve Events from Visual Lifelogs" in *ACM International Conference on Multimedia Information Retrieval*, Vancouver, Canada, 30-31 October 2008.

- H. Lee, A.F. Smeaton, N.E. O'Connor, G. Jones, **M. Blighe**, D. Byrne, A. Doherty and C. Gurrin. "Constructing a SenseCam Visual Diary as a Media Process" in *Multimedia Systems Journal, Special Issue on Canonical Processes of Media Production*, 2008.

- **M. Blighe**, A. Doherty, A.F. Smeaton and N.E. O'Connor. "Keyframe Detection in Visual Lifelogs" in *1^{st} International Conference on Pervasive Technologies Related to Assistive Environments*, Athens, Greece, 15-19 July 2008.

- **M. Blighe** and N.E. O'Connor. "MyPlaces: Detecting Important Settings in a Visual Diary" in *ACM International Conference on Image and Video Retrieval (CIVR)*, Niagara Falls, Canada, 7-9 July 2008.

- **M. Blighe**, S. Sav, H. Lee and N.E. O'Connor. "Mo Mhúsaem Fíorúil: A Web-based Search and Information Service for Museum Visitors" in *The International Conference on Image Analysis and Recognition*, Povoa de Varzim, Portugal, 25-27 June 2008.

- **M. Blighe** and N.E. O'Connor. "Identifying Different Settings in a Visual Diary" in *The 9^{th} International Workshop on Image Analysis for Multimedia Interactive Services*, Klagenfurt, Austria, 19-21 May 2008.

- **M. Blighe**, H. Le Borgne, N.E. O'Connor, A.F. Smeaton and G. Jones. "Exploiting Context Information to aid Landmark Detection in SenseCam Images" in *The 2^{nd} International Workshop on Exploiting Context Histories in Smart Environments - Infrastructures and Design, 8^{th} International Conference of Ubiquitous Computing (UBICOMP)*, Orange County, CA, USA, 17-21 September 2006.

Other Relevant Publications

- **M. Blighe** and N.E. O'Connor. "Object Classification and Image Matching with SIFT" in *Irish Graduate Student Symposium on Vision, Graphics and Visualisation*, Dublin, Ireland, 5 June 2008.

- **M. Blighe**, H. Le Borgne and N.E. O'Connor. "Image Metadata Estimation using Independent Component Analysis and Regression" in *The 7^{th} International Workshop on Image Analysis for Multimedia Interactive Services*, Incheon, Korea, 19-21 April 2006.

- N.E. O'Connor, E. Cooke, H. Le Borgne, **M. Blighe** and T. Adamek. "The AceToolbox: Low-Level Audiovisual Feature Extraction for Retrieval and Classification" in *The 2^{nd} IEE European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies*, London, U.K., 30 November-1 December 2005.

CHAPTER 1

Introduction

1.1 Motivation

Many people keep a journal in order to memorise their daily life. Very often, the process of writing a diary is not simply a recounting of the day's events. Rather, it involves the recording of the emotions and feelings of the individual at that particular place and time. The explosion of online blogging sites can be viewed as an evolution of the diary in the Internet age. A detailed discussion of the reasons why people write diaries and, in particular, why they would be willing to publish personal details of their lives online is beyond the scope of this work. However, we can assume that diaries help people recall what they did and how they were feeling at a particular place and time. Essentially, writing a diary requires that the writer remembers daily events. It is not easy, however, to remember all the events in the day, so other sources of information may be used to trigger memory recall.

The growing ubiquity of media capture devices means that it is now possible to augment the traditional text-based diary with other content such as images and video clips [1]. This is attractive since the inclusion of such content can potentially aid us in reliving important events, more so than is possible with plain text. A parallel can be drawn to the way we create and arrange photograph albums to help us remember a family holiday or wedding, for example. The proliferation of digital cameras and camera-phones means that taking pictures has never been easier, fueling this growing trend of multi-media lifelogging. Providing tools to help automate content organisation and management is thus increasingly important in order to help users tame the inevitable information overload. However, the development of methods for managing digital photos (and video) has not kept pace with acquisition technology, thereby severely degrading the practical usefulness of

visual diaries that rely on these photos.

Significant research effort is currently being invested in the capture and retrieval of multimodal lifelogs in order to automatically generate a record of a user's daily life [2] [3]. Much of the work focuses on using context and content information in order to infer details about one's daily activities [4]. Context information is usually generated using location-based sensing from a mobile phone, GPS device, or other similar sources. Content information is usually derived from the analysis of passively captured audiovisual data, most often in the form of video or digital photos. Using photos, for example, one can construct a Visual Diary of an individual's life. For a single day, this might consist of a sequence of images providing a visual summary of the most important aspects of the day. The underlying challenge is to be able to manage, organise, and search large volumes of photos to judiciously select and present representative samples in a visually coherent manner. Within this broad challenge, a key objective is to be able to identify these representative samples in the first place – they typically need to be selected from thousands of images representing an individual's day (and ultimately from millions over a lifetime) and they should correspond to images that are somehow important to the owner.

1.2 Passive Image Capture

Many researchers have started work on developing passive capture devices - cameras which automatically take pictures without any user intervention. Gemmell et al. describe their work on the SenseCam, the device used in our work [1]. We use version 2.3 of the SenseCam shown in Figure 1.1 (as well as a schematic in Figure 1.2). The SenseCam prototype is the size of a pager and is attached via a neck strap, or clip, to the front of the user's body as shown in Figure 1.3. It is designed to take photos automatically, without user intervention, whilst it is being worn. Photos are triggered by sensor data and/or time interval. Sensor data is recorded every second and pictures are taken approximately every fifty seconds, unless triggered by the sensors before that time has elapsed. The sensors include: a passive infra-red detector (similar to that used in home alarm systems) which can detect living beings directly in front of the individual wearing the camera; an accelerometer which captures data in the X, Y & Z directions; a digital light sensor; and a temperature sensor. In a typical day, the SenseCam will capture anything between 2,000 and 3,000 photos. To be truly passive, the user must not be worried about pointing the camera in a precise direction. As the SenseCam is worn on the body, and the user does not use a viewfinder, the aim

can be unpredictable. A normal lens has too narrow a field of view, yielding many photos that miss the intended target. SenseCam, thus, uses a wide-angle (fish-eye) lens to provide up to 180 degrees of view. By way of comparison, the eye typically has 95 degrees of view. The advantage of a very wide-angle lens for the SenseCam is that most or all of the forward view is captured with a large depth of field. With the wide-angle and large depth of field it is rare that the camera misses what the user is seeing.

Examples of the types of images captured by SenseCam, in different settings, can be seen in Figure 1.4. SenseCam also has a manual trigger button which allows the user to take pictures in a more traditional manner or, alternatively, the user can intentionally capture a photo by simply moving a hand across the front of the camera. The shadow creates a light change and thus an image is captured.

Figure 1.1: Microsoft SenseCam

Various other passive capture systems have been developed as well. The *Casual Photography* project from HP Research Labs is very similar in nature to the SenseCam [6]. A small, wearable, camera passively records video of everything the user sees throughout their day. StartleCam is a wearable video camera, computer, and sensing system which also passively captures images de-

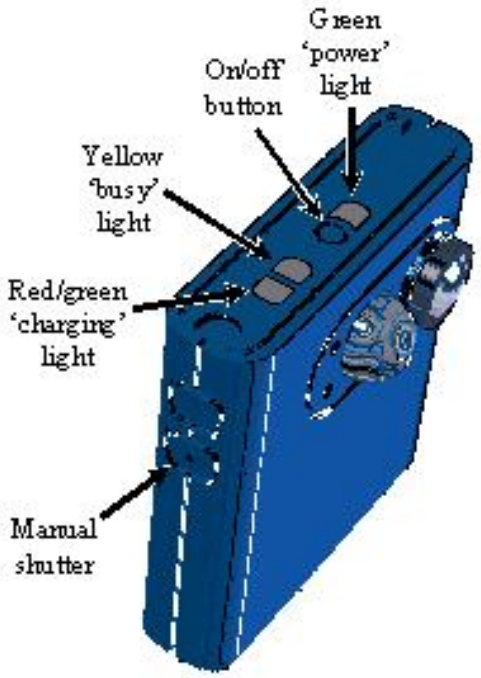


Figure 1.2: Schematic of Microsoft SenseCam [5]

pending on certain events detected by the sensors on the device [7]. The Campaignr project [8] is a software framework for mobile phones that enables owners of smartphones (specifically Symbian Series 60 3^{rd} edition phones) to participate in data gathering campaigns including automatic image capture. A similar project is WayMarkr [9], a system which uses a mobile device's camera to take continuous photographs from the vantage point of the wearer. The ubiquity of mobile devices helps make both WayMarkr and Campaignr unobtrusive, and they are perhaps the first image based devices that are widely available to the general public which enable users to fulfil the Memex vision of storing a lifetime's worth of photos [10]. Hori et al. [11] developed a wearable system that continuously captures video, along with sensors that include GPS, gyroscope, accelerometer, and a brain wave sensor that has produced promising results for indicating interesting scenes (see Figure 1.5). Similarly, Clarkson et al. [12] attempted to recognise a person's situation using only a wearable camera and a microphone.

The main advantage of passive capture is that it allows people to record their experiences without having to operate recording equipment, and without having to give recording a conscious thought. This results in increased coverage of, and improved participation in, the event itself. However, the passive capture of photos presents new problems, particularly, how to manage and organise the massively increased volume of images captured. We argue in this thesis that traditional systems for content-based image retrieval (which typically use global image features such

Figure 1.3: Position SenseCam is worn

as colour or texture) are not adequate for this task. We discuss these problems in more detail in Sections 2.4.3 and 2.7.

1.3 Thesis Objectives

The first objective of this thesis is to outline the current state of the art in lifelogging. This review discusses the most recent advances in the lifelogging area. We also review the issues raised by the lifelogging process. This thesis does not attempt to cover all aspects of lifelogging, nor does it represent a detailed literature review of the vast amount of work published in this field. Its purpose is to frame the remainder of the discussion and to highlight our own particular area of interest within this large field of research. Having presented a broad overview of lifelogging, we subsequently restrict the discussion to the area of content management. The techniques used to manage the content generated during the lifelogging process are discussed, along with the problems lifelogging raises when designing algorithms to manage this data.

The second objective is to investigate a new approach to help solve the content management problem as it pertains to lifelogging and a Visual Diary application in particular. To this end, *Setting Detection* (i.e. detecting images in the collection taken at specific locations) is targeted as a very useful enabling technology and a taxonomy of approaches to Setting Detection is presented. A brief discussion of the issues raised by Setting Detection is also undertaken. One of the main goals of this discussion is to critically evaluate the existing approaches in order to determine the

Figure 1.4: Sample SenseCam Images

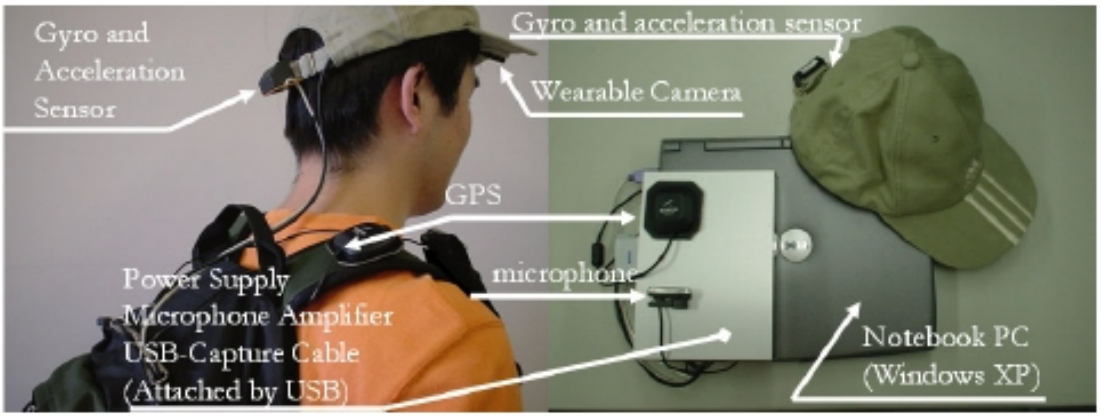


Figure 1.5: Wearable capture system utilising a video camera and numerous sensors [11]

most appropriate techniques necessary to perform Setting Detection in a Visual Diary.

The third objective is to present a number of approaches to Setting Detection in detail and to explore the robustness of these approaches under a variety of scenarios. In each scenario, each of the main parameters incorporated within the proposed techniques is rigorously examined. User studies are undertaken to validate the proposed technique and an application is developed to facilitate this evaluation.

The fourth objective is to analyse the results obtained in order to facilitate the development of techniques which will allow the Visual Diary to dynamically evolve as it grows over time. In particular, an approach is presented which enables the automatic detection of new settings. The approach is discussed in detail and results are presented in order to validate the proposed technique. In addition, feedback from the user studies is analysed in order to improve the application developed.

The final objective of this thesis is to indicate directions for further research, namely to consider possibilities for further improvement of the proposed solutions and to discuss the prospects of using them as a basis for a variety of end user applications.

1.4 Main Research Contributions

The main contributions of this research can be summarised as follows:

- an approach to setting detection in visual lifelogs is developed that facilitates the management and organisation of images generated in the construction of a Visual Diary.
- a Visual Diary application is developed which allows users to easily manage their image

collections.

- the utility of the developed approach to setting detection is validated through user trials with the developed application.

- the characteristics of settings are analysed and an approach to the automatic detection of new settings in a Visual Diary is developed.

Each of these elements can be viewed as a major contribution of the research programme documented in this thesis. However, each consists of a number of additional contributions:

- The contributions in the first case include: (a) a comprehensive evaluation of the most appropriate techniques available to perform setting detection; (b) the determination of the optimal parameters to use in the detection of settings; (c) a comparison between the major components of the system (i.e. SIFT or SURF, K-means or X-means, etc.).

- In the second case this includes: (a) a novel web-based interface to facilitate the management and organisation of a Visual Diary; (b) a discussion of the most appropriate principles in application design used to develop a Visual Diary application, and hence a contribution to application design principles in this area.

- In the third case this includes: (a) a contribution in the area of experimental evaluation is made. This contribution constitutes a technique for evaluating the accuracy and overall utility of a settings based Visual Diary application.

- In the fourth case this includes: (a) an insight into the way different users conduct their daily lives. This contribution constitutes an analysis of the settings detected in order to gain an insight into each individuals life; (b) a technique is developed to facilitate the growth of the Visual Diary as more images are added to it over time. This contribution facilitates the automatic adaptation and growth of a Visual Diary application.

In addition to these research contributions in the areas of image analysis, application design, and experimental evaluation, a further contribution is made in the area of image analysis and application design. This contribution consists of variants of the proposed algorithm and the applications which can be built using minor variations of the proposed techniques. In particular, minor variations in the application design allow other application scenarios to be considered, and these are also described.

1.5 Document Structure

A review of the current literature on lifelogging and setting detection is provided in Chapters 2 and 3 respectively. In Chapter 2, existing approaches to lifelogging, as well as the issues raised by it, are first discussed. This is followed by a discussion on the content management issues associated with lifelogging. Various techniques for managing image and video content are reviewed and setting detection is proposed as a potential solution. In Chapter 3, we similarly review existing approaches and issues relating to setting detection, before focusing on the techniques most appropriate for use in a Visual Diary.

In Chapter 4, an overview of interest point detection techniques is presented and the first contribution of this thesis is made. In addition, three approaches to setting detection (one baseline technique) using interest point detection algorithms are outlined. The experiments performed to evaluate these algorithms are presented and a discussion of results follows.

In Chapter 5, we discuss the development of a Visual Diary application and outline some user experiments performed to determine if the algorithms developed in Chapter 4 are useful in this application scenario. This chapter represents the second and third contributions of the thesis.

In Chapter 6, we analyse the settings detected in order to determine their characteristics and to gain insights into the users lives. We also present an approach to the automatic detection of new settings in a Visual Diary, as well as presenting a technique to facilitate the detection of new settings from a huge volume of lifelog data. This chapter represents the fourth contribution of the thesis.

Finally, the contributions of this thesis are summarised and future work is outlined in Chapter 7. This future work includes improvements to the proposed setting detection algorithms, as well as further improvements to the additional applications presented in Appendix D, the algorithms outlined in Chapter 4, the application discussed in Chapter 5, and the algorithms discussed in Chapter 6.

CHAPTER 2

Lifelogging: An Overview

2.1 Introduction

On the morning of her 22^{nd} birthday, Ellie Harrison ate half a slice of toast covered with Snickers spread. Sixteen minutes later, she ate some banana cake. But before eating both, she took a picture of them. For the following year, following a strict set of guidelines, she took a photo of every single thing she ate. She called this challenge the *Eat22* project. A total of 1,640 photgraphs were taken over the year and all are available on her website [13]. A sample of these photos can be seen in Figure 2.1. Similar projects which Ellie has been involved in include *Gold Card Adventures*, logging all of her public transport journeys for a year, and the *Tea Blog*, which is updated every time Ellie drinks a cup of tea, or other hot drink. Each blog entry contains the thought which is most on her mind at that time. Other digital artists are also experiencing some of the realities of living in a lifelogged world. The artist Stephanie comprehensively recorded all her purchases in 2001. A total of $239,620.80, including a new home, was spent on 2,587 different items and services. All are catalogued online with descriptions, photographs, dates, times, locations, and amounts [14].

Many similar projects, documenting the apparently mundane aspects of every day life, exist online. Some would consider the creators of these archives to be obsessives in the medical sense of the word. However, there is a long tradition in conceptual-art genres of exhaustively chronicling the banal things we do, every day, or obsessively documenting the unremarkable aspects of the world. In the early 1960's, Andy Warhol began his *Time Capsules* project [15]. Spanning a thirty year period, this collection consists of 610 standard sized cardboard boxes filled with random items that accumulated around his desk. The archives also include a collection of over 4,000 audio

Figure 2.1: A portion of Ellie Harrison's Eat22 dietary lifelog [13]

tapes featuring interviews and conversations between Warhol and his friends and associates, as well as a vast array of other documentation and photographs. Long before the term lifelogging was coined, Andy Warhol had integrated the archiving process into the fabric of his every day life. Twenty-first century technology is now allowing us to take these concepts to a new level and to automate the data collection process.

2.2 Lifelogging

The photographic archiving projects, described in Section 2.1 above, are generally restricted to capturing the essence of one particular theme and are governed by a strict set of rules. Lifelogs, on the other hand, are much more all-encompassing. The term *lifelog* refers to a comprehensive archive of all aspects of an individual's life. Created via pervasive computing technologies, the goal of lifelogging is to record and archive all of the information pertaining to a person's life. This includes all text, all visual information, all audio, all media activity, as well as all biological data. A lifelog should provide a detailed record of the past that includes every action, every event, every conversation, and every experience of the user, thus enabling future access and facilitating rememberance. Advances in hardware (processing power, disk capacity, digital cameras, and other sensors), coupled with a clear interest among many people in storing large amounts of personal data on their computers, are fueling the growing interest in this area.

The first reference to what we now call lifelogging was probably in 1945 when Vannevar

Bush's seminal article *As We May Think* was published [10]. In this visionary work, he introduced the concept of the *Memex*, "a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility". Bush did not foresee the exact technology required to accomplish this, but he correctly foresaw two of the fundamental features: annotation and links. His ideas were first realised digitally by Douglas Engelbart in the 1960's [16], and later by Ted Nelson who kept personal recordings of every conversation he had, no matter where or of what importance (although he admits to having never revisited these archives) [17].

By 2008, a number of individuals and research groups have experimented with lifelogs. Steve Mann's Eyetap project goes beyond the mere use of a wearable camera [18]. Mann, a self-described cyborg who's been broadcasting his life onto the Internet in one form or another for more than 20 years, has replaced a large obtrusive helmet camera that he wore in the 1980's with a device that looks like an ordinary pair of sunglasses. Reflections on his many years of experience of lifelogging, or cyborglogging as he refers to it, provide some interesting insights into the social, artistic, and legal issues surrounding this area. He describes people's hostile reactions to the strange physical appearance of the device he wore (see Figure 2.2), the legal challenges he took to allow self-modification of appearance, and the physical abuse he suffered at the hands of security guards in a museum [19]. Some of these issues are discussed in more detail in Section 2.3.

Figure 2.2: The evolution of Steve Mann's Eyetap device. From wearable computers in the 1980's and early 1990's to what look like ordinary sunglasses [18]

Perhaps the most well known lifelogging project is MyLifeBits at Microsoft Research [20].

MyLifeBits is an effort to store all of one's digital media over a lifetime. It extends Bush's vision to handle audio and video, to perform database style queries, and to allow multiple visualisations in the user interface. For the past number of years, Gordon Bell has been documenting every aspect of his work life. He wears a SenseCam around his neck to capture images of everything he sees and does during the day. Every keystroke on his computer, every email, every conversation, every website he visits, is recorded and archived. Any data that can be digitised and recorded is logged and placed into the system. To achieve this, he makes use of an array of desktop devices and wearable sensors (see Figure 2.3), requiring self-conscious acts of collection and storage.

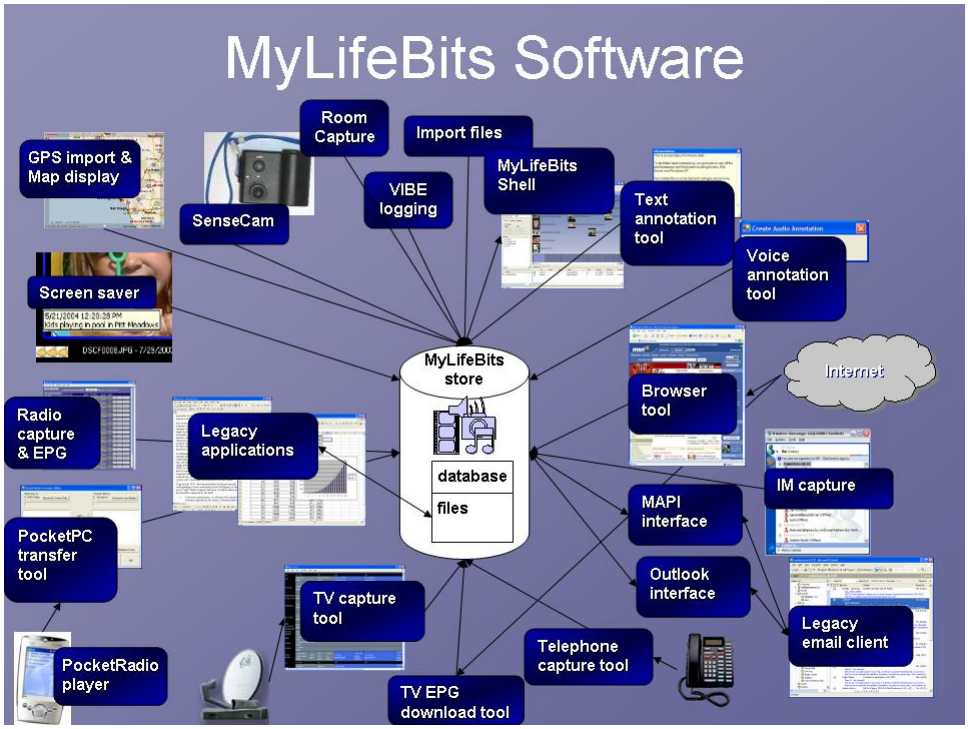


Figure 2.3: The MyLifeBits store, capture, and display tools [3]

Besides the two lifelogging projects mentioned, there are a growing number of related projects. Ellis et al. have investigated using audio to generate an audio record of one's life [21]. Nokia Lifeblog is a multimedia diary that automatically collects all the photos, videos, and sound clips that the user creates on their mobile phone. It organises all the contents in a timeline and renders the diary searchable via its contents and via automatically and manually created metadata, including time, location, tags, descriptions, filenames, sender, and recipient information [22]. *Total Recall* is a lifelog research project of the Internet Multimedia Lab at the University of Southern California [23]. The system records an individual perspective of the world using personal sensors such as a microphone in a pair of glasses or a camera in a necklace. Cyber-goggles (see Figure

2.4) is another project using a pair of glasses with built in camera, display screen, and object recognition system [24]. Examples of other projects, briefly discussed in Section 1.2, include [6, 7, 11, 12]. Indeed, such is the interest and importance of this area that the United Kingdom's Engineering and Physical Sciences Research Council has designated Memories for Life a "grand challenge" [25].

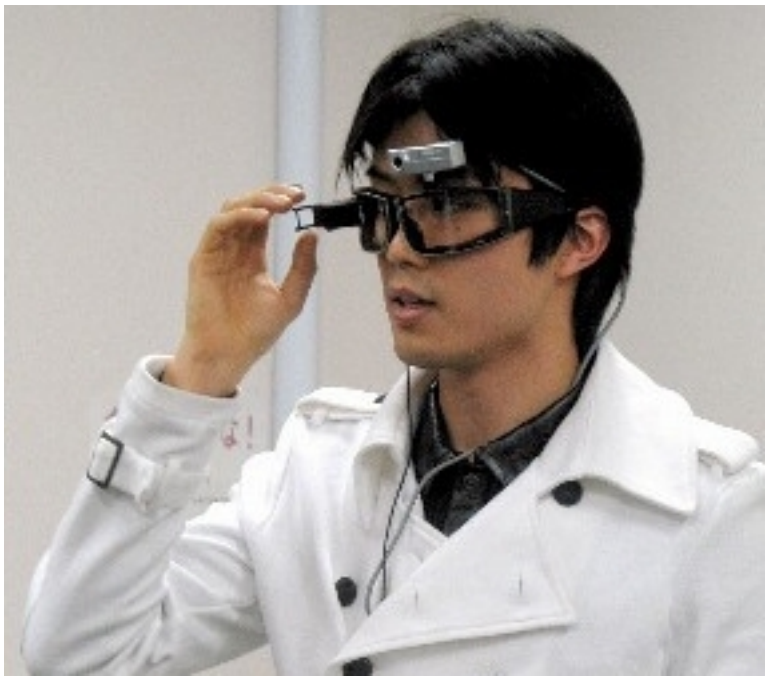


Figure 2.4: The Cyber-goggles image capture device [24]

So, how close are we to realising this "grand challenge"? Products, such as Nokia's Lifeblog, already exist and are on the market. Other commercial systems include Deja View's Camwear [26], that supports information capture via a head-worn video camera, and devices from Bodymedia, which continuously monitor and record physiological information (see Figure 2.5) [27]. However, the technology that will enable people to fully and continuously document their entire lives is still in the research and development phase. Meanwhile, plentiful storage encourages everyone to keep more and more of their memories in digital form. Gordon Bell estimates that sixty years of human experience constitutes one terabyte of data [3]. That amount of data can be stored on a €300 hard drive today, but tomorrow will be storable on cheap mobile phones, as cheap as Andy Warhol's cardboard boxes. Therefore, the lifelogging devices of the future should be relatively inexpensive and users will eventually be able to keep every document they read, every picture they view, all the audio they hear, and a good portion of what they see.

Figure 2.5: BodyMedia SenseWear R Pro 2 [27]

2.2.1 The Appeal of the Lifelog

The rationale for lifelogging centres on the idea of a "memory for life", but is there any value to it? Lifeloggers will point to the existence of hard drives full of personal data, photo albums full of photos, collections of home movies, old letters and christmas cards, and bookshelves and filing cabinets full of important books and documents. The existence of these items seems to indicate that the vast majority of us hoard different items for a variety of reasons [28]. At a very basic level, a lifelog could help reduce some of this physical clutter. When people have lost everything, perhaps due to natural disaster, the one thing they often miss most are photo albums or other items of a sentimental nature. A lifelog can ensure their continued existence in perpetuity. A lifelog will also provide a digital memory of people you met, conversations you had, places you visited, and events in which you participated. This memory would be searchable, retrievable, and shareable. It also provides a complete archive of one's work and play, an analysis of which could assist productivity, creativity, physical fitness, and overall well-being.

Lifelogging can also unleash hidden talents within all of us. Many of the individuals discussed in Section 2.1 would not have described themselves as artists, but their work is considered as art. Other examples include the emergence of journalists, entertainers, and communicators, using other forms of technology such as Internet blogs or photo and video sharing websites [29, 30, 31]. Lifelogging offers the potential to bring all of this information together and novel products and services may emerge from this which improve the quality of life. Lifelogging might also encourage introspection and self-knowledge by providing a mechanism of organising, shaping, and under-

standing one's own life. The capacity to share lifelogs could increase intimacy, understanding, and accountability in personal relationships. Inheriting the lifelog of a deceased parent, spouse, or child, could help preserve family history and ease the pain of loss.

On a physiological level, a lifelog can record vital measurements such as body temperature, heart rate, blood pressure, and the presence or absence of biochemicals. This data could serve as a warning system and also as a personal basis upon which to diagnose illness and to prescribe medicines. It could be used to enhance the recalling of frail memories, particularly in aging populations where there might be significant memory loss [32]. It could also monitor stress levels, fitness, and dietary concerns. Systems already exist to gather and monitor some of this data in isolation. The lifelog vision seamlessly brings it all together.

A record of all our personal experiences, designed solely for private consumption, appears innocent enough. However, the extreme form of lifelogging is still viewed by many as quite radical. This involves wearing microscopic cameras and microphones that record everything you see and hear, computers that archive every action you take, devices to track your location, and dozens of sensors to monitor your vital signs. The technical aspects of recording this information are quite feasible and emminently within reach at the moment. However, there are many technological, legal, and social issues, which must be solved to make lifetime recording valuable and practical in real scenarios. It is only when these issues have been resolved that lifelogging will gain large scale public acceptance.

2.3 Issues Raised by Lifelogging

In 2003, the Defence Advanced Research Projects Agency (DARPA) launched their own lifelog project. The lifelog technology DARPA conceived "can be used as a stand-alone system to serve as a powerful automated multimedia diary and scrapbook" [33]. Moreover, "by using a search engine interface", the user of the lifelog that DARPA hoped to create, could "easily retrieve a specific thread of past transactions, or recall an experience from a few seconds ago or from many years earlier in as much detail as is desired, including imagery, audio, or video replay of the event". The aims of the project were to gather in a single place everything an individual says, sees, or does. As a concept, this is not dissimilar to the MyLifeBits project. However, DARPA's lifelog project was cancelled. Officials cited a "change in priorities" for the cancellation, although most people believe the privacy and ethical implications of the project were the main reason. Interestingly,

the project has since been revived under the Advanced Soldier Sensor Information System and Technology (ASSIST) program [34]. The aims of this project are similar, although more limited in scope. DARPA hopes this project's more explicit military goals will ease the concerns raised by the Lifelog project (see Figure 2.6).

Figure 2.6: Future combat gear may feature wearable sensors, including cameras and audio pick-ups, to enhance the soldier's "situational awareness" and after-action reports as a result of the ASSIST project [34].

Current lifeloggers could learn a lot from the experiences of DARPA's Lifelog project. Once lifelogging becomes prevalent, dozens of legal and cultural puzzles immediately surface. What does it mean for society if every individual retains a detailed record of their entire lives? Does lifelogging mean that we will never be able to forget traumatic, or embarassing, experiences from the past; never able to move on? What part of your life is someone else's privacy? Can lifelogs be accessed by others? Can I take back a conversation I had with you? What happens if the lifelog and the biological memory differ? Which is more "correct"? How do we regularly backup, search, and protect a terabyte or more of personal information? Lifelogging raises many serious issues which merit significant attention from researchers from different disciplines. We discuss some of

these concerns in the following sections.

2.3.1 Ethical, Social & Privacy

For many skeptics, the social challenges of lifelogging will doom it to a small minority, or else earn it full prohibition. They don't endorse ubiquitous lifelogging, and find it implausible that anyone else will once they see it in action. In his work with audio lifelogging, Daniel Ellis describes how he frequently encountered shock and resistance from acquaintances when he described the project to create continuous audio archives [21]. Other authors have experienced similar reactions [23, 19]. Ellis also outlines how these concerns can act as a major impediment to the development of these technologies. There's clearly a strong intuitive resistance to having a more detailed record of what people are saying or viewing than memory already provides.

Besides outright resistance to the continuous recording of audio or video, broader social and political questions also need to be addressed. In particular there are questions concerning who owns lifelogged data, how it can be used, and the limits to what is captured. In an era where information about ordinary people travels from the offline world to YouTube or MySpace via mobile camera phones, perhaps this resistance will gradually be worn down. Changing social norms may mean that eventually people will simply accept lifelogs as a fact of life. In such circumstances, their main priority may simply be to protect their own lifelogs in order to safeguard their lives and careers. It has even been suggested that anti-data capture technologies may be developed to block the ability of other people's lifeloggers to record information about them [23]. In an "information age", it may not be permissible for individuals to keep a lifelog private. The changed social context may negate any legitimate expectations of privacy. Former celebrities or criminals, hoping to conceal their past, often find that because information about their past is readily available, it is published. These matters often end up in the courts (e.g. [35]) and it is likely that details from individual lifelogs will too.

Another interesting question relates to the distinction between biological memories and digital memories and which are seen as more objective and true. Biological memory is highly selective and fallible [36]. We do not remember all of our conscious experiences; we misremember many of our experiences; and memory fades over time [37]. One can envisage legal proceedings where the lifelog and the biological memory differ. Which can be relied upon and should third parties, such as legal representatives, have access to someone's lifelogs? By providing your own lifelog, are you effectively giving evidence against yourself? The fact that human memory is fallible has

sometimes been viewed as a weakness of the human mind. However, forgetting is an essential element of the human condition, as illustrated by the story of "AJ", a woman whose inability to forget events and experiences from the past places an enormous strain on her life [38]. AJ describes her memory as "nonstop, uncontrollable, and automatic". AJ spends an excessive amount of time recalling her personal past with considerable accuracy and reliability. If given a date, she can tell you what she was doing and what day of the week it fell on. Some quotes from her demonstrate the burden this infallible memory has placed on her: "I only have to experience something one time and I can be totally scarred by it..."; "I can't let go of things because of my memory...". AJ describes her extraordinary abilities as a burden which dominates her life. Current lifelogging systems are being developed in a manner which could place a similar burden on all of us. Many researchers are now recognising this and are advocating strategies for "forgetting" as an integral part of lifelogging [23, 39, 40].

There are many other similar questions which need to be addressed, however, they are beyond the scope of this thesis. Ultimately, it's likely that society will adapt, and new social norms will come into place, to assist us in working out when and where lifelogging is appropriate, or not. As technology adapts, total recording will become as pervasive as text is to us now. It will be everywhere and we won't even notice it.

2.3.2 Security & Surveillance

The pervasive nature of lifelogging technology raises serious security and surveillance issues. Despite the intuitive negative reaction to audio or video lifelogging, described in Section 2.3.1, there is already large scale public acceptance of mass surveillance, through the use of CCTV cameras. It's worth noting that differences do exist across national boundaries. In the US and Canada, for example, citizens are much less likely to be in favour of CCTV [41]. In comparison, in 2002 the UK had approximately 4.2 million CCTV cameras. This constitutes approximately 20% of the CCTV cameras in use across the world and represents one camera for every 14 people [42]. In addition, in workplaces across the globe, people's activities are intensively watched and recorded. Swipe cards, or computer software to record keystrokes or telephone conversations, can be used to provide rewards and punishments for different workers [43]. Besides the workplace, traffic patterns are increasingly being monitored, providing detailed information on our movements [44] and many of us are willing participants in retail loyalty schemes, thus providing detailed information on our shopping patterns.

Increasingly, these sources of information are being linked together, or used in ways different to their original purpose, in order to provide new information on the living patterns of individuals [45, 43, 46]. In this environment, lifeloggers wearing and recording their own images and video introduce a new element of information which has been characterised as personal sousveillance [19]. Personal sousveillance (see Figure 2.7) refers to the act of bringing the cameras down from on high, controlled by a higher authority, to eye-level, for personal recording of experiences. Lifeloggers capture data from their own perspective, however, lifelogging could also be considered as a surveillance device as lifeloggers will capture information about others who may also be engaging in these acts. As an example, the SenseCam contains a passive infra-red sensor. This sensor is triggered when warm objects, people in particular, pass in front of the camera. This means that one person's lifelog will inevitably capture detailed information about the lives of others, particularly those they are in regular contact with such as family members or work colleagues. Would lifeloggers engaging in personal sousveillance become partners in mass surveillance with higher authorities such as the government? Should personal sousveillance be regulated in some fashion? Perhaps the individuals captured in your lifelog should have the right to demand their removal, but is this really feasible in an era where everyone is engaging in lifelogging?

Figure 2.7: SURveillance ("eye-in-the-sky") versus SOUSveillance: bringing cameras from the heavens, "down to earth" [19].

From a security perspective, lifelogs offer the potential to increase personal safety by providing visual evidence of crimes. However, if the experience with CCTV cameras is replicated, this may

only exist in a forensic capacity, as there is some evidence to suggest that the installation of CCTV cameras doesn't actually reduce crime - they simply help in it's resolution [47]. Lifelogging may have similar effects. The technology may become so pervasive that people simply ignore it, and carry out crimes as usual. Another, more interesting, or sinister, scenario (depending on your perspective) is presented by Frank Nack [48]. The author wonders whether behaviours will be altered in a society of lifeloggers, leading to a conformist society fearful of breaking the rules. In this scenario, personal security would certainly be increased, but at a significant cost.

2.3.3 Technological Challenges

Technically, the challenges involved in lifelogging are enormous. Although the data capture problem has largely been solved (due to the existence of cameras like SenseCam), significant problems remain in the management, maintenance and on-going access to this data. For example, lifelogging will require data to be accessible over many decades. This will require ongoing and active management, as information, and the ability to read it, can currently be lost in just a few years [49]. The MyLifeBits team have already encountered problems in this area, which they describe in the Scientific American [50], where Gordon Bell discovered that he could not access documents because their formats were obsolete. File format issues already cause problems for users as software migrates from one generation to the next, so we must ensure that users are able to open their files long after the systems that created and originally stored them have gone. The need to develop computer systems whose storage will encompass such large periods of time is increasing as we store ever increasing amounts of personal information. For example, proposals for identity cards may require databases that remain in existence for more than 100 years. Serious questions also surround the storage of different types of data (text, audio, visual, log files) in a manner which easily adapts to new hardware and software. How can new information be integrated when technology advances and how can old memories adapt to the new questions which may be asked of it as society changes?

Another issue is the backup and security of all of this data. Backups already cause problems for users and research has indicated that a substantial amount of personal data is not backed up [51]. As the volume of data increases over the decades, the risks associated with the loss of this information also increase. The impact of losing decades worth of digital memories are unknown at this time, but one could envisage deep distress and psychological trauma. The experience of Jim Gemmell, a researcher working on MyLifeBits, provides an interesting insight into this issue.

He described MyLifeBits as like “having a surrogate memory” which created a “freeing, uplifting, and secure feeling”. However, he subsequently lost four months data due to a hard drive crash [52]. This was described as “a severe emotional blow - perhaps like having one’s memories taken away”. One can only speculate on the emotional trauma caused by the loss of years, as opposed to months, of data. Besides the risks associated with data loss, there is the practical issue of backing up over a terabyte of information. With existing technology, it is not practical for the average citizen to backup this volume of information on a regular basis. Data could be stored in online repositories, with users accessing their information using personal devices when required, but this raises privacy concerns. In particular, access rights to the data, and the security of the data from attack, are two issues that would have to be comprehensively addressed in order to satisfy public concerns.

There are many other technological challenges. For example, searching and retrieving audio-visual data is problematic due to the lack of textual annotations. These annotations may be added, but they are unlikely to cover the full range of associations possible with this type of information. Research has also shown that users are often reluctant to annotate large quantities of visual information [53]. Another challenge relates to the ever increasing storage capacity of computers. The focus up to now has generally been on increasing storage so that we can store absolutely everything. Gmail already boasts about how “you’ll never need to delete another message” [54]. Search and retrieval of text is relatively straightforward. But, how would we deal with images, audio, and video? Perhaps an even greater challenge is to understand how we can then use these vast stores of information to generate knowledge. This will be necessary if we want to represent people’s knowledge, experience, and beliefs in such systems. Novel sensor technologies, such as haptic interfaces, will no doubt become commonplace in the future. How can these be integrated and how can the interfaces be adapted to the information extracted from people’s lifelogs from decades earlier? New visualisation techniques will need to be investigated, and existing research such as Lifestreams [55] or Xanadu [17] may provide some of the answers.

However, the primary challenge relates to the management and organisation of this huge volume of information. As computers become increasingly capable of storing a lifetime’s worth of memories in various forms, the question of managing these stores is a serious one, and this particular problem has been designated as one of the main goals of the Memories for Life Challenge [25]. An interview with Gordon Bell in 2006 [56] gives us an insight into this problem:

> "MyLifeBits is now so big that it faces a classic problem of information management: It's hellishly difficult to search, and Bell often finds himself lost in the forest. He hunts for an email but can't lay his hands on it. He gropes for a document, but it eludes him. While eating lunch in San Francisco, he tells me about a Paul Krugman column he liked, so I ask him to show it to me. But it's like pulling teeth: A MyLifeBits search for 'Paul Krugman' produces scores of columns, and Bell can't quite filter out the right one. When I ask him to locate a phone call from one of his colleagues, he hits a bug: He can locate the name of the file, but when he clicks on it the data are AWOL. 'Where the hell is this friggin' phone call?' he mutters to himself, pecking at the keyboard. 'I either get nothing or I get too much!"

Bell's frustration reflects a problem which researchers have long been aware of. Often described as the *shoe-box* problem [5], referring to the manner in which old photographs are often discarded in old shoe-boxes, the sheer quantity of media means that it is extremely difficult to find what you're looking for. Most items will have been forgotten about and it's likely that a huge volume of the stored material will never interest us again. So, how do we search for that one special photo, video, or document, amongst the thousands which will be stored in our lifelogs? How do we design software that can enable computers to perform useful tasks by tapping into this huge database of information? The challenge outlined by Memories for Life, as they relate to Multimedia searching, state that the goal is to search for images or audio by presenting examples, rather than text [25]. This area, the management and organisation of large collections of images, is a key challenge and is the focus of the work reported in this thesis. The overriding concern is how to read, retrieve, and use, this huge ocean of data that your life will generate. We discuss current research in this area in more detail in the following section.

2.4 Image Content Management

The relentless rise in digital imaging technologies, highlighted in Section 2.3.3, has led to massively increased demand for multimedia data storage in integrated database systems. Simply storing these images, however, is the easy part. Storing them in a manner which enables them to be understood, indexed, annotated, and easily retrieved, is challenging. Traditionally, man has outperformed machines in this task, mainly due to our ability to understand the semantics of the imagery. However, traditional methods have generally involved using text to attempt to describe the image

and providing a concrete description of an image in this fashion can prove elusive. Naturally, the interpretation of what we see is hard to characterise, and even harder to teach a machine. However, if the images captured via lifelogging are to ever become truly useful, we must understand the requirements for managing image-based systems, as well as investigating new technolgies for organising, searching, and retrieving images from image databases.

2.4.1 The Need for Image Data Management

The need for efficient storage and retrieval of images is especially important in interactive systems, such as image or multimedia databases. In these systems the user interacts and waits for the database to respond before deciding what the successive interaction should be. This modus operandi poses significant questions for image retrieval, in terms very different than in traditional databases. Traditional databases work in a transaction-oriented mode. A transaction is composed of a query that the user sends to the database and an answer that the database returns to the user. There is no interaction during the execution of the query. As a consequence, real-time retrieval is not terribly important. Although the system needs to be fast, the main criteria is to get a correct answer. The situation in highly interactive image databases is exactly the opposite. Here, a fast answer is more important than a completely correct one, as errors can be corrected in successive iterations, while too slow a response breaks the flow of interaction.

In addition, images are generally described in high dimensional feature spaces, while records are described by a small set of partially independent keys [57]. An elementary search in a database consists of matching a query against the keys. This reduces the problem to a combination of single dimensional problems, in which keys are naturally ordered, thereby allowing the designer to use indexing techniques based on ordered trees. In high dimensional feature spaces, it is impossible to define a total order and, consequently, it is impossible to use the same indexing techniques. To make things worse, image databases require operations like nearest neighbour searches, which are considerably more complex than the simple matching typical of symbolic databases. The combination of a high dimensional feature space and of more complex operations typical of similarity databases create a challenging data management and indexing problem.

2.4.2 Text-Based Image Retrieval

Most existing Image Retrieval systems are text-based (e.g. Google and Yahoo! image search engines), but images frequently have little or no accompanying textual information. The solution historically has been to develop text-based ontologies and classification schemes for image description. Text-based indexing has many strengths including the ability to represent both general and specific instantiations of an object at varying levels of complexity [58, 59].

Attempts to provide general systems for image indexing include: the Getty's Art and Architecture Thesaurus (AAT), which consists of over 120,000 terms for the description of art, art history, architecture, and other cultural objects; and the Library of Congress Thesaurus of Graphic Materials (LCTGM). The AAT currently provides access to thirty-three hierarchical categories of image description using seven broad facets (Associated Concepts, Physical Attributes, Styles and Periods, Agents, Activities, Materials, and Objects). The approach in many collections, particularly general library environments, has been to apply an existing cataloguing system to image description using the LCTGM, or ICONCLASS [60].

However, the textual representation of images is problematic because images convey information relating to what is actually depicted in the image, as well as what the image is about. The ability to read visually oriented material is subject to the knowledge and subject expertise of the reader, and their ability to translate and interpret meaning. For example, the goddess Venus may be used to symbolise love. This example illustrates that in order to interpret the "non-verbal symbolism" of a picture, some degree of background knowledge and subject expertise is required on the reader's part. The description and meaning of a visual image is also open to individual interpretation [61]. Sunderland demonstrates that the ability to interpret visually oriented material is subject to a number of factors which compound the problem further e.g. age, gender, social grouping, etc. [62]. The phrase "one picture is worth a thousand words" emphasises the difficulty faced by information professionals in developing effective indexing systems to manage such material.

Manual assignment of textual attributes is also both time consuming and costly. As we have found in the MediAssist project, the manual annotation of a large collection of images takes a considerable amount of man-hours to complete [63]. The MediAssist collection consisted of approximately 11,000 images and required the work of up to 10 individuals to manually annotate the images over an extended period of time. When we consider that lifelogging devices like the SenseCam produce approximately 2,000 images per day, and therefore up to 14,000 per week, we can

see that this approach is not feasible for the management of images captured during lifelogging.

As more projects involving the electronic storage and retrieval of images have been developed, the problems of indexing digital images have become more acute. The difficulties involved in finding appropriate solutions to the problems of indexing images, combined with the physical task of indexing the volume of images, make the task of human indexing infeasible. It is evident from the problems outlined that there is now a growing requirement to develop alternative approaches and methods to manage, organise, navigate, and retrieve an ever increasing number of visually oriented material.

2.4.3 Content-Based Image Retrieval

In contrast to the text-based approach of the systems described above, content-based image retrieval uses features of the image itself to aid retrieval. Colour, texture, and shape are usually used as part of this process. These visual features are extracted automatically in the indexing process when images are entered into a multimedia database. Queries and retrieval can be based directly on the visual properties of the image, and returned results ranked by the degree of content matching. This ability to store and retrieve images within a single application has created exciting new opportunities for improved management and access to visually oriented material. Two reviews of content-based image retrieval systems outline the numerous techniques, and diverse applications, in which these systems have the potential to play a principal role [64, 65].

For example, many attempts have been made to organise collections of personal photographs into albums or events. In Similarity Pyramids, photographs are organised and clustered according to their colour [66]. A similar approach was taken by Rodden et al. [67]. PhotoTOC uses colour histograms to assist the clustering process, while Loui et al. use a block-based colour histogram correlation method [68, 69]. Jaimes et al. employ the use of both colour and edge features in their work, as well as using information specifically related to the orientation of objects within the image [70]. Boutell et al. use a combination of colour histograms and texture features extracted in a 4×4 block configuration and then classified using a Support Vector Machine (SVM) [71]. Cooper et al. state that "events are difficult to define quantitatively or consistently" and that photos taken at the same event often exhibit little coherence in terms of both low-level features and visual similarity [72]. However, even after drawing these conclusions, they still employ temporal and low-level data to detect events in their digital photo collections. AutoAlbum uses a content-based clustering algorithm, known as best-first probabilistic model merging, which forms clusters out

of temporally contiguous photographs [73]. They employ both time and content based clustering and obtain results which are often semantically meaningful. Other systems employing the use of colour or texture features include [74, 75, 76, 77, 78], while a more detailed description of these features can be found in [79, 80].

Besides global descriptors, such as those described above, other content-based features can be used to identify similar objects within a database of images. This is a challenging problem due to viewpoint or lighting changes, deformations, and partial occlusions that may exist across different examples. Global image features, based on image properties such as colour or texture, have proven to be of limited use in these real-world environments. Instead, researchers have recently turned to representations based on local features that can be reliably detected and are invariant to the transformations likely to occur across images (i.e. photometric or various geometric transformations).

One approach has been to use a corner detector to identify repeatable image locations, around which local image properties can be measured. Schmid et al. [81] developed one of the earliest object matching systems using these features. They extracted local gray value feature points with a Harris corner detector, and then created a local image descriptor at each interest point. These image descriptors were used for robust object recognition by looking for multiple matching descriptors that satisfied object-based orientation and location constraints. However, this approach only examined an image at a single scale. As the change in scale becomes significant, these detectors respond to different image points.

More recently, there has been great progress in the use of invariant features [82, 83] for object matching. With these features, robustness to small changes in viewpoint as well as to partial occlusion is achievable and objects can be recognised anywhere in an image, with arbitrary size, rotation, and without using a previous object segmentation step [84]. It follows, therefore, that these features can be matched more reliably than traditional methods such as cross-correlation using Harris corners.

A key element, therefore, of many photo management tools is the use of low-level content based tools to assist in the management and organisation of the images. It is generally acknowledged that providing truly efficient user-centric access to large content archives requires indexing of the content in terms of the real world semantics of what it represents. However, many photo management tools also use context information, and many authors claim that the use of context information alone is superior to using low-level features. Naaman et al. state that content based tools are not yet - and will not be in the near future - practical for meaningful organisation of

photo collections [53]. They believe that while low-level features can be extracted, the gap between these and understanding semantics is still wide. Davis et al. come to similar conclusions in their work developing browsing and sharing software for mobile phone images [85, 86]. They state that the challenge of finding salient moments in any one photographer's personal collection is a very difficult problem using signal-based analysis techniques, and that if the temporal and social correlations in the automatically gathered contextual metadata are analysed, salient events, trends, and patterns in the photos taken by groups and individuals can be more easily determined. By creating a temporal histogram representing how many photos co-located users took over time, they can visualise the photographic activity and level of interest of individual's and groups in a spatio-temporal context. The location based data is provided via the Cell ID of an individuals mobile device or a Global Positioning System (GPS) device when available. Graham et al. developed two photo browsers for collections with thousands of time-stamped digital images [87]. They exploit the timing information alone to structure the collections and to automatically generate meaningful summaries. A similar approach is employed by Cooper et al. where the temporal information is combined with the low frequency discrete cosine transform (DCT) coefficients from each photo [72]. O'Hare et al. used low-level MPEG-7 features along with a wide range of contextual data, including GPS, temporal, and manually annotated information. In a search for known objects, they demonstrated that combinations of contextual metadata and content-based data can achieve improved results [4]. Aizawa et al. demonstrated similar results through the combination of face-detection techniques with various different sensors, including GPS, and accelerometers [88].

Once image features are extracted, the question remained as to how they can be indexed and matched against each other for retrieval. Most content-based image retrieval systems, using low-level features for image representation, calculate image similarity based on the distances between feature vectors in the feature space. Given this fact, Euclidean (L2) distance has been the most widely used distance measure [77, 76, 78, 89]. Other popular measures have been the weighted Euclidean distance [75, 90], the city-block (L1) distance [74, 89], the general Minkowsky Lp distance [91], and the Mahalanobis distance [76, 89]. The L1 distance was also used under the name histogram intersection [89]. Berman & Shapiro used polynomial combinations of predefined distance measures to create new distance measures [92].

However, as previously discussed, many photo management systems also use features generated by many sources other than low-level features, and not all of these features have the same range. Popular distance measures, for example the Euclidean distance, implicitly assign more

weighting to features with large ranges than those with small ranges. Feature normalisation is required to approximately equalise the ranges of the features and make them have approximately the same effect in the computation of similarity. In most of the database retrieval literature, the normalisation methods were usually not mentioned or only the normality assumption was used [74, 78, 93, 90]. The Mahalanobis distance [94] also involves normalisation in terms of the use of a covariance matrix in the calculations and produces results related to likelihood when the features are normally distributed [94].

2.4.4 Visualisation

A key factor in the acceptance and popularity of an image retrieval system is the presentation of the results and the general look and feel of the user interface. In addition, users must be able to easily form queries in order to find what they are looking for. The user interface typically consists of a query formulation part and a result presentation part. Specification of which images to retrieve from the database can be done in many ways. One is to browse through the database one by one. Another way is to specify the image in terms of keywords, or in terms of image features that are extracted from the image, such as those outlined in Section 2.4.3. Yet another way is to provide an image or sketch from which features of the same type must be extracted in order to match these features. Alternatively, the user can provide positive or negative feedback about the retrieval result, so that the system can refine the search.

Relevance feedback, in particular, was a major advance in user interaction technology for image retrieval. Relevance feedback is a query modification technique which attempts to capture the user's precise needs through iterative feedback and query refinement. It can be thought of as an alternative search paradigm, complementing other paradigms such as keyword based search [65]. In the absence of reliable methods of capturing high level image semantics, the user's feedback provides a way to learn case-specific query semantics. Important early work that introduced relevance feedback into the image retrieval domain included the MARS system [90], IBM's QBIC system [77], and MIT's Photobook [76].

Another approach for visual query specification which provides the user with a high degree of freedom is Query-By-Sketch (QBS). QBS is primarily based on having the user compose an example image using colour distributions; basic geometric shapes such as circles, squares, triangles, and rectangles; free-hand sketching, or combinations of these (see Figure 2.8). The main goal of QBS is to let the user create a template of either a completed object or scene, which is used as a

basis for similarity matching against an image collection. According to Venters et al. [95], there is little evidence to support the usability of such query tools, and these interfaces remain one of the least researched and developed elements of CBIR retrieval systems. However, it is generally acknowledged that the main drawback with this approach is that it is highly dependant on the user's ability to create good example images.

Figure 2.8: An example of using QBS for image retrieval

Besides query formulation, the presentation of the search results is also an important factor. The most popular method, used by the Google and Yahoo! image search engines, is to order results based on some relevance score. Other systems display images in chronological order [72, 87]. Other forms of visualisation include clustering of images based on their visual content or associated metadata, or organising them in a hierarchical fashion [96]. Combinations of any of the methods outlined can also be used. An overview of image retrieval systems, highlighting many of the query formulation and presentation strategies discussed, can be found in [97]. Other practical issues involved in visualisation, such as how users frame their queries, are beyond the scope of this discussion, but are discussed in detail in [64, 67, 65].

2.5 Applications

With the appropriate image management tools available, a myriad of applications can be developed using the extensive coverage of one's life provided by a passive capture device. In the following sections, we outline a number of potential applications for Visual Diary style applications. The key requirement for each of these applications is the ability to manage and organise the large volume

of data produced by the passive capture process.

2.5.1 Time Budget Studies

Many studies have been undertaken which examine how people spend their time and how this is related to daily experiences, but there is no generally accepted method for gathering this data. Studies generally focus on the well-being of the population at large and attempt to analyse this using surveys or time-budget studies [98] [99]. Other studies rely on global reports of happiness or satisfaction with life in general, or with specific domains such as work and family [100] [101]. Although there is no universally agreed approach to data gathering, a number of methods have been proposed. The *Day Reconstruction Method* (DRM) assesses how people spend their time and how they experience the various activities and settings of their lives by combining features of time-budget measurement and experience sampling [102]. Participants systematically reconstruct their activities and experiences of the previous day by constructing a diary consisting of a sequence of episodes. They then describe each episode by answering questions about the situation and about the feelings they experienced. The Experience Sampling Method (ESM) [103] is designed to measure the quality of people's lives by prompting them to record where they are, what they are doing, and how they feel, several times throughout the day. The technique is reported to provide a rich description of a sample of moments in respondents' lives, while avoiding the distortions that affect delayed recall and evaluation of experiences. However, experience sampling is expensive, involves high levels of participant burden, and provides little information about uncommon or brief events, which are rarely sampled. The DRM involves a similar burden on the particpants and faces similar problems in practice.

Although these studies are difficult to carry out, their utility is not in question. Kahneman et al. [102] describe how this information is useful to: medical researchers for assessing the onset and development of different illnesses and the health consequences of stress; to epidemiologists interested in social and environmental stressors; to economists and policy researchers for evaluating policies and for valuing non-market activities; and to anyone who wishes to measure the well-being of society. We believe that the next logical step in overcoming the difficulties associated with traditional methods of gathering the required data is the use of passive capture devices to facilitate the automated gathering and structuring of the required information.

2.5.2 Alzheimer's Sufferers

An initial report from Microsoft has demonstrated how the SenseCam can be used in order to assist people with short term memory loss [32]. The most widespread neurodegenerative diseases are Alzheimer's and Parkinson's [104], which are characterised by short and long term memory loss. Alzheimer's disease is an irreversible neurodegenerative disorder that progressively degrades the brain's ability to maintain normal executive, attention, and memory functions. In Ireland, it is estimated that 40,000 people suffer from Alzheimer's [105]. The figure worldwide is approximately 37 million and, with ageing populations, this is expected to increase rapidly over the next 20 years [106]. The economic cost of dementia in Ireland is estimated at 400 million euros, of which up to 76% of the cost is family care [107]. In July, 2006, the worldwide cost of Alzheimer's and dementia care was estimated at a staggering 248 billion dollars, and this is expected to rise sharply as the world's population continues to age [108]. A treatment that could delay the onset of Alzheimer's by 5 years would reduce the number of sufferers by 50% in 50 years. The use of a Visual Diary, as used by Microsoft in their studies, could lead to improvements in a patient's memory and enable them to retain new information for longer, thus combatting the early effects of such conditions.

2.5.3 Behavioural Related Illnesses

Behavioural factors play a critical role in health management. Utterback states that the primary goal of disease management programs is to improve patient health and quality of life [109]. This is achieved by encouraging the patient to alter his or her behaviour, manage his or her health, and control his or her symptoms. They also describe how home health care programs in which nurses and other health professionals visit patients in their homes have served to facilitate disease management in patients who are chronically ill, particularly for those who are less mobile or homebound. However, the acute shortage of clinicians, tighter government regulations, and growth in chronic disease, has compromised the effectiveness of these programs. Previous research has introduced photography into diabetes self-management routines to help patients make their behaviours explicit and work with physicians to see possible correlations between self medication and long-term health [110]. Past approaches have manually collected images, however, passive image capture would allow this process to be automated. Other areas in which digital cameras have been employed to assist in patient monitoring include Congestive Heart Failure, Chronic Ob-

structive Pulmonary Disease, hypertension, and wound care [111]. By using a Visual Diary, we could gain a better understanding of how to improve the diagnoses and treatment of illnesses that are highly influenced by behavioural routines.

2.5.4 Personal Security

Traditional security and surveillance systems involving CCTV cameras are widespread. A lot of research has investigated automating these systems using machine vision techniques [112]. A simple and low cost solution to personal security may be developed using the images provided by a low-cost passive capture device such as the SenseCam, WayMarkr, or Campaignr projects. Any dangerous or harmful incident occurring would be captured by the device and would be easily retrieved using the techniques we develop in this work. It is acknowledged, however, that this would only be useful after the event occurred (e.g. in a forensic capacity).

2.5.5 Stroke Patient Rehabilitation

Stroke rehabilitation is a restorative learning process which seeks to hasten and maximise recovery from stroke by treating the disabilities caused by the stroke, and to prepare the stroke survivor to reintegrate as fully as possible into community life [113]. By it's very nature, it is a lengthy process carried out over a number of months and involving multiple sessions per week. Previous research has focused primarily on patients who have some degree of hemiparesis (partial paralysis affecting only one side of the body), with or without other neurological deficits, and who are candidates for treatment in an interdisciplinary rehabilitation program [114]. Evaluation of these techniques is a recurring and unsolved problem. Colombo et al. use robot-aided techniques for upper limb rehabilitation and have developed new evaluation metrics which should enable the therapist to implement targeted rehabilitative strategies [115]. However, in general, the therapist relies on his or her experience to make judgments on the effectiveness of the rehabilitation. Improvements generally occur gradually and are difficult to detect across sessions. A framework that allows the therapist to evaluate long-term progress, as well as compare patient's at similar stages of therapy, would provide a richer insight into the rehabilitation process. By monitoring a patient's daily activities using a passive capture device, and subsequently constructing a Visual Diary of relevant events over a long time period, it may be possible to assist the therapist in this process.

2.5.6 Simple Memory Aids

Losing car keys, or similar, is a problem everyone has experienced. Using a Visual Diary, one could quickly and simply scan through the diary to find out the location where the keys were last located, as they would presumably have been captured by the camera. This may seem a trivial application but the use of automated object classification/recognition in SenseCam images could make it particularly challenging.

2.5.7 Home Monitoring of Health and Living Patterns

Many home monitoring technologies have been proposed to detect health crises, support aging-in-place, and improve medical care [116]. The potential costs, and fears over breaches of privacy amongst health professionals and members of the public, mean that these technologies have had a limited impact to date. However, there is some evidence that these systems may be more readily adopted if they are developed as tools for personalised use, thus helping users learn about the conditions and variables which affect their physical health. If done well, these same tools can then be used by researchers for ethnographic studies of people and their behaviours in non-laboratory settings. Visual Diary systems could prove useful in this regard.

2.5.8 Nurses Aid

Two large studies, one conducted in Colorado and Utah, and the other in New York, found that adverse events (e.g. anesthesia mortality, medication errors, misdiagnoses, etc.) occurred in 2.9 and 3.7 percent of hospitalisations, respectively. In Colorado and Utah hospitals, 6.6 percent of adverse events led to death, as compared with 13.6 percent in New York hospitals. In both of these studies, over half of these adverse events resulted from medical errors and could have been prevented [117]. One method to help improve safety and decision making in hospitals is to make use of sensor networks. Kuwahara et al. developed a context-aware environment to support a nurse's work in a just-in-time manner [118]. They model the context of a nurse's activity based on nursing care manuals. By understanding and modelling a nurses context, they hope to be able to provide guidance as to what task the nurse should perform next. The system does not currently use any visual information due to the privacy issues associated with taking photographs of patients in a hospital setting. If, however, these issues could be overcome, we believe that the use of Visual Diary style applications could be a valuable additional tool in modelling the nursing environment.

2.6 User Application Scenario

Imagine the following typical scenario: During a normal day in the life of John, he gets up at 7.00am. Between 7.00am and 8.00am he has breakfast, and then he cycles to work, following the same route, arriving at 9.00am. He sits at his desk and works until 10.30am, at which time he joins his colleagues for morning coffee. This is followed by another couple of hours at work, including occasional meetings, until 1pm. He takes an hour at lunch before returning to an afternoon of work at his desk, broken up by people entering his office asking for advice and discussing issues about a research project. At 6pm, he leaves work and cycles home, eats dinner at 7pm, watches TV, and goes to bed at 10pm. Subsequent days may have a similar pattern of activity but John may decide to go to a soccer game after dinner one evening, or meet friends whom he has not seen recently for lunch or coffee. The latter activities would be significant or important when considered over a period of a number of days while his other activities such as meals, travel, and work, are recurring and vary little from day to day.

A feature of the above scenario, familiar to most people, is the notion of an event as an identifiable activity in a person's day. If we are to use SenseCam images as a component of a person's lifelog, it is important to be able to segment, and then identify, these events. We discuss this further in Section 5.2. Once this is done, we then require the ability to classify and to relate, or link, events which are recurring and distinguish them from once-off or unique events. Setting Detection has been identified as a key enabling technology in this regard in Section 3.3 and we outline a number of approaches designed to achieve this in Chapter 4.

At some point, the user will have gathered a certain quantity of images and will decide to upload them to the Visual Diary application (see Figure 2.9). However, certain users may decide to upload their images on a daily basis, whilst other users may only upload images once a month. There will be a significant difference in the volume of images uploaded in each case and we need to develop techniques to reduce the burden on the user in each of these scenarios. For a user who chooses to upload images on a daily basis, it is reasonable to request the user to annotate the settings captured during that day's images in order to allow the system to detect these settings in the Visual Diary. Similarly, if the user only uploaded images on a weekly basis, it would also be reasonable to allow them to annotate the settings in these images. In Section 4.3, we describe our initial experiments which require the user to annotate the settings contained within their collections. This approach is suitable when the user only wishes to upload a relatively small

number of images. However, what happens if a user doesn't upload images for a month, or a year? In this case, annotation may not be feasible due to the huge volume of images involved. In Section 6.3.3, we describe an automatic approach which allows the system to propose new settings to the user as images are uploaded. This enables the user to upload a significant volume of images without the associated increased annotation effort.

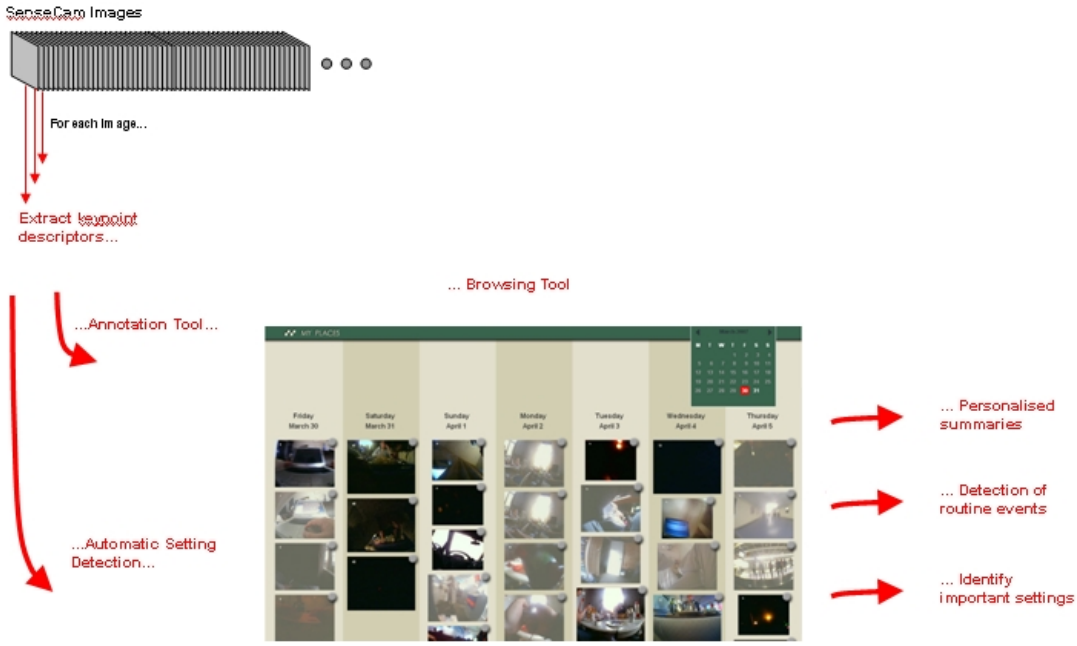


Figure 2.9: An illustration of the user application scenario. The user captures a number of SenseCam images and loads them into the Visual Diary. The keypoints are extracted and depending on the volume of images involved, the user may decide to annotate the images or allow an automatic setting detection algorithm to run. Once completed, the user can browse through their images via the user interface. The interface also facilitates further analysis of the images, such as personalised summaries of a day, detection of routine settings, and the highlighting of settings as being important.

Once settings have been detected, the user needs to be able to browse through their collection. In this regard, a browsing tool is developed which displays keyframes from the events detected in a user's image. Certain keyframes will be linked together, representing the settings detected in these images. This application allows the user to browse through different events representing the activities which occurred in the uploaded images. The detected settings are linked together in this application, allowing the user to quickly view images from the same setting. This browsing tool is discussed in more detail in Chapter 5, where we also report on a user evaluation of this application.

Another important element in a Visual Diary application is the ability for users to mark settings

as being important or to highlight them as being a favourite. The annotation tool described in Section 4.3.3.1 allows the user to mark settings as being important to them during the annotation process. Any matched settings detected are then also given a similar level of importance. However, if settings are being proposed using the automated approach described in Section 6.3.3, or if they wish to change the importance assigned during the annotation phase, they can achieve this by toggling the favourite identifier on or off on the user interface. We discuss this scenario further in Section 6.3.2.

Besides the basic facilities of detecting settings and browsing via the user interface, the user may have other requirements which can be facilitated by providing a more detailed analysis of the detected settings. By analysing the detected settings we can determine certain characteristics of particular settings, and hence, characterise the user's activities during that time. A profile of a users day can be produced, detailing where they spent their time during that day. Alternatively, a profile might be developed which characterises a user's activity across a particular time period (e.g. one week). We can then match these profiles to determine if there are any deviations in the user's activities. For example, in the scenario outlined above, John's routine profile might consist of the regular routine activities which normally occur during the day. By extracting and analysing this profile, we can then attempt to summarise John's day (in terms of settings), and also try to detect deviations from John's normal routine, such as when he goes to play soccer. We discuss this analysis of the detected settings in Section 6.2, where we describe a number of tools to analyse a user's Visual Diary and provide a personalised summary of their lifelog.

2.7 Discussion

The ultimate dream of lifelogging is to create and preserve a complete and useable record of one's own life. The growth of digital imaging, social networking, blogging, etc., would seem to indicate a public appetite for something similar. However, the implications of comprehensive lifelogging are not clear, and have certainly not been openly discussed outside of the academic environment. Indeed, as disucssed in Section 2.3.1, public reactions have sometimes been hostile to the realities of full-life logging. The overall purpose of a lifelog is also not yet clear, although applications from entertainment, improving health, sharing experiences, etc., can easily be imagined. However, whatever the motives for lifelogging, the creation of such a detailed record of an individual's life has unsettling implications for society at large.

As discussed in Section 2.2, the technology to enable people to fully record every aspect of their lives does not yet exist outside of the laboratory. However, it's clear from the discussion in Section 2.3 that many issues need to be addressed before the public at large will accept these technologies. Now is the time to consider these implications whilst much of the technology is still in the design phase. For example, no one should be required to keep a lifelog, nor should suspicions arise if someone decides not to keep one. Personal lifelogs should remain the property of that individual and recordings of others should not be made without permission. Facilities should be put in place to delete or add content at will. The issues raised warrant a much more detailed discussion than is possible in this work. We must hope that the changes in quality of life which occur due to the proliferation of lifelogs will not come at a cost of a deterioration in privacy, increased surveillance, or an escalation of state interference in individuals' lives.

The technological challenges discussed in Section 2.3.3 are intriguing. For example, ensuring long-term access to information sounds straight-forward, but there are many issues to consider. Simply leaving the data on a disk, or allowing a commercial entity to host it, will not ensure continued access to the information. What happens if you don't pay your subscription fee or the company hosting the data goes out of business? For information to last decades, it will need to migrate from disk to disk, emulate old applications and environments, and possibly even change format. However, the type of information we store is constantly changing. The locations we store data are diverse (e.g. flash drives, hard disks, camera, phones, photo sharing sites), so ensuring the continued existence of our information will require constant attention and careful management. The diversity of data formats, as well as the differing locations, also create issues for backup, archival, and security, of information. In addition, we must really ask ourselves whether we really need to store all of this information? Do we want our children, or grand children, to see this when they go through our archives? The importance of forgetting was highlighted in Section 2.3.1 and is an essential part of what defines us as individuals and it may be necessary to integrate it into our lifelogs.

The issues discussed here are extremely important ones, however, the main focus of this thesis is the management and organisation of the visual content generated by lifelogging. In particular, we are interested in managing the images generated by passive capture devices. In the future, passive capture may incorporate video, but for the purposes of this investigation, we restrict ourselves to images captured via devices similar to those outlined in Section 1.2. Although the capture and storage of these images has become a trivial matter, the increasing volume of images generated

by passive capture devices are creating problems. Specifically, the methods of managing these collections have not kept pace with the technology used to acquire them. Naaman et al. [53] describe how the photo collection management problem can be categorised into tools which enable easy annotation of photos, tools which allow fast visual scanning of the images, and content-based tools. However, they also identify the problems associated with each of these, such as difficulties for consumers with annotation, inability of tools to allow fast visual scanning to scale to many thousands of images, and the semantic gap in relation to content based tools. As highlighted in Section 2.4.2, a significant amount of effort is required to manually annotate a relatively modest image collection consisting of 11,000 images. When we consider the potentially exponential rise in the size of image databases generated by lifelogging, we can see that this approach is not scalable in practice. Tools which allow the user to quickly scan through the images are not sufficient in isolation. A software interface has been provided by Microsoft to allow the fast visual playback of a day's worth of SenseCam images, but it does not allow meaningful interaction beyond pause, and very often the user spends extended periods of time watching repetitive events and images.

Other forms of visualisation, discussed in Section 2.4.4, such as relevance feedback or QBS, are not particularly suitable for a Visual Diary style application. These methods of interaction are more suited to applications requiring a more interactive style of querying and retrieving images from an image repository. In these applications, different queries can be formed, depending on the user's requirements at different points in time, and feedback can be provided in order to obtain improved results. There may be other applications involving data from a visual lifelog where this would be suitable, however, a Visual Diary revolves more around browsing through the collection as opposed to interactively querying and retrieving specific images from it. With this in mind, the goal should be to structure the images using an appropriate method before presenting them to the user. The design and implementation of such an interface is outlined in more detail in Section 5.2.

In Section 2.4.3 we discussed the use of context information in order to facilitate the automatic structuring of image collections. Some authors believe context data alone is sufficient to organise a photo collection and research in this area is promising. However, there are many problems with the use of context data in our work. For example, temporal data can be used to assist in the structuring of a photo collection. Platt et al. use the creation time of the photographs and their colour histograms to automatically detect events [68]. If the creation time is not available, they use the order the photographs were taken to impose a temporal order on the photos. A similar approach is taken by Loui et al., where they use K-means clustering to organise the photos based

on date/time and then use colour information to detect events [69]. However, these applications are designed for personal image collections, where photographs are not taken on a continuous basis. Hence, detecting events based on date/time information and colour information alone is not suitable in a continuous visual lifelog of images. Other authors have exploited the "bursty" nature of photo capture [87]. For example, lots of pictures will be taken at a wedding, but few, if any, may be taken until another significant event takes place. Even with the wedding event, it may be possible to detect sub-events. Bursts of photographs may be taken when the bride walks up the aisle, when the couple exchange vows, when speeches are being made, etc. In a visual lifelog, pictures are taken continuously, and not in short bursts, so this method is not suitable.

However, temporal data is widely used in the development of algorithms to detect shot transitions and scene changes in video [119]. Many current systems operate at the shot-level following an initial temporal segmentation. This is desirable for both computational efficiency and the extraction of semantics associated with some temporal duration. Because shots provide the most natural organisational unit for video above the frame, shot segmentation enables hierarchical processing of content in video management systems. The continuous nature of recording which occurs using passive capture devices, such as SenseCam, means that the associated content could be viewed as being similar to the individual frames in a video. Therefore, temporal data can be used in this scenario to provide an initial structuring of visual lifelog data into events and we discuss this issue in more detail in Section 5.2. Furthermore, temporal data can also be used to further analyse the results of the setting detection process. We discuss this in more detail in Chapter 6. However, although temporal data can be used to facilitate the detection of a group of similar images (temporally aligned), thus facilitating the automatic detection of a single setting (as discussed in Section 6.3), we believe it is of limited benefit in matching images across settings which occur across numerous days, weeks, months, or years. Similar work in the area, comparing matching individual frames versus shot-based matching, concluded that matching on a frame by frame basis was more effective (although computationally more wasteful) [120]. In addition, as the amount of frames used in the shot-based matching technique increased, performance significantly deteriorated.

Other forms of context data include logging GPS or GSM data. Toyama et al. outline a number of reasons why location information is so important [121]. They describe how a synergy exists between location information and images and how location is intimately tied to the semantics of imagery. For example, knowing that a photograph was shot at Croke Park, a large football stadium in Dublin, Ireland, says a lot about the photo even before a single pixel is viewed. They also note

the following:

- Location is universal. Location, if represented properly, offers a universally understood context that transcends language, culture, and user-dependent taxonomies
- Location scales well. Location data can contain arbitrary degrees of accuracy and precision
- Browsing by location, whether via maps or by textual place names, is well-understood and intuitive to users
- Studies show that users associate their personal photos with event, location, subject, and time. Three of these are frequently, if not always, tied to location: event = time + location; location is location; and subject is often defined by combinations of who, what, when, and where.
- Finally, location data is becoming increasingly available from a number of channels

For these reasons, we believe that accurate information on a user's location can provide critical information to assist in structuring a lifelog of images. In particular, we focus on locations deemed to be important to the user.

While acknowledging the potential importance of location information, tracking a user's location over an entire lifetime is a challenging problem. It seems clear that existing technology is not sufficient to gather the required information over an entire day, week, year, or lifetime. For example, GPS does not work indoors where many people spend a significant portion of their lives. Recent studies have also shown that GPS coverage is only available for 4.5% of the time a user carries a device over a typical day [122]. Although GSM signals are ubiquitous and work almost everywhere, determining location based on GSM signals is difficult. Although services are being rolled out by network operators, they are usually expensive - as much as $1 per user query in the US [123]. Researchers, therefore, have focused on methods of determining location using software loaded onto the mobile device itself. The most extensive work is that conducted by Intel Research, where excellent results were obtained, but only by logging the GPS coordinates of every GSM cell tower in the Seattle metropolitan area [124]. Although this information is available in the US, it is not widely available in the EU for commerical and security reasons. It's also not clear how these approaches would work in rural environments, where cell density is often extremely low. In addition, although people believe that their mobile device is always at hand, there is evidence

to suggest otherwise [125]. Other forms of tracking a user's location, such as bluetooth or WiFi, are also problematic due to the insufficient numbers of access points or bluetooth enabled devices in the environment. In specific application scenarios, each of these technologies may be suitable, but not to track location across an entire lifetime. Although research in this area continues, an appropriate approach for now may be to allow the user to identify locations important to them, and to use this information to assist in structuring the Visual Diary.

Another potential source of context data is the sensors on the SenseCam itself. However, previous work has found the infra-red sensor and the temperature sensor to be of little benefit in segmenting SenseCam images into events [126]. The optimal combination of features to segment a lifelog of image data into events was found to be a combination of low-level MPEG-7 features extracted from the image and data from the light and accelerometer sensors from the SenseCam. However, this work made no attempt to recognise similar locations or settings that have been identified by the user as important, a key objective in this work. The light sensor is useful for detecting transitions from one room to another, but the settings we're interested in generally have little light change as they occur in the same location. In addition, a user is not necessarily completely stationary at a particular location. There may be movement to the left or right within the same location. An analysis of the accelerometer data, therefore, may not yield much additional information either.

However, if we assume a suitable method for constructing events from SenseCam images exists, we can leverage this event based structuring of the data to enable us to recognise settings which are of importance to the user [126, 127]. For example, in Figure 2.10, the process of segmenting a single day's SenseCam images into events is illustrated. Essentially, the images are analysed using content or context (or both) information, and subsequently segmented into distinct events based on specific thresholds or criteria used by the event detection process. However, this process does not allow us to recognise those locations or settings which the user has indicated are important. On the other hand, the detection of settings allows us to leverage this initial event-based structuring of the data to locate images from similar locations without resorting to the types of localisation technologies previously discussed (e.g. GPS, etc.). In this scenario, setting detection can be seen as an additional layer, on top of the event detection process, which allows us to link images from similar locations together using the techniques developed in this thesis (discussed further in Chapter 4). An illustration of the setting detection process is shown in Figure 2.11. In this example, a number of similar settings have been detected across eight days SenseCam images. In addition, a number of unique settings have also been detected. The distinction between settings

and events is necessary as an event based analysis provides only an initial partition of the data, which although helpful in structuring the data, is limited in providing additional insight into the activities contained in the users Visual Diary.

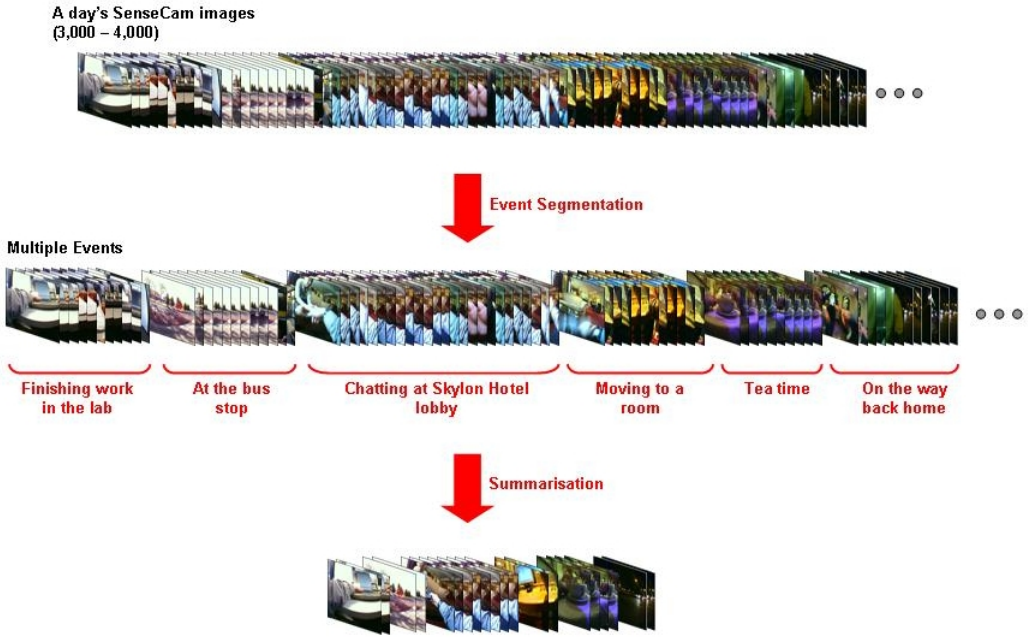


Figure 2.10: The event detection process in a single day's SenseCam images

For these reasons, we believe the best solution is to provide the user with tools to construct a Visual Diary based on images selected using content based analysis tools. In order to assist in determining locations of importance, an annotation tool is required which minimises the amount of effort required by the user. Not withstanding the availability of such a tool, the task of identifying similar images within a database remains challenging due to viewpoint or lighting changes, deformations, and partial occlusions that may exist across different examples. Global image features based on image properties such as colour or texture (as highlighted in Section 2.4.3), have proven to be of limited use in these real world environments. Indeed, existing colour information in images is often discarded because of the fact that invariance to different lighting conditions, such as shadows or illumination changes, are hard to achieve when using colour features. Instead, researchers have recently turned to representations based on local features that can be reliably detected and are invariant to the transformations likely to occur across images (i.e. photometric or various geometric transformations). Local features often correspond with more meaningful image components such as rigid objects and entities, which make association of semantics with image

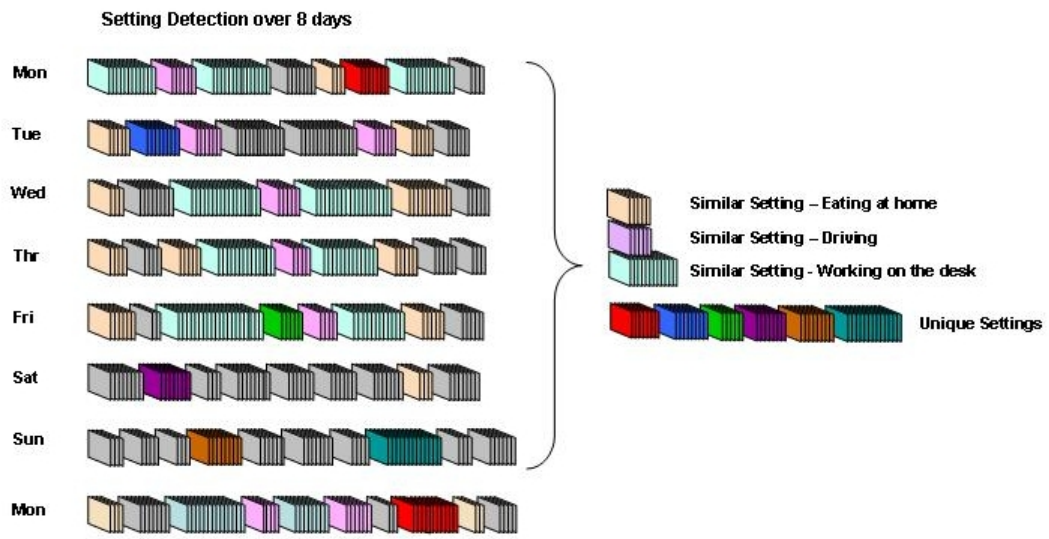


Figure 2.11: The setting detection process over 8 day's SenseCam images

portions more feasible in practice [65].

Many local descriptors exist, however, extensive studies have shown that SIFT descriptors outperform other texture descriptors for object recognition on various types of image data, including 3D objects and real world scenes [128, 84]. The SIFT descriptor is a gradient orientation histogram robust to illumination and viewpoint changes [83] and has been successfully used in various works (e.g. [129]). However, the recently proposed Speeded-Up Robust Features (SURF) method has been reported to acheive greater matching accuracy and computational efficiency [130]. We propose to use these features in our work, and a more detailed discussion of relevant approaches follows in Chapter 3.

2.8 Conclusion

In this chapter, an overview of the current state of the art of lifelogging was presented, thus completing the first objective of the thesis. Some of the commercially available applications were discussed, as well as highlighting the lifelogging efforts of various digital artists in their own specific domains. A detailed overview of lifelogging applications in the academic environment was also presented. The benefits and drawbacks of lifelogging were also introduced and, although lifelogging has many potential benefits for individual users, it's clear that there are many complex legal, social, and ethical issues that researchers are only just beginning to examine.

In addition to these issues, a number of technological challenges also exist if full lifelogging is to be realised. Of these, the issue of managing the increased volume of image data is the focus of this thesis. Techniques for managing image data over the years were introduced, but existing approaches, such as textual annotations, were found to be unsuitable for managing a lifetime's worth of images. Other forms of context data, such as temporal or location data, were also found to be unsuitable, mainly due to the continuous nature of capturing lifelog images and the difficulties associated with tracking location over a lifetime. However, although existing technologies to track location are not sufficient, location was acknowledged as an important source of information.

The one source of data which can be relied upon, as it is always available, is content-based data - data gathered from the image itself. Although content-based data guarantees the supply of information necessary to enable us to analyse each image, there are a number of choices available when choosing content-based descriptors. Global information, such as colour, texture, or shape, works well in specific domains, however, in real-life scenarios, where images may be occluded, or variations in lighting or viewpoint may exist, they often fail. In addition, global descriptors often fail to capture the semantics of an image. Local and invariant features in particular, have been found to perform significantly better with real-world images. For this reason, we choose to use local invariant features in this work, and these features are examined in more detail in the following chapter to determine how they can be used to facilitate the construction of a Visual Diary of lifelog images.

Finally, we also outline a number of potential applications for a Visual Diary, as well as presenting a typical user application scenario. We discuss how a user might use information about settings in a Visual Diary application. We presented a basic scenario where a user wants to upload different volumes of data and browse through the Visual Diary application. We outline how settings might be detected using an automated approach or an approach requiring the annotation of a user's settings. We also discussed how the user might select their favourite settings using both of these approaches. Besides these core application features, we also discussed other tools a user might use to analyse their settings in order to acquire a personalised user profile of their activities during the time period the images were captured.

CHAPTER 3

Setting Detection

3.1 What is Setting Detection?

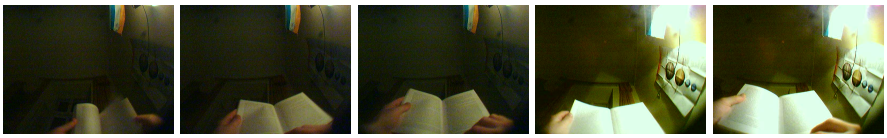
The previous chapter highlighted how local invariant descriptors can potentially be used to assist in the management of large collections of images. In Chapter 1, we also discussed how passively captured images can be used to construct a Visual Diary of an individual's life, providing a visual summary of the most important aspects of their day, week, month, or year. In order to construct the Visual Diary, one major challenge needs to be overcome, namely, the management and organisation of the associated large volume of images (discussed in Section 2.4). Methods to assist in solving this problem are vital to enable the presentation of images in a visually coherent manner. In order to identify representative samples to present to the user in the Visual Diary, we propose an approach involving two key elements. The first element of the strategy involves an analysis of the images using local invariant features such as SIFT or SURF. The second involves the identification of images taken in locations or *settings* that are deemed to be important to the owner. The combination of these two elements is what we refer to as *Setting Detection*.

A *setting* in this context refers to those images taken at the same location in the real world (e.g. in the dining room at home, in front of the computer in the office, in the park, etc.) that have been flagged by a user as being important to him/her for some reason. Examples of ten distinct settings can be seen in Figure 3.1. These types of image sequences occur frequently when a user is wearing a passive image capture device, such as the SenseCam. Any user wearing the device, and remaining in the same location for a period of time, captures images of the scene which are very similar in terms of their visual content. There may be some slight movement from side to side as the user naturally shifts their position slightly, but the essential elements of the scene remain the

same.

Although the criteria discussed above form the essential definition of a setting, the images shown in Figure 3.1 illustrate a number of characteristics of settings which are a result of the distortions introduced by the wide-angle lens used by the SenseCam. In particular, it is clear that many of the images annotated as being in a setting consist of a dominant object in the centre of the scene, with other objects in the scene being dramatically reduced in size due to the properties of the lens. This is evident in the images captured both indoors and outdoors in Figure 3.1, although a single object is not always as prominent in those images captured outdoors. Thus, although the key characteristics are that the images occur in exactly the same location, they must also occur when the user is in a relatively static position for the images captured to remain relatively similar. For example, a user sitting in a chair at work will capture many images of themselves working on their laptop. These images are considered a setting. However, if the user rotates in their chair 180 degrees, the subsequent images captured are not considered to be part of the same setting (although the user is in the same location). This subtle difference between location and setting is another characteristic which can be utilised in the detection of settings as not only can the items near the user captured by the camera (and which will dominate the image) be used for matching, but the background features can also be used as these will also remain relatively static. An example of a sequence of images which would not be considered a setting is when the user walks down a street. The images captured may have various objects which dominate each individual image as the user walks by street furniture or other items such as vehicles, buildings, people, etc., as well as having a slowly changing background. However, these images are not considered a setting as the user is not in a static location (although they may be visually quite similar). We contend in this thesis that detecting such settings is a key enabling technology that allows us to structure the large numbers of images that passive capture devices collect and that in turn allows us to help the user in constructing and maintaining a Visual Diary.

To perform setting detection, it is necessary to select and implement suitable methods to identify visually similar backgrounds in SenseCam images using visual features, as described in Section 2.7. However, another benefit of the use of visual data alone is that it allows the technology to be easily deployed to other devices (e.g. a mobile phone running the Campaignr software [8] - see Section 1.2) and to be adapted to other application scenarios, discussed in Appendix D. The approach used must also be capable of dealing with changes in image viewpoint or perspective, varying lighting conditions, partially occluded images, and other distortions which may occur



(a) Reading in bed

(b) Having dinner

(c) At a restaurant

(d) Sitting in the park

(e) Eating ice cream

(f) Working on computer

(g) At a cafe

(h) Reading in the castle grounds

(i) On an aeroplane

(j) On a train

Figure 3.1: Sample Images from 10 different settings

naturally in images. In the remainder of this chapter, we review related work in this particular area.

3.2 Related Approaches

The most relevant approches to our work are in the fields of object recognition, scene classification and video segmentation. The ability to detect specific objects with certain visual attributes within images would be of clear benefit in the construction of a Visual Diary. For example, one can imagine how the ability to detect a laptop or PC would be very useful when detecting settings where the user is working. Alternatively, we may be interested in detecting specific classes of objects within images. In this instance, rather than attempting to detect a specific object, we try to learn a model that corresponds to all the objects of a particular class. The semantic label of the object class can be relatively broad (i.e. encompassing many different sub-classes), or quite restrictive, depending on the application requirements.

In scene classification, the goal is to place an image automatically into one set of *physical* (e.g. indoor/outdoor, orientation, etc.) or *semantic* categories (e.g. beach or party) [131]. For example, if a person recognises sky at the top of a photo, sea in the middle, and sand at the bottom, he or she may surmise that the image is a beach scene, even if they cannot make out every detail in the image. However, classifying images into semantic categories is a difficult problem in practice. A large collection of photos may need to be grouped into many different categories like landscape, portrait, animal, indoor, outdoors, beach, party, mountain, forest, etc., to support efficient browsing. To search over a large collection of images, we might want to classify the images by the depiction of certain scenes (e.g. an office), and objects (e.g. buildings), and by semantic topics (e.g. politics). The choice of categories, scene types, objects, or semantic concepts, which a user could choose from is potentially limitless, so this remains an extremely challenging problem. However, scene classification is extremely valuable in image retrieval from databases because an understanding of the scene content can be directly used for efficient and effective database organisation and browsing.

Video segmentation consists of detecting shot boundaries and scenes in video content. A shot is an unbroken sequence of frames from one camera. Thus, a movie sequence that alternated between views of two people would consist of multiple shots. A scene is defined as a collection of one or more adjoining shots that focus on an object or objects of interest. For example, a person

walking down a hallway into a room would be one scene, even though different camera angles might be shown. Three camera shots showing three different people walking down a hallway might be one scene if the important object was the hallway and not the people. Many automatic and semi-automatic approaches to video segmentation have been developed. The rate at which video content is now being generated precludes metadata creation with substantial manual processing, thus, there are strong similarities between this domain and that of detecting settings in passively captured lifelog data.

Setting Detection is a niche area in image classification. To the best of our knowledge, no other authors have attempted to detect settings in the context of a Visual Diary application. Identifying images taken "at a computer" or "on the train" is slightly more general than classifying according to whether the image contains a particular object, or set of objects, but less general than scene classification, where images can be taken at a different physical location, but still be part of the same scene. Video segmentation generally involves detecting shot boundaries or attempting to characterise the shot. Typically, shots can be characterised by two factors: the underlying scene, and the camera work. Therefore, shot detection involves a slightly more complex set of parameters than setting detection. However, we review related research in these areas as they are most relevant for this work, and can provide some insights into the approach necessary to perform setting detection in passively captured images.

3.2.1 Object Detection & Recognition

It has long been recognised that the ability to detect or classify objects in images plays a vital role in visual systems in order to divide them into manageable categories [132, 133, 134]. Because humans outperform the best machine vision systems, building a system that emulates object recognition in the visual cortex has always been an attractive idea. Indeed, many models of biological vision have been used in object recognition tasks [135, 136, 137]. It seems intuitive that information about the presence or absence of particular objects in images contributes significantly to the overall semantics of the image. For example, knowing that a photograph contains a car says a lot about the photo even before the image is actually viewed. Therefore, the detection and recognition of these objects can help to narrow the semantic gap and is extremely useful for a variety of applications.

There is an extensive body of literature on object recognition and many different approaches exist (e.g. [138, 133, 139, 140, 129, 141, 142]). These include neural network based approaches,

graph matching, genetic algorithms, and fuzzy systems [143]. Schmid et al. describe how these approaches can be divided into model- and appearance-based approaches [81]. Model-based approaches use 3D models of the object shape to represent an object with geometric features such as lines, vertices, and ellipses, while global or local photometric features are used for appearance-based approaches. In this discussion, we focus on appearance-based approaches as they are the most relevant to the approach employed in this thesis.

As previously mentioned in Section 3.1, the task of identifying similar objects within a database of images is extremely challenging due to viewpoint or lighting changes, deformations, and partial occlusions, that may exist across different examples. Many object recognition systems use global features which describe an image as a whole. Such features are attractive because they produce very compact representations of images, where each image corresponds to a point in a high-dimensional feature space. As a result, any standard classifier can be used. The earliest work on appearance-based object recognition mainly utilised global descriptions such as colour or texture histograms [77, 75]. The main drawback of such methods is their sensitivity to real-world sources of variability such as viewpoint and lighting changes, clutter, and occlusions. As a result, these approaches implicitly assume that an image only contains a single object, or that a good segmentation of the object from the background is available. Therefore, global features have been shown to be of limited use in real-world scenarios. Other works have used Boosting to detect faces in images where the weak hypotheses employed were the thresholded average brightness of collections of up to four rectangular regions [144]. Agarwal et al. [145] used Winnow as the underlying learning algorithm for the recognition of cars from side views, while Shneier used colour and shape features to detect road traffic signs [145, 146].

A different paradigm is to use local features, which are descriptors of local image neighbourhoods computed at multiple interest points. Local features are usually extracted from numerous image regions around interest points and store visual information (colour or texture) about these regions in local descriptors [81]. One of the key issues in dealing with local features is that there may be differing numbers of feature points in each image, making a comparison between images more complicated. Typically, interest points are detected at multiple scales and are expected to be repeatable across different views of an object. The interest points are also expected to capture the essence of the object's appearance. The feature descriptor describes the image patch around an interest point. The usual paradigm of using local features is to match them across images, which requires a distance metric for comparing feature descriptors. This distance metric is used to devise

a heuristic procedure for determining when a pair of features is considered a match (e.g. by using a distance threshold). The matching procedure may also utilise other constraints, such as the geometric relationships among the interest points.

A significant amount of recent work has focused on the use of local features that can be reliably detected and are invariant to the transformations likely to occur in realistic environments [82] [83]. A typical object recognition system that works with local features performs the recognition task in the following steps:

1. First, the objects of interest are learned, meaning that local descriptors are extracted from images of these objects and are stored in an object database.

2. Local descriptors are also extracted from the test images, in order to detect objects in these images.

3. These are subsequently matched against the descriptors in the object database.

4. After the best matching descriptor pairs are found, an optional verification step (e.g. geometric verification) can be performed to decide whether an object appears in the test image or not.

One of the more popular local descriptors to emerge has been the SIFT descriptor, used to match objects in a manner that is invariant to location, scale, and orientation. Some robustness to small shifts in local geometry is also achieved by representing the local image region with multiple images representing each of a number of orientation planes [83, 147, 148]. Schugerl et al. used a combination of SIFT keypoints and MPEG-7 features extracted from the same interest point to obtain better results than either descriptor on their own [143]. Low quality images, large view and scale changes, and blur negatively influence these results. Another approach has been to use a corner detector to identify repeatable image locations, around which local image properties can be measured [149]. The exploitation of quantised local descriptors was used by Csurka et al. for object recognition [150]. The authors proposed to represent images using a histogram of the quantised local descriptors (bag-of-keypoints). Similar work was carried out in the museum environment by Fockler et al. and Bay et al. [151, 141]. Other variants of SIFT, such as PCA-SIFT and GLOH (Gradient Location and Orientation Histogram) have been successfully applied for many image matching applications [152, 128]. The recently proposed SURF method also locates interest points and extracts an invariant descriptor for each point. However, SURF achieves

greater computational efficiency by using integral images [130]. Zhang et al. employ a generative probabilistic approach using a Gaussian Mixture Model in order to improve the results of Lowe's original work [153].

Other researchers have used parts-based models which are a combination of local descriptors and their spatial distributions. The nature of the spatial relationship imposed between the local parts influences the model used. Examples include fully independent (bag-of-features, each representing a part or region), and fully connected (constellation model). For example, Fergus et al. model objects as a flexible constellation of parts, where each part has an appearance, relative scale, and can be occluded or not [133]. Using an EM-type learning algorithm, they achieved very good recognition performance. However, a fully connected model limits the number of parts that can be modeled, since the algorithm complexity grows exponentially with the number of parts. This often means that a good deal of the available image information must be ignored, especially in cases where the objects have many parts, either naturally, or because fine grained local visual features are being used to characterise them. Indeed, such structural approaches often fail to compete with geometry-free "bag-of-features" style approaches because the latter make better use of the available image information. As a compromise, sparser topologies have been proposed, such as: the star topology [154]; a hierarchy, with the lowest levels corresponding to local features [155]; and a geometry, where local features are spatially dependent on their nearest neighbours [156]. Other authors have used contextual information, corresponding to the distribution of local structures, and boosting to yield improved levels of performance [157, 158], while Zhang et al. use segmentation to reduce the number of salient points for enhanced object representation [159]. Finally, Dorko et al. introduced an approach for constructing and selecting scale-invariant object parts, however, objects of interest are manually pre-segmented, dramatically reducing the complexity of distinguishing between relevant patches on the objects, as opposed to background clutter [138].

3.2.2 Scene Classification

Scene classification is concerned with the automatic labelling of images with a semantic concept or category. Examples might be beach, mountain, indoor, or outdoor scenes (see Figure 3.2). Scene classification is important as it can vastly improve the ability to manage large image collections, whilst also facilitating subsequent processing on the images, like object recognition [160]. One can easily imagine that if an image has been classified as a beach scene, we would be more likely to try to detect objects related to that scene (e.g. people, umbrella, boat), as opposed to attempting

Figure 3.2: Images showing different categories of scenes such as indoor, outdoor, urban, mountain, etc.

to detect a mountain hut. Scene classification, therefore, is extremely useful for applications requiring the ability to efficiently and effectively organise and browse large image collections, and it has been widely used in content based image retrieval systems [161, 162, 163].

Object detection or recognition attempts to detect a single, or perhaps every, object in an image. Scene classification differs in that a scene can be classified without having a full knowledge of every object in the scene. Clearly, this is not an easy task because of the variations in illumination and scale which exist in natural images. In some cases the use of low-level information, such as colour and texture, might be enough to classify some scenes. However, in complex applications, although object recognition might be necessary, it may be sufficient to only detect certain objects from the scene, instead of all of them. For instance, if a person sees trees at the top of an image and grass at the bottom, he can hypothesise that he is looking at a forest scene, even if he can not see every detail in the image [131].

However, the question remains whether we can use image features alone to describe a scene or do we need to know which objects are present. The most common approach has been to compute low-level features (e.g. colour, texture, shape, etc.), which are processed with a classifier engine for inferring high-level information about the image. This approach has been used for several years to classify images into several semantic classes such as indoor, outdoor, city, landscape, beach, mountain, etc. [64]. Boutell et al. conducted an extensive survey on the state of the art in semantic scene classification [131]. They examined the features available, such as low-level features and camera metadata, and also provided a brief review of the learning and inference engines used for classification (e.g. K-nearest neighbour, Bayesian classifier, Support Vector Machines (SVM), etc.). They provided a review of scene classification systems and divided these systems into two types. Exemplar based systems use pattern recognition techniques using low-level image features or semantic features. Model-based approaches leverage the expected configuration of a scene. Interestingly, the use of camera metadata is not mentioned in any of the systems reviewed. The use of camera metadata, in combination with low-level features, is discussed in further work by Boutell et al. [164]. Here, a Bayesian network is used to fuse content-based data and metadata, with some promising results in specific contexts (e.g. indoor/outdoor classification). Vailaya et al. used histograms of different low-level cues to perform scene classification [165]. Different sets of cues were used depending on the two-class problem at hand: global edge features were used for city vs. landscape classification, while local colour features were used in the indoor vs. outdoor case. Boutell et al. use only LUV colour moments in a 7×7 block layout to perform multi-label

scene classification, but the use of colour means that their system is not very robust to viewing angle or lighting changes [71].

The methods outlined above assume that any type of scene can be described by the colour or texture properties of the image. For instance, a forest scene presents highly textured regions (trees). On the other hand, a beach scene is described by an important amount of blue (sky) and yellow (sand), while the presence of straight horizontal and vertical edges denotes an urban scene. This works well for restricted classes of scenes, but is limited for natural images which may vary much more in scope. More advanced approaches have attempted to model scenes by using a semantic intermediate representation in order to help reduce the semantic gap (e.g. a mountain scene mainly contains trees, rocks, or snow) [166, 167].

Another issue worth considering is whether feature information is relevant at all for scene classification. This is an open question as some researchers have found that humans can classify complex natural scenes extremely quickly with little need for a detailed analysis of the image [168, 169]. Both of theses studies pose a serious challenge to the currently accepted view that to understand the context of a complex scene, one needs first to recognise the objects and then in turn recognise the category of the scene [170]. However, this work is more concerned with the modelling of the human visual system and is beyond the scope of the work reported here.

More recently, a number of approaches have emerged using local features which attempt to deal with the semantic gap between low-level features and high-level concepts. Many of these approaches use visual codewords as shown in Figure 3.3 [171, 172]. For such whole-image categorisation tasks, bag-of-features methods, which represent an image as an orderless collection of local features, have recently demonstrated impressive levels of performance [153, 173]. Schaffalitzky & Zisserman describe a system to match camera shots which are images of the same real world location in a film [120]. They use two features: one based on interest point neighbourhoods, the other based on the Maximally Stable Extremal Regions (MSER) of Matas et al. [174]. In both cases, an elliptical image region is used to compute the invariant descriptor. Their system also employs semi-local and global contraints (e.g. using epipolar geometry) to boost matching accuracy. In more recent work, Quelhas et al. and Fei-Fei et al. have shown that a bag-of-keypoints representation can be further decomposed into mixtures of latent semantic models [175, 176]. Such latent models enable clustering and ranking of images into meaningful groups. Other authors have used the so called "wide-baseline" methods which allow for significant variation in the scale and viewpoint of different scenes [177, 174, 178, 82].

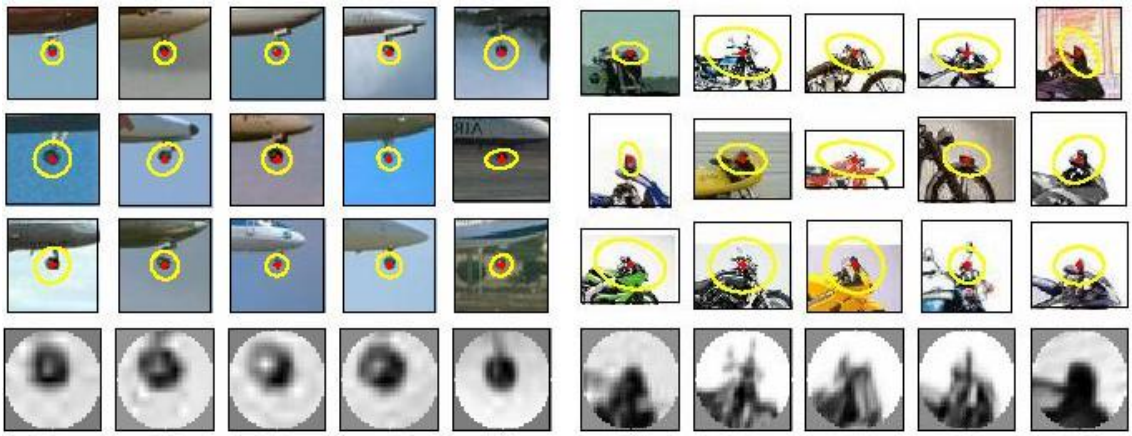


Figure 3.3: Two examples of visual codewords. The top three rows show examples of the visual word and the bottom row shows the affine normalised regions for the top rows of images [179].

Other relevant works include the scene completion work of Hays et al. where the authors patch up holes in images by matching similar scenes using the GIST descriptor of Torralba et al. [180, 181]. Brown et al. construct panoramas of matching images using the SIFT descriptor [182].

3.2.3 Video Segmentation

The goal of video segmentation is to divide the video stream into a set of meaningful and manageable segments (shots) that are used as basic elements for indexing. Each shot is then represented by selecting key frames and indexed by extracting spatial and temporal features. The retrieval is based on the similarity between the feature vector of the query and already stored video features. A shot is defined as an unbroken sequence of frames taken from one camera. There are a number of different types of transitions or boundaries between shots. A cut is an abrupt shot change that occurs in a single frame. A fade is a slow change in brightness usually resulting in, or starting with, a solid black frame. A dissolve occurs when the images of the first shot get dimmer and the images of the second shot get brighter, with frames within the transition showing one image superimposed on the other. A wipe occurs when pixels from the second shot replace those of the first shot in a regular pattern such as in a line from the left edge of the frames. Many other types of gradual transition are possible and gradual transitions are generally more difficult to detect than cuts [183].

A large number of shot detection algorithms have been developed and they can be classified based on a few core concepts [184, 185]. Algorithms operate on either compressed or uncompressed video (although the majority work on raw data). For example, Yeo & Liu's algorithm uses

an MPEG compressed video stream rather than the raw footage [186]. However, the distinction between compressed and uncompressed data is not very important, since practically all of the algorithms can be applied to both. There may be differences in how certain features are computed, but the core concept of the algorithm remains the same. Typically, systems work by defining a similarity measure between successive images. When two images are sufficiently dissimilar, there may be a cut. Based on the metrics used to detect the difference between successive frames, the algorithms can be divided broadly into numerous categories, briefly described below. A detailed analysis of all of these methods is beyond the scope of this thesis. However, a detailed overview of the merits, limitations and performance characterisation of each of the approaches is found in [187].

3.2.3.1 Pixel Differences

The easiest way to detect if two frames are significantly different is to count the number of pixels that change in value more than some threshold. The total is compared to a second threshold to determine if a shot boundary has been found. This method is sensitive to camera motion. Zhang et al. implemented this approach with the additional step of using a 3×3 averaging filter before the comparison to reduce camera motion and noise effects [188]. They found that by selecting a threshold tailored to the input sequence, good results were obtained, although the method was somewhat slow. However, manually adjusting the threshold is unlikely to be practical.

In contrast to template matching based on global image characteristics (pixel by pixel differences), block-based approaches use local characteristics to increase the robustness to camera and object movement. Each frame i is divided into b blocks that are compared with their corresponding blocks in $i + 1$. The difference between blocks can be measured using a variety of measures. Shahraray divided the images into 12 regions, and found the best match for each region in a neighbourhood around the region in the other image [189]. This matching process duplicates the process used to extract motion vectors from an image pair. The pixel differences for each region were sorted, and the weighted sum of the sorted region differences provided the image difference measure. Gradual transitions were detected by generating a cumulative difference measure from consecutive values of the image differences.

Hampapur et al. computed what they call chromatic images by dividing the change in grey level of each pixel between two images by the grey level of that pixel in the second image [190]. During dissolves and fades, this chromatic image assumes a reasonably constant value. They also

computed a similar image that detects wipes. Unfortunately, this technique is very sensitive to camera and object motion.

3.2.3.2 Statistical Differences

Statistical methods expand on the idea of pixel differences by breaking the images into regions and comparing statistical measures of the pixels in those regions. For example, Kasturi et al. compared corresponding blocks using a likelihood ratio [191]. Compared to template matching, this method is more tolerant to slow and small object motion from frame to frame. On the other hand, it is slower due to the complexity of the statistical formulae. An additional disadvantage is that no change will be detected in the case of two corresponding blocks that are different but have the same density function. Such situations, however, are very unlikely. It also generates many false positives (i.e., changes not caused by a shot boundary).

3.2.3.3 Histograms

Histograms are the most common method used to detect shot boundaries. The simplest histogram method computes grey level or colour histograms of the two images. If the bin-wise difference between the two histograms is above a threshold, a shot boundary is assumed. Swanberg et al. used grey level histogram differences in regions, weighted by how likely the region was to change in the video sequence [192]. This worked well because their test video (CNN Headline News) had a very regular spatial structure. They did some simple shot categorisation by comparing shots with the known types (e.g., anchor person shot) in a database. They were also able to group shots into higher level objects such as scenes and segments by matching the shot types with the known temporal structure. Zhang et al. compared pixel differences, statistical differences and several different histogram methods and found that the histogram methods were a good trade-off between accuracy and speed [188]. In order to properly detect gradual transitions such as wipes and dissolves, they used two thresholds. If the histogram difference fell between the thresholds, they tentatively marked it as the beginning of a gradual transition sequence, and succeeding frames were compared against the first frame in the sequence. If the running difference exceeded the larger threshold, the sequence was marked as a gradual transition. To reduce the amount of processing needed, they compared non-adjacent frames and did finer level comparisons if a possible break was detected. A major advantage of using a histogram as a feature is that the histogram is relatively insensitive to the object-position in the frame and, thus, this technique is suitable in the presence

of camera and/or object motion. However, this measure is sensitive to noise, illumination changes, and object-scaling, and does not scale well with matching [187]. Nonetheless, the histogram remains the most commonly used feature in video segmentation due to its ease of implementation and its effectiveness.

3.2.3.4 Edge Tracking

Zabih et al. compared colour histograms, chromatic scaling, and their own algorithm based on edge detection [193]. They aligned consecutive frames to reduce the effects of camera motion and compared the number and position of edges in the edge detected images. The percentage of edges that enter and exit between the two frames was computed. Shot boundaries were detected by looking for large edge change percentages. Dissolves and fades were identified by looking at the relative values of the entering and exiting edge percentages. They determined that their method was more accurate at detecting cuts than histograms and much less sensitive to motion than chromatic scaling.

3.2.3.5 Motion vectors

Zhang et al. used motion vectors determined from block matching to detect whether or not a shot was a zoom or a pan [188]. Shahraray used the motion vectors extracted as part of the region-based pixel difference computation described above to decide if there is a large amount of camera or object motion in a shot [189]. Because shots with camera motion can be incorrectly classified as gradual transitions, detecting zooms and pans increases the accuracy of a shot boundary detection algorithm. Motion vector information can also be obtained from MPEG compressed video sequences. However, the block matching performed as part of MPEG encoding selects vectors based on compression efficiency and thus often selects inappropriate vectors for image processing purposes.

3.3 Discussion

It is clear from the discussion above, that a variety of approaches to object detection/recognition, scene classification and video segmentation exist. In terms of object detection/recognition and scene classification, many of these approaches are concerned with closing the semantic gap and with obtaining high performance with real-world, or natural, images. Wang et al. have described

images as having four levels of semantic content [163]:

- semantic types (e.g. landscape photograph)
- object composition (e.g. a bike and a car parked on a beach, a sunset)
- abstract semantics (e.g. people fighting, a happy person, an objectionable photograph)
- detailed semantics (e.g. a detailed description of a given picture)

The research described, and indeed our own work described in this thesis, is concerned with the first two levels in this hierarchy. The bottow two levels are generally considered as scene understanding. Good progress has been made at the first level, which relates closely to scene classification, as outlined in Section 3.2.2. Similarly, research is ongoing at the second level, which is closely related to object detection and recognition, as outlined in Section 3.2.1. The second level, dealing with the content of the image, and therefore with a higher level of semantic content, is more difficult to classify than the first. Questions such as: "Is this an image of a field or a beach?"; "Is it a portrait or a picnic?"; remain difficult to answer. Our primary interest in this investigation is to answer questions similar to these. Images taken at the same location, or setting, require the system to identify images taken at work, at home, in the pub, etc.. However, the detection of settings goes beyond this. While many of the scene classification systems described above focus on classifying general scenes (e.g. all beach scenes or all mountain scenes), or on detecting specific objects (Is there a red car in this image?), or specific classes of objects (Find all images with cars?), setting detection requres the ability to detect images of the same beach, the same mountain, or the same car, with the images all taken from slightly different perspectives over extended periods of time. As we discussed in Section 3.1, although the lens on the SenseCam introduces distortions which make certain objects more prominent in the image than others, the key goal is to detect images taken in exactly the same location with similar background. Therefore, the requirements of setting detection differ from those of object detection.

Shot detection algorithms are generally classified based on their suitability for detecting specific types of transitions (e.g. hard-cut, fade, wipe, slide or dissolve). Decision parameters vary for transition types and the chosen algorithm. In addition, numerous features have been used for video segmentation (as previously discussed). Apart from thresholding, hidden-Markov models, tree-classifiers, supervised learning, and clustering were used by different researchers to detect

shots. Lienhart discussed the underlying concepts behind each transition type based on the characterisation of the video-data in terms of higher-level semantics, and suggested guidelines for use of the tested approaches [194]. Therefore, given the wide range of approaches, its clear that the choice of the best video partitioning method is not straightforward.

The typical forms of transition which occur in video segmentation do not generally occur in passively captured images (e.g. fade, cut, dissolve). Although state of the art video segmentation systems exhibit excellent abrupt transition detection performance, the detection of gradual transition detection remains relatively poor [185]. As previously stated, the principal challenge in most of these systems is distinguishing amongst transition effects, object motion, and camera motion using low-level frame features. Techniques to detect gradual transitions may be relevant to passively captured lifelog data, but the other effects typical of video should not generally occur. However, techniques applied in the temporal segmentation of video can assist in other areas of our work (discussed in Section 2.7), whilst the work of others to detect similar scenes in movies using local invariant features (discussed in Section 3.2.2), is relevant to the detection of settings. Thus, while the previous research described can certainly inform our approach, our requirements remain slightly different.

Therefore, our approach is based on the assumption that specific locations or settings undergo only minor changes between all occurrences in the lifelog. For instance, an individual's clothes may change across different pictures of him/her standing in the kitchen at home, but the kitchen itself generally remains the same. Other examples, taken from the five different users whose images have been used in these experiments, can be seen in Figure 3.4. These images show how changes in clothing, viewpoint, even having different individuals in the image, can change over time, but the essential components of the setting remain the same. Essentially, the learned sample and the other occurrences of the setting remain similar, despite varying lighting conditions and viewpoints in these scenes. The excellent results achieved in Section 4.5.1 indicate that this assumption is valid, although we acknowledge that the data sets used in this thesis cover a limited time frame.

However, detecting these settings is a very hard problem since the pixel values that correspond to multiple pictures of a particular setting can undergo significant transformations. These transformations occur because of differences in viewing angle, distance, and lighting conditions. As discussed in Sections 3.2.1 and 3.2.2, global features are extremely sensitive to these transformations, and for this reason, they are of limited use when detecting settings in lifelog images.

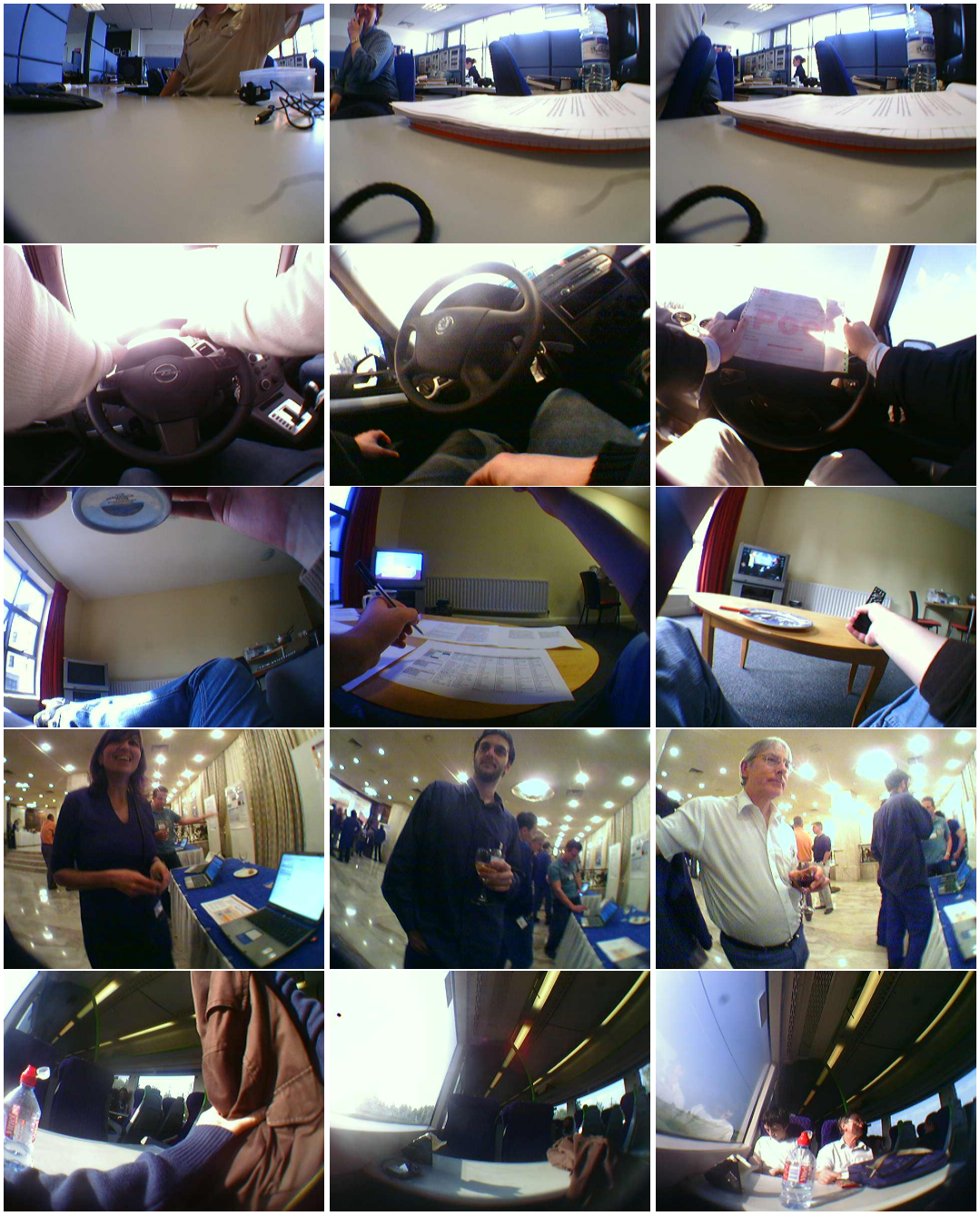


Figure 3.4: Variations in settings from five different users across the lifelog

Similarly, in Section 3.2.3, we highlighted how the most commonly used method in video segmentation, the histogram, is also sensitive to these kinds of transformations. Therefore, we have identified local image features, which are expected to be invariant to all kinds of affine transformations, as the most promising features to use in our work. They also do not require a segmentation of the object from the background, unlike many texture features, or representations of the object's boundary (shape features). The most popular choice is the SIFT descriptor. Mikolajczyk et al. have compared several descriptors for matching and found that SIFT descriptors perform best [128]. However, as discussed in Section 2.7, the more recent SURF descriptor has been reported to achieve better performance than SIFT, and therefore warrants consideration in our work.

One issue worth considering when working with local features is the required level of invariance. This is generally dependent on the expected geometric and photometric deformations which may exist in the image collection. In setting detection, extremely large changes in viewpoint are not expected, as we are only interested in images taken at very specific locations with small changes in viewpoint. For example, an individual sitting at their desk facing the computer may capture numerous images of the computer whilst working. There may be small changes in viewpoint due to the natural movement of the individual, or due to a small amount of rotation to the left or right in their chair. However, larger movements which result in an image with little of the relevant scene remaining, would not constitute part of the setting, and are therefore not considered in our analysis. We only focus on the detection of those images captured repeatedly in exactly the same location. This may seem restrictive but, due to the high rate of image capture, they occur quite frequently in passively captured images. Therefore, the "wide-baseline" methods may not be suitable in this scenario. Indeed, Lowe describes how the additional complexity of full affine-invariant features often has a negative impact on their robustness and does not pay off, unless very large viewpoint changes are expected [83]. Of more interest are features that are invariant to changes in scale, illumination, rotation, and small changes in viewpoint, as these are the most commonly occurring deformations. In some cases, even the rotation invariance can be left out, as the camera only rotates around the vertical axis. This has been incorporated into the SURF descriptor in a version known as "upright SURF" (U-SURF) [130]. The benefit here is reduced complexity and increased speed and discriminative power. Concerning the invariance to changes in illumination, a simple linear model with a scale factor and offset is generally used.

The SIFT and SURF features, therfore, would appear to be the most relevant features available to perform setting detection. As previously mentioned, to the best of our knowledge, no other

authors have attempted to detect settings in a Visual Diary application. The most similar work by other researchers is probably that of Schaffalitzky et al., where the authors automatically detected similar locations in movies [120]. This work is interesting because, rather than attempting to detect general locations (scenes) in the movies (e.g. all shots taken of buildings), they attempted to detect when shots of the same scene occurred at different points in the movie. This is extremely close to our requirements for setting detection. However, processing time in this system is of the order of hundred's of hours and significant tuning of separate processes is necessary in order to create a working system. Another interesting work from this author is the VideoGoogle system, which attempts to introduce the speed and flexibility of Google searching to image search and retrieval [129]. The analogy with text retrieval is that the descriptors are quantised into clusters which represent visual words in text retrieval. Other interesting work is the detection of near-duplicate images described by Ke et al. [195]. They describe near-duplicate images as images which have undergone a slight alteration from the original. The transformations possible in setting detection are more significant, nonetheless, work on near-duplicate detection is relevant, particularly as they use PCA-SIFT in their work, achieving excellent results.

3.4 Conclusion

In this chapter, setting detection was introduced. Therefore, this work fulfils the second objective of the thesis, and also constitutes part of the first research contribution, namely, a comprehensive evaluation of the most appropriate techniques available to perform setting detection. We described what setting detection means in relation to lifelog images and we also outlined the two key elements involved in detecting settings in a visual lifelog. These are the extraction of low-level features using local invariant descriptors and the identification of settings that are important to the user. The different existing approaches to the extraction of local invariant features is the main focus of this chapter as the identification of important settings is performed using an annotation tool, discussed in Section 4.4.1.

Following the description of setting detection, we discussed a number of approaches relevant to setting detection. The most relevant research is in the related fields of object detection and recognition, scene classification and video segmentation. A detailed overview of the literature in these areas was provided, and although it cannot cover the entire body of work in these areas, it provides a good overview of the challenges associated with this work. Research in these areas

has been ongoing for many years and in all fields there has been a growing interest in the use of local descriptors, replacing the initial research which focused on global descriptors. In terms of object detection and recognition and scene classification, local descriptors have been viewed as being much more tolerant to the variations in objects and scenes which generally occur in natural images and can help to reduce the semantic gap. In video segmentation, local features can help increase robustness to camera and object movement.

The review of research in these areas informs the approach we take to setting detection. However, setting detection differs slightly to all of these problems. In a spectrum from object detection to scene classification, setting detection could be viewed as being somewhere in between. We are not interested in detecting specific objects in an image, nor are we interested in detecting different categories of scenes or classifying them into semantic concepts. Rather, we seek to detect images taken of the same scene, at the same location, but from differing viewpoints and with the images captured over a potentially extended period of time. The most similar research involves detecting the same scenes in movies (or other video content) or in near-duplicate image detection (as discussed in Section 3.3). This work, along with the work of previous researchers, will help inform our approach. In particular, local invariant features such as SIFT or SURF seem to be appropriate choices. These features are examined in more detail in the following chapter to determine how they can be used to facilitate the construction of a Visual Diary of lifelog images. We will also outline a number of approaches based on the use of these features and present our experimental results.

CHAPTER 4

Setting Detection using Visual Interest Point Detectors

4.1 Introduction

In the previous chapter, we presented a definition of setting detection and reviewed it in relation to the most relevant research areas of object detection and recognition, scene classification, and video segementation. We argued that setting detection was not the same as these tasks, as we are not seeking to detect specific objects or detect specific classes of scenes. Nor are we interested in characterising sequences of lifelog images in terms of hard-cuts, fades or dissolves, as these operations are not applicable. We also demonstrated how local interest point detectors have been used successfully in each of these areas and demonstrated how these approaches can inform our own work on detecting settings in passively captured lifelog images. In particular, the SIFT and SURF features were identified as an appropriate choice of descriptors to use in our own approaches to Setting Detection.

In this chapter, we discuss these interest point detectors in more detail, outlining the merits of both approaches considered. In Section 4.2, we present a brief overview of interest point detectors. The purpose of this overview is not to provide a comprehensive review of the entire body of work in this area. Instead, this section is presented to provide some context for the introduction of the two descriptors in which we are most interested in this work, namely the SIFT and SURF descriptors. In Sections 4.2.2 and 4.2.3, we describe both of these descriptors in more detail.

We also present an overview of the two algorithms used to perform Setting Detection (Section 4.3). Both techniques are described in detail and our experimental results are presented in Sections

4.4 and 4.5. This represents the first major contribution of this work. One of our main objectives, as described in Section 1.3, is to investigate a new approach to assist in the management and organisation of visual content in a Visual Diary application. Setting Detection has been targeted as a useful approach in this regard, and the robustness of the two approaches presented is examined under a variety of scenarios. In each scenario, each of the main parameters incorporated within the proposed techniques are rigorously examined, demonstrating the utility, or otherwise, of the proposed approaches.

4.2 Interest Point Detectors

An interest point is simply any point in the image for which the signal changes two dimensionally [84]. The simplest local feature is a point in an image that is distinct from its neighbours. Interest points can be caused by a number of local properties including lighting, texture, and structure. There has been a great deal of research in recent years on local feature detection in order to derive an algorithm that is robust, efficient, and repeatable across multiple views of the same scene. Figure 4.1 shows an example of general interest points detected on a SenseCam image.

In Chapter 3, we discussed how local features are well suited to tasks such as setting detection in visual lifelog images. These detectors extract salient image features, which are distinctive in their neighbourhood and are reproduced in corresponding images in a similar way (i.e they are repeatable, in that they can detect the same point independently in different images). At the same time, interest operators supply one or more characteristics which can be used to perform image matching. Interest points generally correspond to physical corners in the scene, such as L-corners, T-junctions, and Y-junctions; but also to black dots on white backgrounds (in fact any location with significant 2D texture) [84]. Haralick et al. [196] define criteria for an optimal interest point detector:

- Distinctness: An interest point should stand out clearly against the background and be unique in its neighbourhood.
- Invariance: The determination should be independent of the geometric and photometric distortions.
- Stability: The selection of interest points should be robust to noise.

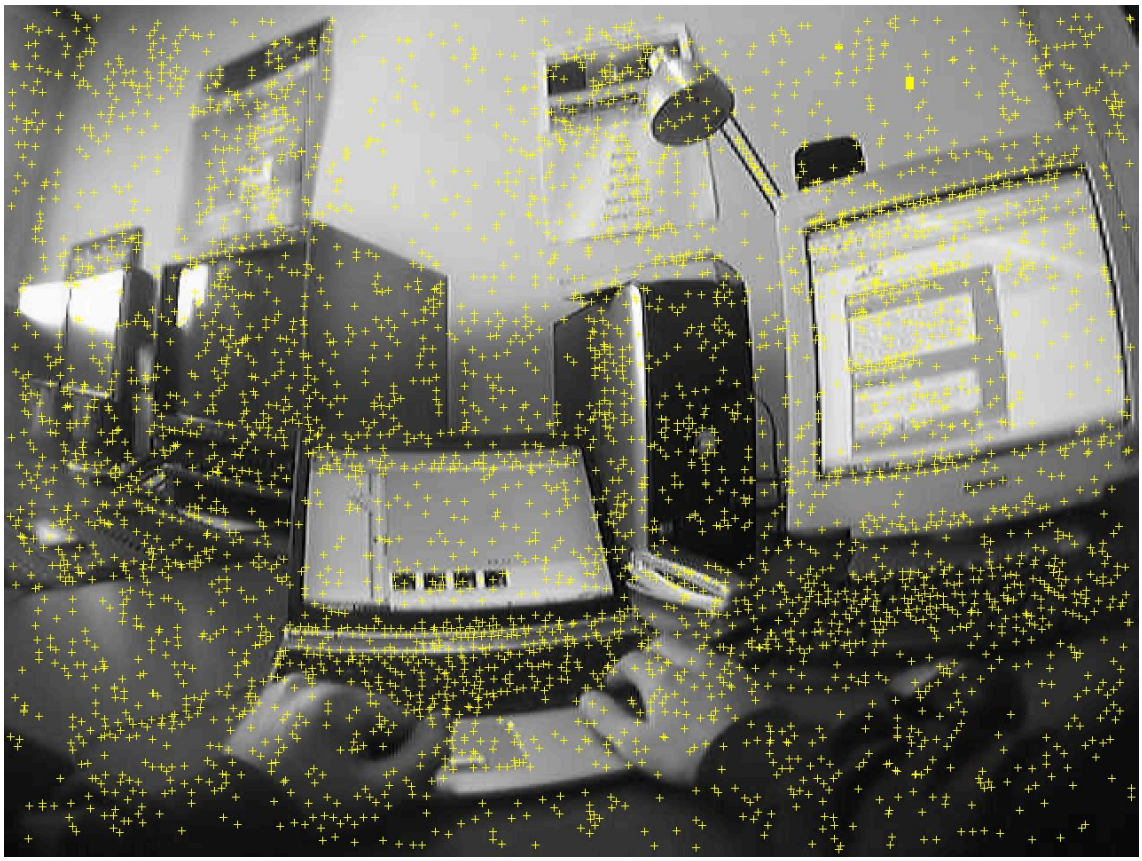


Figure 4.1: Interest points detected on a sample SenseCam image. The detector used was the Difference of Gaussian detector as part of Lowe's SIFT algorithm [83]. 896 points have been detected.

- Uniqueness: Apart from local distinctiveness, an interest point should also possess a global uniqueness, in order to improve the distinction of repetitive patterns.

- Interpretability: Interest values should have a significant meaning, so that they can be used for correspondence analysis and higher image interpretation.

These properties, if achieved in practice, make interest points very successful in the context of feature based image matching. While the characteristics of distinctiveness, invariance, and stability define the main requirements of an interest point operator, the characteristics of uniqueness and interpretability intensify the meaning of the term *interesting*.

4.2.1 Corner Detectors

Historically, the notion of interest points goes back to the earlier notion of corner detection, where corner features were detected with the primary goal of obtaining robust, stable, and well-defined image features for object tracking and recognition. In practice, however, most corner detectors

are sensitive not specifically to corners, but to local image regions which have a high degree of variation in all directions. The use of interest points also harks back to the notion of regions of interest, which have been used to signal the presence of objects, often formulated in terms of the output of a blob detection step [197].

One of the first interest operators was developed by Moravec for stereo vision control of an autonomous vehicle [198]. The goal of this interest operator is to identify a selection of points that are relatively uniform across the image. A measure of the variance of the image in four directions is used to select the points that have the maximum variance in a local neighbourhood. As pointed out by Moravec, one of the main problems with this operator is that it is not isotropic: if an edge is present that is not in the direction of the neighbours, then it will not be detected as an interest point.

Due to its simplicity and speed, the Harris detector is often used in practice [149]. The image gradient is computed over the entire image. About each point, a small window is used to construct a correlation matrix between both components of the image gradient. The eigenvalues of the resulting 2×2 matrix are calculated. If both eigenvalues are large, this means there are significant changes in two orthogonal directions, which implies a corner. Likewise, if only one eigenvalue is dominant, then a corner is not identified. Mikolajczyk et al. present a corner detector that identifies interest points along with an affine invariant neighborhood [199]. A multiscale Harris corner detector is used to determine initial interest points. For each identified point, its second moment matrix across scale is used to determine the correct position, scale, and space of the interest points. The method works well for interest points identified inside of flat planes with high information content. This technique has been further refined to be scale invariant [200]. A comprehensive overview of the current methods for the extraction of feature points performed by Schmid et al. found that the Harris operator was the most stable of all [84]. However, a more recent review and evaluation contradicted these findings where the authors found the F*ö*rstner operator obtained the best results with regard to distinctness, invariance, stability, uniqueness, and interpretability [201].

Two primary criteria for the evaluation of interest point detectors are localisation accuracy and repeatability [202, 84]. Localisation accuracy measures the deviation of the identified corner from the true centre. Repeatability measures the detection of identical features in the same scene taken from different views. However, with the growing interest in wide baseline stereo [203, 174], there has been an increased need for interest point detectors that are invariant to large deviations

resulting from affine, rotation, scale, and other similar distortions. A number of descriptors have been proposed to deal with these distortions, such as shape context [139], steerable filters [204], PCA-SIFT [152], differential invariants [205], spin images [172], SIFT [148], SURF [130], complex filters [177], moment invariants [206], and cross-correlation for different types of interest regions. Mikolajczyk & Schmid compare all of these descriptors, except SURF, for different types of interest points [128]. They observed that SIFT descriptors perform best and steerable filters come second. PCA-SIFT was found to be less distinctive than SIFT, and their proposed approach, gradient location and orientation histogram (GLOH), is more computationally expensive without yielding significant performance improvement. Regarding SURF, Bay et al. claim that SURF outperfoms SIFT, as well as a number of other descriptors [130]. Therefore, in the following sections, we focus on the SIFT and SURF descriptors. A detailed description of both descriptors can be found in Appendix E.

4.2.2 Scale Invariant Feature Transform (SIFT)

SIFT detects interest point locations and also extracts features from around the points that can be used to perform reliable matching between different views of an object or scene [83]. The SIFT features are invariant to image orientation and image scale, and provide robust matching across a substantial range of affine distortions, changes in 3D viewpoint, addition of noise, and changes in illumination. In addition to these properties, they are highly distinctive, relatively easy to extract, allow for correct object identification with low probability of mismatch, and are easy to match against a large database of local features. They are also robust to occlusion; as few as three SIFT features from an object are enough to compute its location and pose. In addition to object recognition, the SIFT features can be used for matching, which is useful for tracking and 3D scene reconstruction. Recognition can be performed in close-to-real time for small databases on modern computer hardware. The calculation of the features occurs in a multiphased filtering process that discovers interest points in scale space. Keypoints are generated which account for the local geometric deformations by characterising blurred image gradients in numerous orientation planes and at various scales [148]. For image matching, SIFT features are first extracted from a set of reference images and stored in a database. A new image is matched by individually comparing each feature from the new image to this previous database and finding candidate matching features based on Euclidean distance of their feature vectors. The algorithm produces a large number of keypoints that allow robust object recognition in cluttered or partially occluded images [148].

Prior to the SIFT algorithm, local feature generation was not invariant to scale and had a greater sensitivity to affine transformations, and rotation. Extraction of keypoints is performed using the following four steps:

1. Scale-space extrema detection: This stage successively blurs the images convolved with a Gaussian kernel of increasing variance. The Difference-of-Gaussian function is computed from the resulting octave of blurred images. From the Difference-of-Gaussian the potential interest points are identified using a corner detection threshold [83].

2. Keypoint localisation: This step begins by using a quadratic least squares fit to refine the location of the detected extrema. Keypoint candidates are chosen from the extrema in scale space, and keypoints are selected based on measures of their stability [83].

3. Orientation assignment: One or more orientations may be assigned to each keypoint based on local image gradient directions. With the orientation assigned, all future operations are performed on image data that has been transformed relative to the assigned orientation, scale, and location for each feature. This provides invariance to these transformations [83].

4. Keypoint descriptor: Local image gradients are measured at the selected scale in a 16×16 pixel patch around each keypoint. This information is transformed into vectors of 128 elements that allow significant levels of local shape distortion and change in illumination [83].

4.2.3 Speeded Up Robust Features (SURF)

SURF, introduced by Bay et al., is a robust image descriptor which can also be used to perform reliable image matching [130]. It attempts to improve the efficiency of SIFT by combining a Fast-Hessian detector together with a descriptor based on the distribution of Haar wavelet responses limited to 64 dimensions. The speed of the SURF algorithm arises mainly from the concept of integral images, introduced by Viola et al., where the time needed to compute the SURF keypoints are reduced significantly by convolving the image with large box filters [144]. Experimental results showed that SURF outperformed the current state of the art (SIFT and GLOH as well as many others reviewed in [128]) in terms of recognition accuracy and speed for image retrieval applications [130].

4.3 Setting Detection

Matching reliability is an issue with a large database of keypoints. The simplest approach to take involves calculating the Euclidean distance between the descriptor vectors of keypoints to determine if there is a possible match. However, many of these initial matches will be incorrect due to ambiguous features or features that arise from background clutter. To increase robustness, an approach is often used where matches are rejected for those keypoints for which the ratio of the nearest neighbour distance to the second nearest neighbour distance is greater than 0.8. This discards many of the false matches arising from background clutter, however, many false matches can still remain. In a large collection of images, such as a visual lifelog, the performance of this simple algorithm degrades rapidly. More complex approaches are necessary, and in the following sections, we outline an initial baseline approach, followed by the two main approaches to setting detection developed in this thesis.

4.3.1 Evaluation Metrics

In order to evaluate the performance of the developed approaches, a number of performance measures have been used. The main methods used are precision / recall values and an overall classification error for each individual experiment. Precision is the fraction of a search output that is relevant for a particular query. Therefore, its calculation requires knowledge of the relevant and non-relevant hits in the evaluated set of images. Thus, it is possible to calculate absolute precision of each algorithm which provides an indication of the relevance of the system. In this context precision is defined as:

$$\frac{Relevant\ \ Retrieved}{All\ \ Retrieved} \tag{4.1}$$

Recall, on the other hand, is the ability of a retrieval system to obtain all or most of the relevant images in the collection. Thus, it requires knowledge not just of the relevant and retrieved images, but also of those not retrieved [207]. The recall value is defined as:

$$\frac{Relevant\ \ Retrieved}{Relevant\ \ in\ \ Collection} \tag{4.2}$$

Precision / Recall values for all experiments are presented in Appendix A. This provides a detailed breakdown of the results for each user. For clarity, these tables are not included in this section, due to the large volume of information presented. Instead, we present the classification

error for each experiment in the following sections. This gives an overall indication of the performance of each variation of the algorithms used. This is calculated as follows:

$$1 - (\frac{Relevant\ \ Retrieved}{Relevant\ \ in\ \ Collection}) \tag{4.3}$$

4.3.2 Baseline Algorithm

An initial baseline algorithm was developed in order to provide a reference point with which to judge the results obtained using the two algorithms developed for setting detection in this thesis. In the first step of this approach, the user reorganises their SenseCam images to represent real settings. This is performed using a simple annotation tool (see Figure 4.3) which allows the user to update the setting information for each image. Once the training data has been organised into distinct settings, keypoints are extracted for each individual setting using the SIFT and SURF descriptors. SIFT and SURF keypoints are also extracted from each individual image in the test database. The X-means algorithm (an unsupervised variant of Kmeans) is used to perform the clustering of the keypoints extracted from the settings selected from the training database.

4.3.2.1 Xmeans Algorithm

X-means is an extension of the K-means algorithm, where not only the position of the centres, but also the optimal number of clusters is estimated [208]. The algorithm starts with a small number of clusters (e.g. 3 centroids in Figure 4.2 below). Each cluster centre is split into two child-points, to which the data-points belonging to the parent-point are distributed using a local K-means (with $k = 2$). The decision that has to be made is whether to replace the parent-point with the two child-points or not (i.e. the decision estimates whether the child-points model the structure of the data better than the original parent model). The criterion we use to determine which model provides a better representation is the Bayesian Information Criterion (BIC), although other kinds of probabilistic reasoning criterions may be used [209]. The BIC measure is based on the maximisation of a log-likelihood score [210]. For a given model M, the BIC measure is given by

$$BIC(M) = l(D) - P/2 \bullet (logR) \tag{4.4}$$

where l is the log-likelihood of the data according to model M taken at the maximum likelihood point, and P is the effective number of parameters [208]. For finite samples, BIC often chooses

simple models to avoid placing a heavy penalty on complexity. As the size of samples increases, the probability that the BIC measure favours the correct model also increases [210].

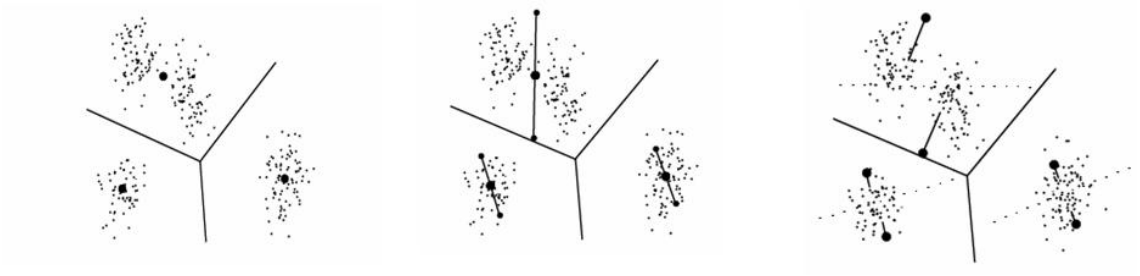


Figure 4.2: Operation of X-means algorithm

4.3.2.2 Image Signatures and the Earth Mover's Distance

After clustering the keypoints for each setting using X-means, we then generate an image signature, $\{(p_1,u_1),...,(p_m,u_m)\}$, where *m* is the number of clusters, p_i is the centre of the i^{th} cluster, and u_i is the relative size of the cluster (the number of descriptors in the cluster divided by the total number of descriptors extracted from the image [153]). Signatures have been introduced by Rubner et al. as representations suitable for matching using the Earth Mover's Distance (EMD) [211]. Numerous works have shown the EMD to be particularly effective for measuring similarity between image signatures [211, 153, 212].

The EMD is used to calculate the distance between signatures. It is defined as the minimum amount of work needed to change one signature into the other. The notion of work is based on a user-defined ground distance, which is the distance between two features. We use Euclidean distance as the ground distance. The EMD between two image signatures, *S1* :$\{(p_1,u_1),...,(p_m,u_m)\}$ and *S2* :$\{(q_1,w_1),...,(q_n,w_n)\}$, is defined as:

$$D(S_1, S_2) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{i,j} d(p_i, q_j)}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{i,j}}$$

where $f_{i,j}$ denote a set of flows that minimise the overall cost, and $d(p_i, q_j)$ is the ground distance between cluster centres p_i and q_j.

Computing the EMD is based on a solution to the well-known transportation problem [213]: Suppose that several suppliers, each with a given amount of goods, are required to supply several consumers, each with a given limited capacity. For each supplier-consumer pair, the cost of transporting a single unit of goods is given. The transportation problem is then to find a least-expensive

flow of goods from the suppliers to the consumers that satisfies the consumers' demands. Matching signatures can be naturally cast as a transportation problem by defining one signature as the supplier and the other as the consumer, and by setting the cost for a supplier-consumer pair to equal the ground distance between an element in the first signature and an element in the second. Intuitively, the solution is then the minimum amount of work required to transform one signature into the other.

For our application, the signature / EMD framework offers several advantages over the alternative histogram / χ^2 distance framework [171, 214]. A signature is more robust and descriptive than a histogram, and it avoids the quantisation and binning problems associated with histograms, especially in high dimensions [211] (recall that our SIFT, U-SURF64, and U-SURF128 descriptors are 128-, 64-, and 128-dimensional, respectively). The EMD has been shown to be (relatively) insensitive to the number of clusters, i.e., when one of the clusters is split during signature computation, replacing a single centre with two, the resulting EMD matrix is not greatly affected [215]. This is a very important property, since automatic selection of the number of clusters remains an unsolved problem. In addition, in several evaluations of colour and texture-based image retrieval [216, 211], EMD has performed better than other methods for comparing distributions, including χ^2 distance. Finally, the EMD / signature framework has the advantage of efficiency and modularity. It frees us from the necessity of clustering descriptors from all images together and computing a universal texton dictionary, which may not represent all texture classes equally well [214].

4.3.2.3 Experimental Results

The overall classification error for this approach can be seen in Table 4.1. The algorithm performed quite poorly in all cases with the lowest classification error being 0.6784 for *User 5* using SIFT features. However, it provides an initial baseline with which to judge the algorithms developed in this thesis, and also influences some of the design choices made within those algorithms. In the baseline implementation, we believe that the number of clusters produced by X-means did not provide enough discriminative power to sufficiently model the settings in question. For example, the X-means clustering returned only 17 clusters for settings containing several hundred thousand keypoints. It seems clear that the total number of cluster centres used to create each setting signature was insufficient to generate good matching results. Despite these problems, the X-means algorithm has been found to give good results when clustering SIFT keypoints and, in general terms, it has been found to provide superior results to those obtained using a range of k

User	SIFT Error	U-SURF64 Error	U-SURF128 Error
User 1	0.7085	0.7623	0.7967
User 2	0.7992	0.9146	0.9365
User 3	0.7182	0.8792	0.8892
User 4	0.6943	0.7384	0.7543
User 5	0.6784	0.7826	0.8149
Average	0.7197	0.8154	0.8383

Table 4.1: The classification error for all users for each descriptor using the baseline approach.

values and the K-means algorithm [217, 208]. However, the algorithm has several free parameters and the resulting number of clusters can vary greatly depending on the parameters used. In this work, the question over the number of clusters is addressed by using K-means clustering and by experimentally determining an optimum value for k (see Section 4.4.2). Indeed, the value for k obtained is validated by similar work in the literature [150].

4.3.3 Bag of Keypoints Algorithm

In order to perform setting detection, the first approach we describe is a method similar to that outlined by [150]. The basic idea is that a set of local image patches is sampled using some method (e.g. densely, randomly, using a keypoint detector, etc.) and a vector of visual descriptors is evaluated on each patch independently. The resulting distribution of descriptors in descriptor space is then vector quantised against a pre-specified codebook to convert it into a histogram of votes for codebook centres, and the resulting global descriptor vector is used as a characterisation of the image (e.g. as a feature vector on which to learn an image classification rule using a multi-class classifier).

The main steps used in this approach are as follows:

- The training images are annotated into pre-defined settings.
- Samples of multiple image patches are taken from each image.
- Patch feature vectors are extracted using the SIFT and SURF descriptors.
- Codebooks are generated with k-means clustering over the extracted patch feature vectors.
- All patch feature vectors are assigned to the nearest codebooks, and a set of patch feature vectors for each image are converted into one histogram vector of assigned codebooks.
- A multi-class classifier is trained with all the histogram vectors in the training data.

- All the histogram vectors of the test images are classified into the appropriate setting by applying the trained clasification rules.

This approach is designed to maximise classification accuracy while minimising computational effort. The vocabulary used should be large enough to distinguish relevant changes in image parts, but not so large as to distinguish irrelevant variations such as noise. Our goal is to use a vocabulary that allows good categorisation performance on a given training dataset. Each of these steps is described in more detail below.

4.3.3.1 Setting Annotation

In the first step of our approach, the user reorganises their SenseCam images to reflect the real settings depicted that are particularly important to him/her. This is performed using a simple annotation tool (see Figure 4.3), which allows the user to update the setting information for each image. The tool is simple and intuitive to use. The user can visually scan over all images very quickly, easily identifying collections of images which constitute an important setting. Note that we asked the user to provide an importance score between 0 (not very important) - 5 (very important) for each setting. This is used to facilitate the user in browsing and locating important settings in the Visual Diary application (discussed further in Section 4.4.1). The objective here is to provide the user with a low-overhead mechanism for organising his/her Visual Diary in terms of specific settings of interest. Given this user generated training data, we train a multi-class classifier using the bags of keypoints as feature vectors.

4.3.3.2 Feature Extraction

Similar to terms in a text document, an image has local interest points (or keypoints) defined as salient image patches (small regions) that contain rich local information of the image, usually around the corners and edges of image objects. In order to extract and describe these interest points, we use the SIFT and SURF descriptors, as discussed in Sections 4.2.2 & 4.2.3. However, a number of variations of the SURF descriptor exist, including the U-SURF descriptor discussed in Section 3.3. U-SURF, or 'Upright SURF' is a version of SURF with the rotation invariance left out. There are advantages to using this version of the SURF descriptor, also discussed in Section 3.3. In addition, an extended version of the SURF descriptor, to 128 dimensions, is also available. SURF-128 treats sums of d_x and $|dx|$ separately for $d_y < 0$ and $d_y > 0$. Similarly, the sums of

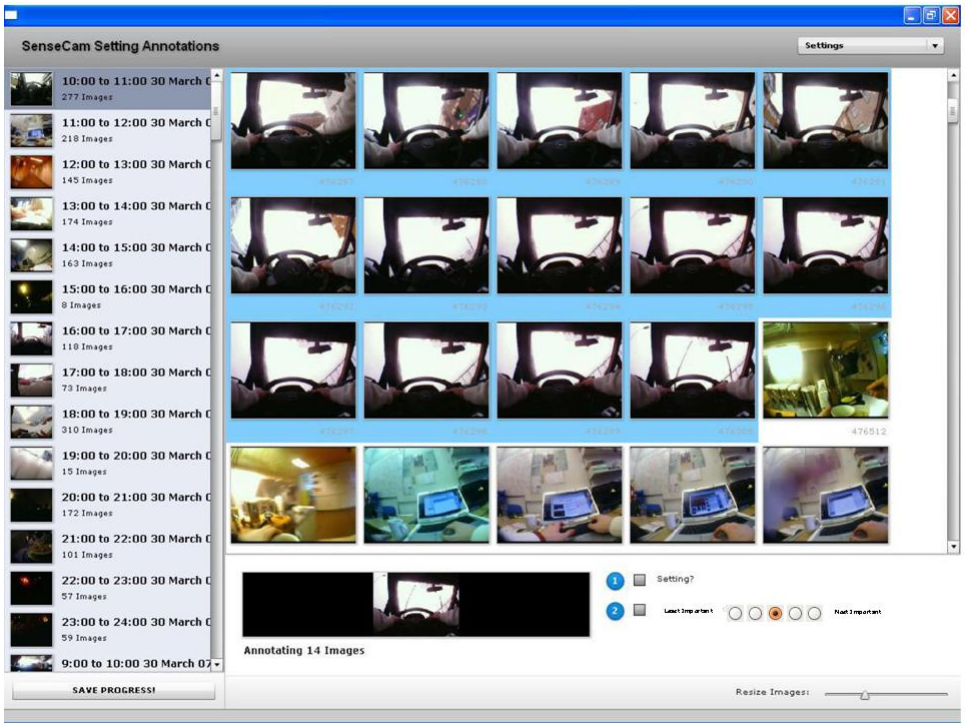


Figure 4.3: SenseCam Setting Annotation Tool. In this particular instance, the user is annotating a setting where the individual wearing the SenseCam is driving.

d_y and $|dy|$ are split up according to the sign of d_x. This doubles the number of features (128 instead of 64) resulting in a more distinctive descriptor, which is not much slower to compute, but slower to match due to its higher dimensionality (see Appendix E). For the purposes of this work, we focus on the SIFT, U-SURF64, and U-SURF128 descriptors. Since all our images are captured with the SenseCam, which the user wears around their neck, the requirement for rotation invariance is not relevant to our work. Hence, U-SURF is an appropriate version of the SURF descriptor to use for feature extraction. Features are extracted from all images using each of the three descriptors mentioned, in order to determine their utility for an application of this nature.

4.3.3.3 Visual Vocabulary Construction

Csurka et al. describe the construction of a visual vocabulary as a way of constructing a feature vector for classification that relates new descriptors in query images to descriptors previously seen in training [150]. An extreme example of this approach would be to compare each query descriptor to all of the training descriptors in the database. For most applications, this is not feasible due to the huge number of training and test descriptors involved and the large amount of processing time this would require.

Instead, we use the vector quantisation technique which clusters the keypoint descriptors in

their feature space into a large number of clusters using the K-means clustering algorithm [218], and encodes each keypoint by the index of the cluster to which it belongs. The algorithm proceeds by partitioning the input points into k initial sets, either at random or using some heuristic. It then calculates the centroid of each set and constructs a new partition by associating each point with the closest centroid. Then, the centroids are recalculated for the new clusters, and the algorithm repeated until convergence, which is obtained when the points no longer switch clusters (or alternatively centroids are no longer changed). The choice of K-means is justified because the Euclidean distance is meaningful in the SIFT-descriptor space. One problem with the K-means algorithm is that the number of clusters, k, is an input parameter. Methods do exist to facilitate the estimation of the number of clusters [208], however, in this scenario we are not really interested in a correct clustering in the sense of feature distributions, but rather in accurate categorisation into the correct settings. The choice of k used in this work is examined further in Section 4.4.2.

Each cluster generated is representative of a visual word which represents a specific local pattern shared by the keypoints in that cluster. The clustering process, therefore, generates a visual-word vocabulary which describes different local image patches in the images. The number of clusters generated via K-means clustering determines the size of the vocabulary, which can vary from hundreds to over tens of thousands. We can then represent each image in the data set as a histogram of visual words drawn from the vocabulary. This representation is analogous to the bag-of-words document representation in terms of form and semantics. Both representations are sparse and high-dimensional, and just as words convey meanings of a document, visual words reveal local patterns characteristic of the whole image.

The bag-of-keypoints representation can be converted into a visual-word vector similar to the term vector of a document. The visual-word vector may contain the presence or absence information of each visual word in the image, the count of each visual word (i.e., the number of keypoints in the corresponding cluster), or the count weighted by other factors. Visual-word vectors are used in our image classification approach. The process of generating a visual-word representation is illustrated in Figure 4.4.

4.3.3.4 Classification

Once descriptors have been assigned to clusters to form feature vectors, we can use different classification methods in the image descriptor space. The problem is effectively reduced to that of multi-class supervised learning, with as many classes as defined visual categories. Many different

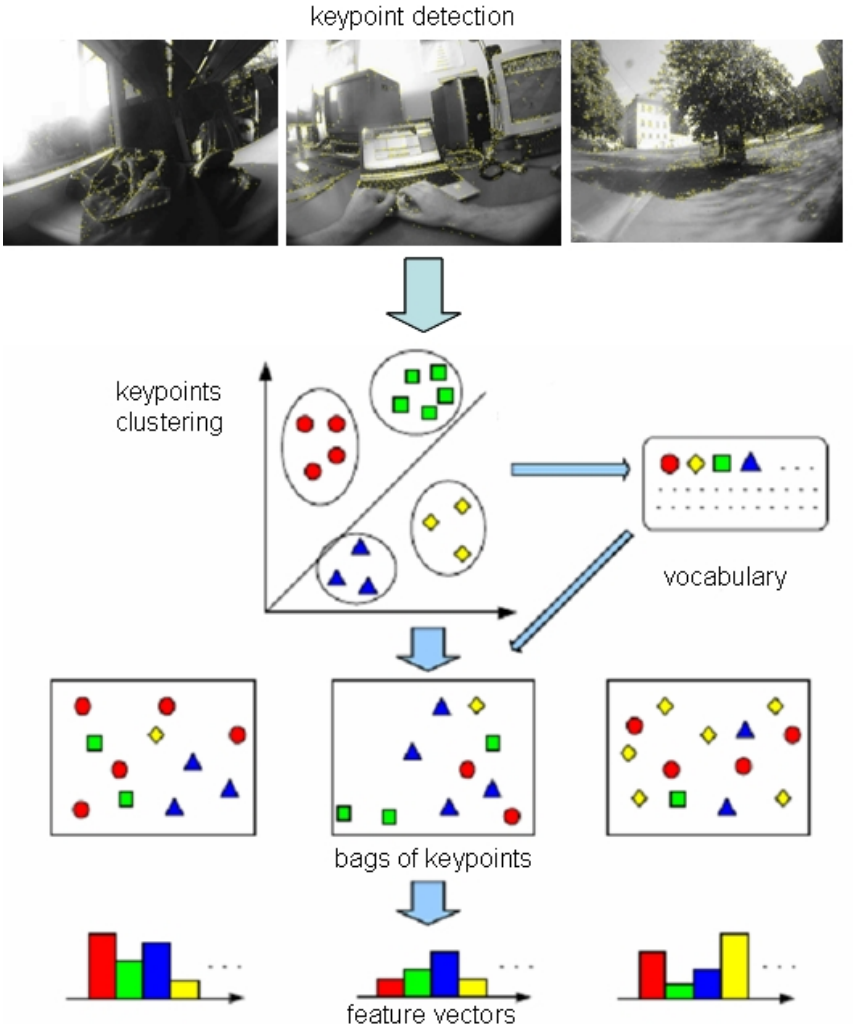


Figure 4.4: Visual-word image representation based on vector-quantised keypoint features

types of classifiers exist, however, a detailed analysis of them all is beyond the scope of this thesis. Therefore, we have chosen three classifiers which are representative of a broad range of classification algorithms. The algorithms we have chosen to use in this work are the K-Nearest Neighbour (KNN) classifier, the Multiclass Linear Perceptron (MLP), and a Support Vector Machine (SVM) [218]. KNN provides a good baseline approach, MLP is representative of artificial neural network type classifiers, and the SVM is representative of hyperplane style approaches.

In the KNN algorithm, a setting is classified by a majority vote of its neighbours, with the setting being assigned to the class most common amongst its k nearest neighbours. The neighbours are taken from the training data for which the correct classification is known. In order to identify neighbours, the settings are represented by position vectors in a multidimensional feature space. It is usual to use the Euclidean distance, though other distance measures, such as the Manhattan distance, could in principle be used instead.

The algorithm proceeds as follows: Given the training data, $D = x_1, ..., x_n$, as a set of n labeled examples, the nearest neighbour classifier assigns a test point, x, the label associated with its closest neighbour in D. In our work, this distance is measured using Euclidean distance. Given the distance function, the nearest neighbour classifier partitions the feature space into cells consisting of all points closer to a given training point than to any other training points. The K-Nearest Neighbour classifier classifies x by assigning it the label most frequently represented among the k nearest samples. In other words, a decision is made by examining the labels on the k-nearest neighbours and taking a vote (see Figure 4.5).

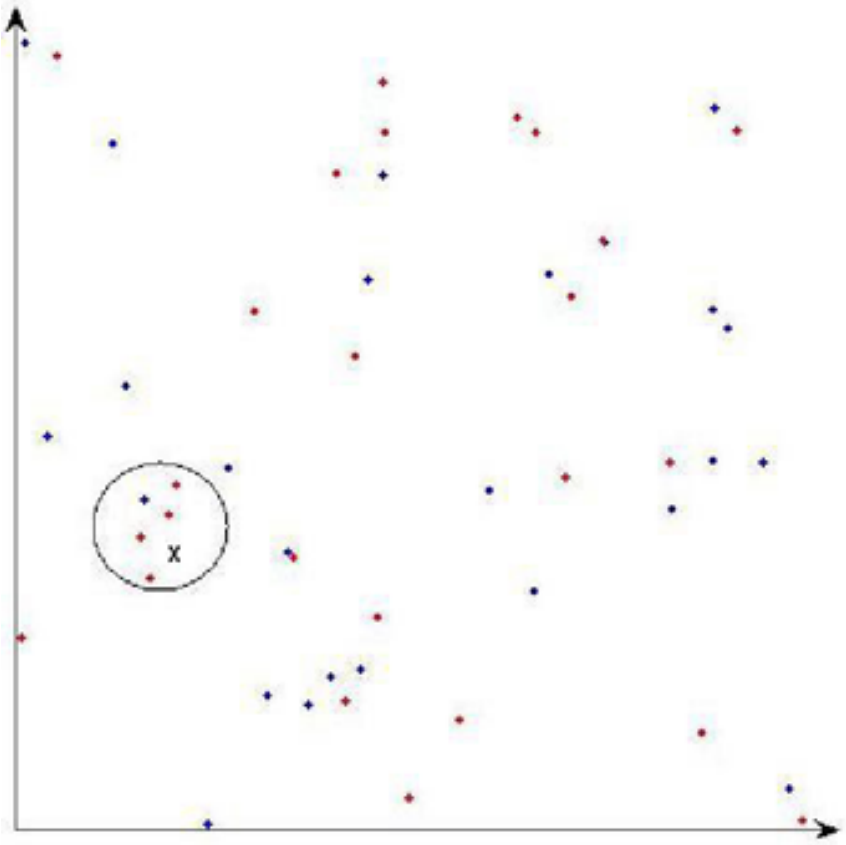


Figure 4.5: The K-nearest neighbour query forms a spherical region around the test point x until it encloses k training samples, and it labels the test point by a majority vote of these samples. In the case for $k = 5$, the test point will be labeled as red.

The Perceptron algorithm is a well studied and popular classification learning algorithm. Despite its age and simplicity it has proven to be quite effective in practical problems [219]. The Perceptron maintains a single hyperplane which separates positive instances from negative ones.

In [218], this binary perceptron algorithm was extended to construct a MLP algorithm consid-

ering all classes at once. In this case, c linear discriminant functions have to be defined, i.e.,

$$f_i(x) = w_i^T x + b_i \; i = 1, ..., c, \tag{4.5}$$

where w and b denote the weight vector and threshold of the i^{th} discriminant function. For some input vector x, if $f_i(x) > f_j(x)$ for all $j \neq i$, assign this vector to the i^{th} class.

For some training sample, x_q, assigned to the i^{th} class, if there is at least one $j \neq i$ for which $f_i(x_q) \leq f_j(x_q)$, this vector is referred to as a misclassified sample. In the multiclass linear perceptron algorithm, such a misclassified sample is used to modify some weight vectors and thresholds in equation 4.5 according to the learning rule:

$$\begin{aligned} w_i &\Leftarrow w_i + x_q, b_i \Leftarrow b_i + 1 \\ w_j &\Leftarrow w_j - x_q, b_j \Leftarrow b_j - 1 \end{aligned} \tag{4.6}$$

That is, the weight vector and threshold for the desired class is increased by the misclassified sample, the vector and threshold for the incorrectly chosen class is decreased, and all others are left unchanged. For multiclass classification problems, if the weight vectors and thresholds can classify all training samples correctly, this training set is said to be linearly separable. Based on Kesler's construction [218], this multiclass problem can be reduced to a binary one. Thus, its convergence for linearly separable cases can be proven using the perceptron convergence theorem for binary classification.

SVM's are a set of related supervised learning methods used for classification and regression [218]. For classification, SVM's operate by finding a hyperplane which separates two-class data with maximal margin [220] (see Figure 4.6). The margin is defined as the distance of the closest training point to the separating hyperplane. For given observations X, and corresponding labels Y, which take values ± 1, one finds a classification function $f(x) = sign(w^T x + b)$ where w and b, represent the parameters of the hyperplane. This hyperplane will attempt to split the positive examples from the negative examples. Viewing the input data as two sets of vectors in a n-dimensional space, an SVM will construct a separating hyperplane in that space, one which maximizes the margin between the two data sets. To calculate the margin, we construct two parallel hyperplanes, one on each side of the separating one, which are 'pushed up against' the two data sets. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the neighbouring datapoints of both classes. The hope is that, the larger the margin or distance

between these parallel hyperplanes, the better the generalisation error of the classifier will be. The split is chosen to have the largest distance from the hyperplane to the nearest of the positive and negative examples. Intuitively, this makes the classification correct for testing data that is near, but not identical, to the training data.

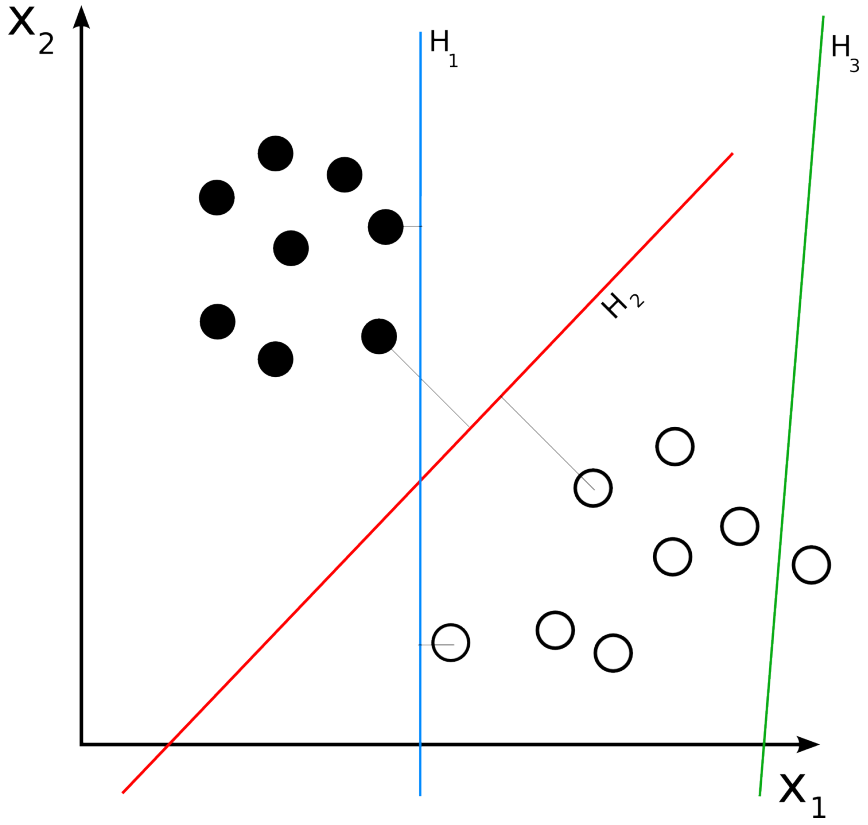


Figure 4.6: H3 doesn't separate the 2 classes. H1 does, with a small margin, and H2, with the maximum margin.

SVM's operate by preprocessing the data to represent the patterns in higher dimensions than the orignial feature space. Using a nonlinear mapping, $\varphi(\cdot)$, the data is projected to a higher dimensional space. Data from two categories can be better separated by a hyperplane once the data has been projected to a sufficiently high dimension.

Data sets are not always linearly separable. The SVM takes two approaches to this problem: Firstly it introduces an error weighting constant, C, which penalises misclassification of samples in proportion to their distance from the classification boundary; Secondly, a mapping, φ, is made from the original data space of X to another feature space. This second feature space may have a high or even infinite dimension. One of the advantages of the SVM is that it can be formulated entirely in terms of scalar products in the second feature space, by introducing the kernel $K(u, v) = \varphi(u) \cdot \varphi(v)$. Both the kernel, K and penalty C are problem dependent and need to be

determined by the user.

In the kernel formulation, the decision function can be expressed as:

$$f(x) = sign(\sum \underline{i} y_i \alpha_i K(x, x_i) + b) \tag{4.7}$$

where x_i are the training features from the data space, X, and y_i is the label of x_i. Here, the parameters, α_i, are typically zero for most i. Equivalently, the sum can be taken only over a select few of the x_i. These feature vectors are known as support vectors. It can be shown that the support vectors are those feature vectors lying nearest to the separating hyperplane. In our case, the input features x_i are the binned histograms formed by the number of occurrences of each keypoint v_i from the vocabulary, V, in the image, I_i.

SVM's are inherently two-class classifiers, however, in [221] they were extended to the multi-class case used in these experiments. The traditional way to perform multiclass classification with SVM's is to reduce a single multiclass problem into multiple binary problems. For instance, a common method is to build a set of binary classifiers where each classifier distinguishes between one of the labels to the rest (commonly referred to as one-versus-all or OVA classification). In this situation, the chosen class is the one which classifies the test data with greatest margin. Another strategy is to build a set of one-versus-one classifiers, and to choose the class that is selected by the most classifiers. While this involves building multiple classifiers (dependent on the number of classes), the time for training classifiers may actually decrease, since the training data set for each classifier is much smaller. However, while these methods provide a simple and powerful framework, they cannot capture correlations between the different classes since they break a multiclass problem into multiple independent binary problems.

A better alternative, and certainly one more aligned with Vapnik's principle of always trying to solve a problem directly, is provided by the construction of multiclass SVM's which consider all classes at once [220]:

$$\min \ \frac{1}{2} \sum_{m=1}^{k} (w_m \cdot w_m) + C \sum_{i=1}^{l} \sum_{m=1}^{k} \xi_i^m \tag{4.8}$$

$$s.t. \ (w_{z_i} \cdot x_i) + b_{z_i} \geq (w_m \cdot x_i) + b_m + 2 - \xi_i^m, \quad \xi_i^m \geq 0 \tag{4.9}$$

where z_i contains the index of the class x_i belongs to, and w_m and b_m are the weight coefficients

and bias term of the separating hyperplane for class m. This gives the decision function:

$$f(x) = arg \;\; max_n \left[(w_n \cdot x) + b_n\right] \tag{4.10}$$

The objective function of Equation 4.8 is also composed of the two regularisation and classification error terms. The regularisation term tries to minimise the norm of all separating hyperplanes simultaneously. The classification errors for each class are treated equally and their sum is added to the objective function. There are also modifications to this approach by using different values for errors of different classes according to some loss criteria or prior probabilities. The constraint of Equation 4.10 aims to place each instance on the negative side of the separating hyperplane for all classes except the one it belongs to. The solution to this optimisation problem is given by the decision function [221]:

$$f(x) = arg \;\; max_n \left[\sum_{i=1}^{l}(c_i^n A_i - \alpha_i^n)(x_i \cdot x) + b_n\right] \tag{4.11}$$

4.3.4 Alternative Approach

The first step in this approach is to organise the training data to represent real settings. As in the previous approach, this step is performed by the user using the annotation tool. Once the training data has been organised into distinct settings, local image patches are extracted from each setting independently. Test vectors of visual descriptors are then evaluated on the patches from each individual setting. The distance between the test descriptors and the local patches extracted from each individual setting is then calculated. The test image is deemed to belong to the setting where this distance is minimal. The main steps used in this second approach are as follows:

- The images are annotated into pre-defined settings.
- Samples of multiple image patches are taken from the images contained in each setting.
- Patch feature vectors are extracted from all the points using the SIFT and SURF descriptors.
- Codebooks are generated with k-means clustering over the extracted patch feature vectors for the training images of each setting.
- An image signature is generated for all test images and the distance to each setting is calculated.

User	Total number of images
User 1	28105
User 2	80934
User 3	35634
User 4	18467
User 5	44440

Table 4.2: Total number of images captured by each of the five users.

- All test images are classified into the appropriate setting based on their distance from each setting.

This approach is similar to the first approach, however, the key differences are that the keypoints for each setting are clustered independently (as opposed to creating a bag-of-keypoints for all settings), and an image signature and distance measure is used to determine the class a particular image belongs to (as opposed to using a multiclass classifier). The annotation of images into settings, the extraction of features, and a description of K-means clustering, have already been outlined in Sections 4.3.3.1, 4.3.3.2, & 4.3.3.3. Regarding feature extraction, we use the same features in this approach as the previous one (i.e. SIFT, U-SURF64 and U-SURF128). After clustering the keypoints for each test image using K-means, we then generate an image signature and calculate the distance between signatures using the Earth Mover's Distance (EMD) [211] (see Section 4.3.2.2).

4.4 Experiments

The experiments were carried out on SenseCam images gathered by five different users over different periods of time. The total number of images in the collection is 207,580. The number of images captured by each individual user can be seen in Table 4.2. The variation in the size is due to the length of time each user captured images for these experiments. This ranged from a minimum of 6 days to a maximum of 1 month.

Both approaches described in Section 4.3 were applied to each of these image collections. The experiments were carried out three times, once for each feature descriptor, for each of the described approaches, giving a total of six different experiments for each of the five users. This gives a total of thirty separate experiments.

User	Total number of images annotated	Total number of settings
User 1	8858	24
User 2	17771	42
User 3	7751	20
User 4	9005	16
User 5	24897	10

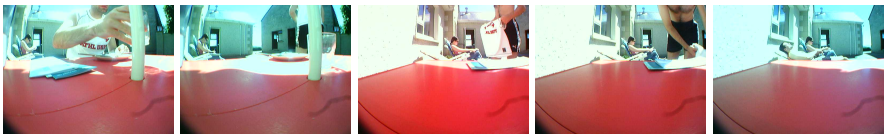
Table 4.3: Total number of images and settings annotated by each of the five users.

4.4.1 Data Annotation

The first step in each approach was to use the annotation tool to generate training data. Each user classified their own images into different settings using the annotation tool. The user was not given any strict instructions as to how they should perform the annotation. The concept of a Visual Diary and the definition of a setting was explained to them and it was left up to him/her to judge what they considered an important setting to be in this context. As a result, the variation in annotations across users was significant. Some users returned relatively few settings in the annotation process, whilst others annotated significantly more images. This highlights the difficulty in using such approaches. Despite providing the same definition of what constituted a setting to each user, each naturally interprets the information slightly differently, and performs annotations accordingly. Thus, some users have a significantly larger number of annotated settings compared to others. Naturally, this presents challenges for our algorithms, but it's important to model real user behaviour and the annotation process allows us to achieve this. The total number of images annotated for each user, and the total number of settings found, can be seen in Table 4.3. Sample images from two settings for each user can be seen in Figure 4.7.

4.4.2 System parameters

In order to determine the optimal parameters to use in each approach, a preprocessing step was first carried out. A number of parameters exist which need to be examined in order to obtain the best results. With a bag-of-keypoints approach, we are faced with a number of implementation choices. These include how to sample image patches, what visual patch descriptor to use, and how to classify images based on the resulting global image descriptor. In this work, we have used the SIFT, U-SURF64, and U-SURF128 features to sample and describe the image patches, and have discussed the reasons behind this choice in detail in Section 3.3 and Section 4.3.3.2. Regarding the choice of classifiers, we opted for three classifers which were representative of a broad range

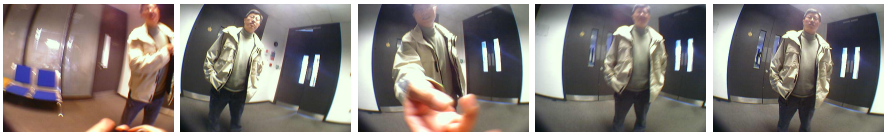


(a) User 1 - Sitting in the sun

(b) User 1 - Eating breakfast

(c) User 2 - At a restaurant

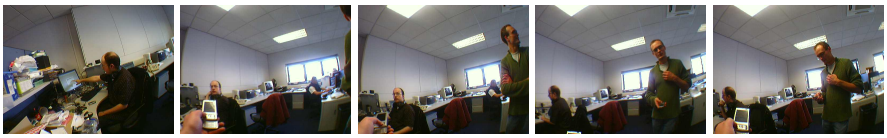
(d) User 2 - At work



(e) User 3 - Chatting with a friend

(f) User 3 - At a meeting

(g) User 4 - At a meeting



(h) User 4 - Chatting with a colleague

(i) User 5 - Having dinner

(j) User 5 - Reading at home

Figure 4.7: Sample Images from 2 settings for each of the 5 users

SVM Kernel	Classification Error	Running Time
Linear	0.0119	2.75
Polynomial	0.0238	43484.42
Radial Basis	0.9286	0.16
Sigmoid	0.9286	58.45

Table 4.4: The classification error and running time (seconds) using different SVM kernels with the SIFT descriptor.

SVM Kernel	Classification Error	Running Time
Linear	0	12.64
Polynomial	0.0476	643925.34
Radial Basis	0.9286	0.08
Sigmoid	0.9524	389.11

Table 4.5: The classification error and running time (seconds) using different SVM kernels with the U-SURF64 descriptor.

of classification algorithms. The three classifers used are KNN, MLP, and SVM, and we discussed this choice in more detail in Section 4.3.3.4.

However, with the SVM classifier, a number of parameters and kernels exist which can potentially influence the final results. In order to determine the optimal SVM parameters to use in these experiments, the overall classification error was calculated for different combinations of parameters in order to determine the optimum setup. The version of SVM we used was a version of SVM-Light known as SVM-Multiclass [221]. A choice of kernels are available in this algorithm, including linear, polynomial, radial basis, and sigmoid. Each kernel was evaluated using default parameter settings in order to determine the most appropriate choice. The results can be seen in Tables 4.4, 4.5, & 4.6.

It's clear from these results that the sigmoid and radial basis kernels are not an appropriate choice, both yielding high error rates. The performance of the polynomial and linear kernels was similar in terms of classification error. However, the running time of the polynomial kernel was in the order of several hours for the SIFT descriptor, and several days for both the U-SURF64 and U-SURF128 descriptors. Clearly, this is not acceptable, particularly when the linear kernel achieves similar performance levels in a matter of seconds. Regarding the choice of c (the trade off between the training error and margin), the default choice of 0.01 was used in the analysis above. In order to determine the appropriate choice of c to use in these experiments, we analysed the classification error as a function of different values of c using the linear kernel only. The results of this analysis indicated that a value of $c = 0.05$ was the most appropriate. Values were tested from the default value up to $c = 0.05$ for each of the three descriptors tested. For the SIFT descriptor, the classifi-

SVM Kernel	Classification Error	Running Time
Linear	0.0238	5.61
Polynomial	0.0714	718811.51
Radial Basis	0.9286	0.11
Sigmoid	0.9524	369.58

Table 4.6: The classification error and running time (seconds) using different SVM kernels with the U-SURF128 descriptor.

cation error remained the same for each value (0.0119), and the running time decreased from 2.75 seconds to 0.95 seconds. Similar results were observed for the U-SURF64 (classification error of 0 and runtime reduced from 12.64 to 5.09 seconds) and U-SURF128 (classification error of 0.0238 and running time reduced from 5.61 to 5.53 seconds). The difference in the classifcation error was thus negligible. The only difference was in the running time of the classifier, but again the differences here were negligible.

Another issue which can impact performance is the size of the visual-word vocabulary. This is controlled by the number of clusters generated. Two contradictory considerations are at work here – the discriminative nature of the descriptor versus it's ability to generalise – so choosing the right vocabulary size involves a trade-off. With a small vocabulary, the visual-word feature is not very discriminative because dissimilar keypoints can map to the same visual word. As the vocabulary size increases, the feature becomes more discriminative, but also less generalisable and forgiving to noise, since similar keypoints can map to different visual words. Using a large vocabulary also increases the cost of clustering keypoints, computing visual-word features, and running supervised classifiers. There is no consensus as to the appropriate size of a visual-word vocabulary. The vocabulary size used in existing works varies from several hundred [153], to thousands and tens of thousands [222]. Csurka et al. [150] found no significant improvement in performance as they moved from $k = 1000$ to $k = 2500$, so they used $k = 1000$ as it provided a good trade off between speed and accuracy. However, it is difficult to directly compare different methods due to the different test corpus and classification methods used.

In these experiments, k was evaluated on a subset of the images for a single user's image collection. A range of values for k were analysed, and the value which minimised the error across all three classifers, and across all three descriptors, was deemed to be an appropriate value to use for all experiments. The results of our analysis for each descriptor can be seen in Figures 4.8, 4.9, and 4.10.

The error rate for the MLP classifier was somewhat erratic across all values of k for all three

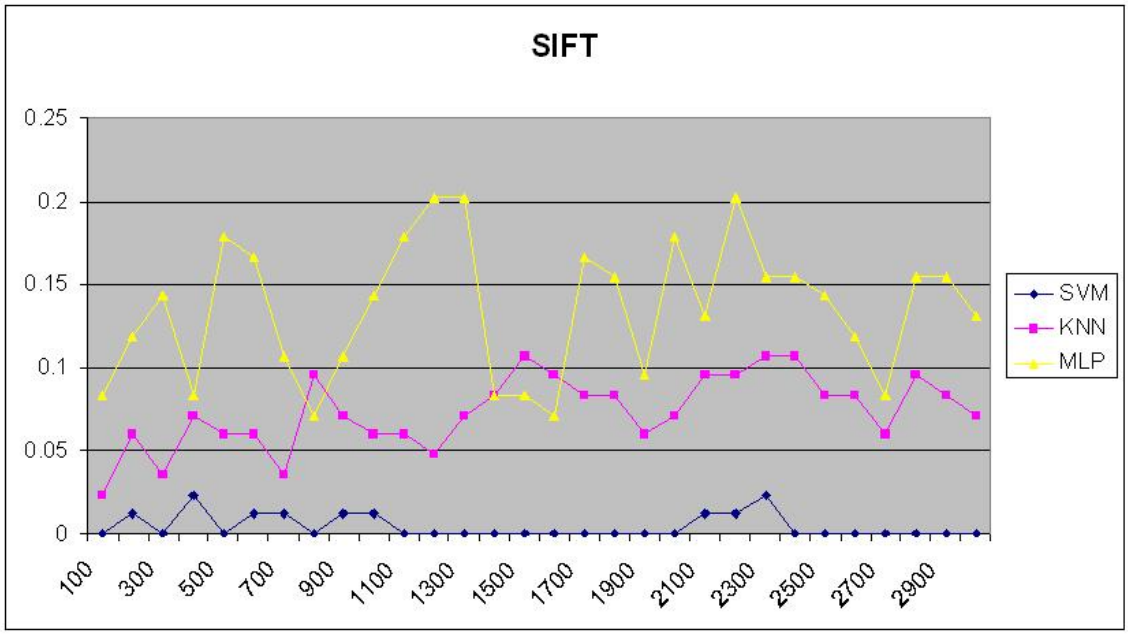


Figure 4.8: The overall error rate found for different choices of k for all three classifiers using SIFT.

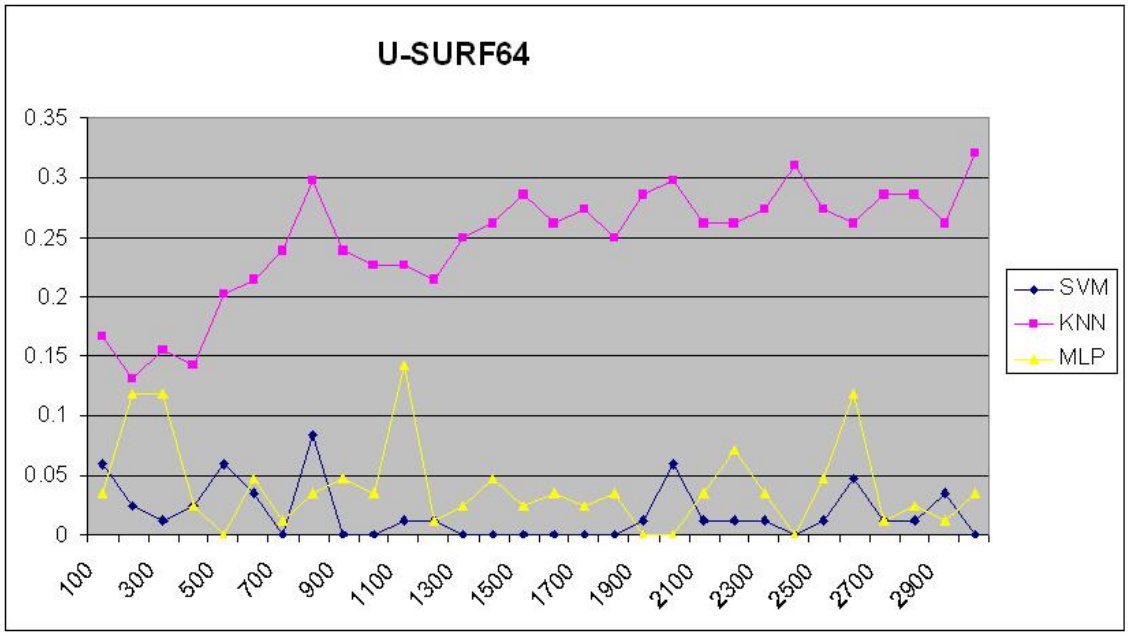


Figure 4.9: The overall error rate found for different choices of k for all three classifiers using U-SURF64.

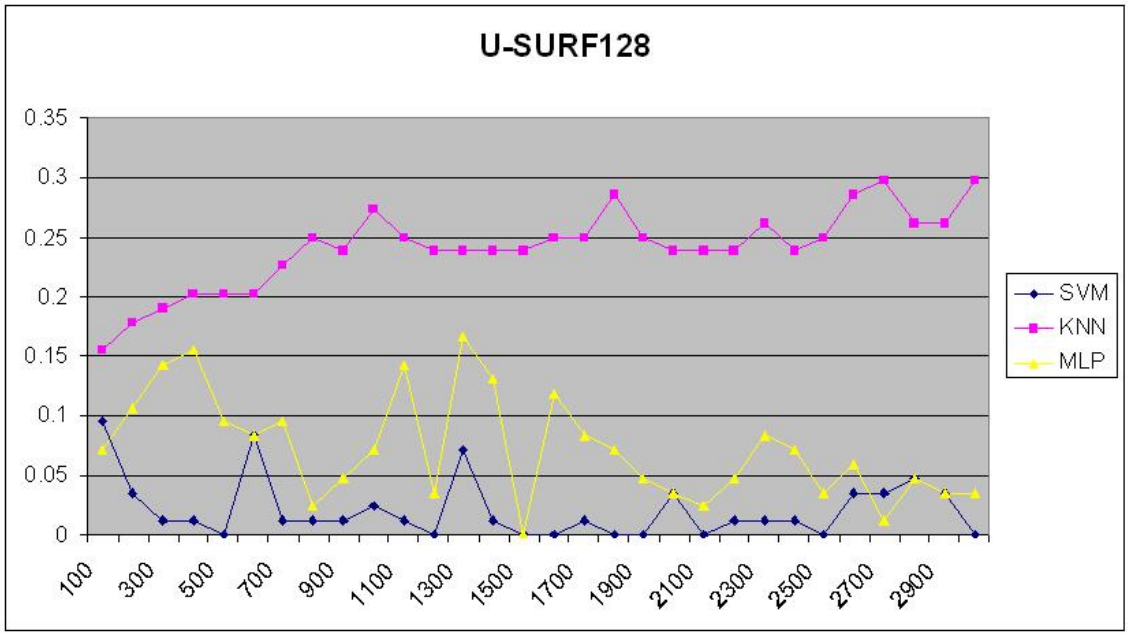


Figure 4.10: The overall error rate found for different choices of k for all three classifiers using U-SURF128.

descriptors, so it is difficult to determine an appropriate value using this classifer. Using the KNN classifer, the error rate tended to climb using all three descriptors. Although the error rates using KNN were minimal at low values of k, the error rates were still higher than those achieved using the other two classifiers, so values of k at this level were not taken into consideration. Previous research also indicates this - to the best of our knowledge values of $k = 100$ have not achieved acceptable results with a large data set across numerous classes. The performance of the SVM was more stable: although it was still somewhat erratic, particularly with the U-SURF128 descriptor, the overall error rates remained low. At values around $k = 1000$, the performance of the SVM using the SIFT and U-SURF64 descriptors became relatively stable. For this reason, and based on previous work in this area, we chose a value of $k = 1000$ for the remainder of our experiments. It is worth reiterating that we are not looking for a value of k which will provide a 'correct' clustering of the data. We are merely trying to find an appropriate value to provide enough discriminative power, without significantly increasing the computation time.

The final choice made in this approach relates to the distance measure used. We have chosen to use the Euclidean distance based on the significant work in the literature which uses this particular measure to determine the distance between keypoint descriptors [223, 224, 225]. In particular, the evaluation performed by Mikolajczyk and Schmid [128], as well as the work on SIFT and SURF described by Lowe [83] and Bay et al. [130] respectively, indicates that the Euclidean distance is

the most appropriate choice when working with these particular descriptors.

With our second approach, a number of similar choices must be made in order to determine the optimum system parameters. Regarding the choice of k used for clustering the keypoints in each individual setting, it was felt that $k = 1000$, as determined above, was the most appropriate choice. The clustering process undertaken here is similar to that performed in the bag-of-keypoints approach. The analysis performed above, along with the work quoted from the literature, justifies this choice.

A second value of k needs to be determined in order to cluster the keypoints of each individual test image. There is little in the literature concerning the determination of k for clustering keypoints in an individual image. Most of the works quoted above are more focused on determining an appropriate value of k when clustering millions of keypoints. However, Zhang et al. [153] extracted 40 clusters per image using K-means clustering. They also found no significant improvement when moving from $k = 40$ to $k = 100$. In fact, using the EMD (with Euclidean distance as ground distance), the performance only increased from 93% to 94%, approaching the performance of the χ^2 kernel, but at significantly reduced computational expense [153]. With some images containing less than 200 keypoints, a value of $k = 100$ is excessive. Instead, we followed Zhang's work and experimented with values around 40. Ranging from $30 - 50$, in steps of 2, we found no significant improvement or deterioration in classification performance or running times using these values. For this reason, we chose a value of $k = 40$ to cluster the keypoints in each image.

Finally, the use of image signatures and the EMD was discussed and justified in Section 4.3.2.2. Regarding the choice of ground distance, the Euclidean distance has been widely used as a ground distance with the EMD, and would appear to be the most appropriate choice [225]. In particular, the work introducing the EMD by Rubner [211] and the detailed evaluation of signature and histogram based methods for texture and object recognition tasks by Zhang [153] all indicate that the Euclidean distance is the most appropriate choice and that it's performance is similar, or better, than other similar metrics such as χ^2. Indeed, Ling et al. found that the performance of their version of EMD, EMD-L_1, was equivalent to the EMD with Euclidean distance as ground distance [226]. However, it's worth noting that the computation time invovled with EMD-L_1 is significantly below that of EMD, with Euclidean ground distance, and this appears to be its major advantage. However, given that computation time is not currently a significant issue with these algorithms, it was felt that Euclidean distance was the correct choice for these experiments.

4.5 Results

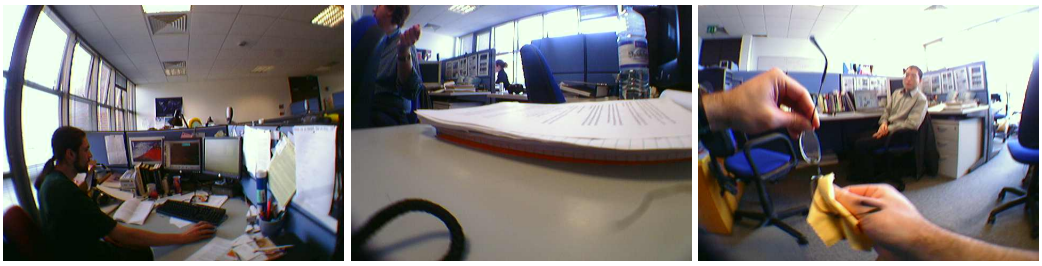
The following sections outline the results obtained using all three descriptors for each of the described approaches. The number of images annotated by each user was split evenly into training and testing sets for each experiment. For completeness, we also present results from experiments carried out using different divisions of the training and testing data. The purpose of this is to explore the impact of a reduction in the training data used (and hence a reduction in processing times and the burden on the user). This is important in the context of a Visual Diary, as we would like to try and obtain high levels of performance as the diary grows. If high performance can be maintained with a low percentage of training data, this would reduce the need to continuously retrain the system every time new images are loaded. We present the results of this experiment for a single user's images. All other results presented use an even split between training and testing data.

The database is challenging, not only because of the large number of settings involved for individual users, but also because of the significant variations in lighting, pose, occlusion, and background clutter contained in the images. In particular, *User 2* has 42 different settings, many of which are visually very similar. For example, images taken in one individual's office can be extremely similar to images taken in another individual's office in the same building. The lighting, decor, etc., are all extremely similar and hence extremely challenging to classify into different settings. Figure 4.11 illustrates some of these issues for a selection of images across different users. Each user has annotated the images to be in a different setting, however, the images contain many similar elements making it extremely difficult to correctly classify them. Figure 4.12 illustrates the variations in lighting, viewpoint, background clutter, and occlusion, possible within a setting.

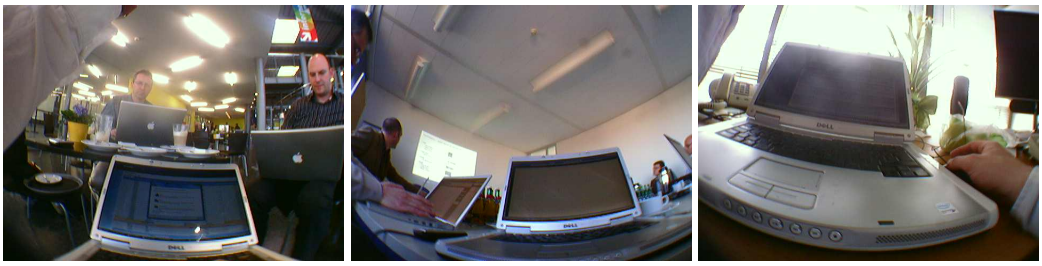
4.5.1 Bag-of-Keypoints Approach

The average classification error for all users can be seen in Table 4.7. The overall classification error for each user using this approach can be seen in Tables 4.8, 4.9, 4.10, 4.11, and 4.12. As previously mentioned, detailed precision and recall figures for each experiment can be found in Appendix A.

Looking at the average figures, the best performing version of the algorithm is the SVM using the U-SURF128 descriptor. However, the average classification error using the SVM is very similar for all three descriptors, varying from 0.0933 for SIFT to 0.0898 for U-SURF128. In



(a) User 1 - Different office scenes



(b) User 2 - Working on a laptop in different locations

(c) User 4 - Meeting colleagues in different offices

(d) User 3 - Chatting to colleagues in different locations

(e) User 5 - Dining out in different restaurants

Figure 4.11: Theses images highlight the problem associated with classifying visually similar images into different settings. Each user has annotated the images above to be in different settings, however, some are extremely difficult to classify. The office scenes in particular are challenging due to the similarities in the colour of decor, structures, and lighting. Other settings prove challenging due to the similarity in a particular object, such as a laptop, dominating the image.

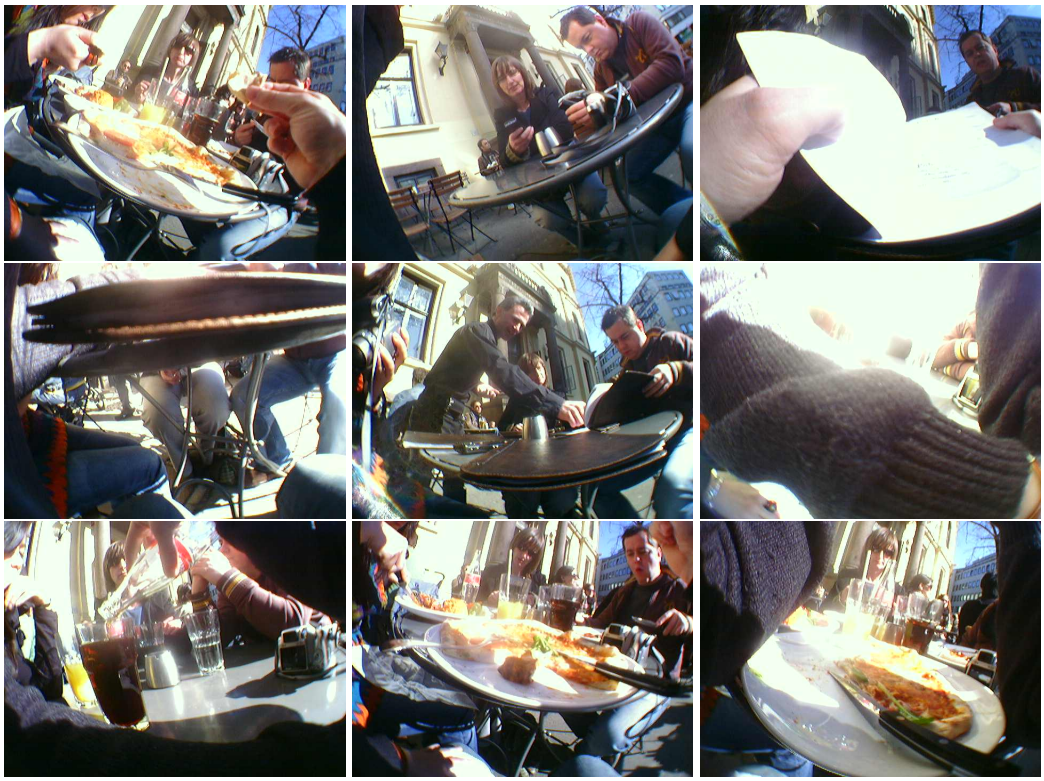


Figure 4.12: These images illustrate problems such as changes in lighting, viewpoint, and occlusion, which can occur between images in the same setting.

general terms, we can say that all three descriptors perform similarly with the SVM, although the computational performance will differ significantly due to the differences in descriptor length. For the MLP classifier, the overall classification error was, again, similar for all three descriptors. The KNN algorithm revealed significant discrepancies across all three descriptors, with the SIFT features performing best. However, the average error for all three descriptors using the KNN algorithm was significantly higher than the worst performing case with the other two classifiers.

An examination of the classification error for individual users reveals that the SVM outperformed the other two classifiers in all cases. This is not surprising given the power of the SVM algorithm. However, the performance of the MLP classifier is quite close to that of the SVM for a number of users (e.g. *User 1* using U-SURF64 and *User 5* using U-SURF128) and, in general, this classifier also performs quite well. The KNN classifier's performance is far below that of the SVM or MLP, and when viewed in light of the performance of these algorithms, it appears relatively poor. The performance of all algorithms suffers as the number of settings increases. In particular, *User 2* has almost double the amount of settings compared to other users. The performance of KNN in this situation is extremely poor, however, the SVM and MLP still perform well. It's worth noting that the quality of the annotations, and subsequent variations in the settings, will

Descriptor	Support Vector Machine	Multiclass Linear Perceptron	K-Nearest Neighbour
SIFT	**0.0933**	0.137	0.2454
U-SURF64	**0.0931**	0.1163	0.3081
U-SURF128	**0.0898**	0.1102	0.3528

Table 4.7: The average classification error for all users for all descriptors.

Descriptor	Support Vector Machine	Multiclass Linear Perceptron	K-Nearest Neighbour
SIFT	**0.0451**	0.0767	0.2239
U-SURF64	**0.0419**	0.0543	0.1706
U-SURF128	**0.0425**	0.0662	0.2081

Table 4.8: The classification error for User 1 for all descriptors. The number of settings in this user's collection is 24.

also have an impact on overall performance. The drop off in performance as the number of settings increases cannot, therefore, simply be attributed to an increase in the number of settings. For example, *User 1* and *User 3* both have a similar number of settings, but the performance is quite different for both user's collections. This is mainly due to differences in the nature of their respective collections (discussed further in Section 6.2). In terms of the descriptors used, the performance is similar across all experiments, with U-SURF64 and U-SURF128 outperforming SIFT for some users and SIFT outperforming U-SURF64 and U-SURF128 for others. The differences, in terms of overall classification error, between descriptors are negligible. Due to the performance benefits associated with its reduced descriptor length, U-SURF64 is the best performing, but purely from a computational viewpoint.

Finally, in order to demonstrate the impact of the levels of training and testing data on the classifiers, we present the classification errors for different divisions of the database in Tables 4.13 and 4.14. The database was split into 10% training data and 90% testings data, and 30% training data and 70% testing data. The SVM performs particularly well, even with only 10% training data. With 30% training data, the performance of the SVM approaches the baseline figures of 50% training data. The same is true for the remaining classifiers, where performance also drops as the amount of training data is reduced. However, the reduction in classification error is not significant for the MLP classifier. Even for the KNN classifier (although the increase is more pronounced), the performance is still quite good at lower levels of training data.

4.5.2 Alternate Approach

The overall classification error for each user, as well as the average classification error, using the alternate approach can be seen in Table 4.15. As previously mentioned, detailed precision and

Descriptor	Support Vector Machine	Multiclass Linear Perceptron	K-Nearest Neighbour
SIFT	**0.2316**	0.2817	0.5397
U-SURF64	**0.2076**	0.2815	0.6629
U-SURF128	**0.2074**	0.242	0.7291

Table 4.9: The classification error for User 2 for all descriptors. The number of settings in this user's collection is 42.

Descriptor	Support Vector Machine	Multiclass Linear Perceptron	K-Nearest Neighbour
SIFT	**0.2272**	0.2454	0.3314
U-SURF64	**0.1494**	0.1587	0.351
U-SURF128	**0.1336**	0.1375	0.4449

Table 4.10: The classification error for User 3 for all descriptors. The number of settings in this user's collection is 20.

Descriptor	Support Vector Machine	Multiclass Linear Perceptron	K-Nearest Neighbour
SIFT	**0.0382**	0.0653	0.0894
U-SURF64	**0.0551**	0.0726	0.1613
U-SURF128	**0.051**	0.0835	0.1946

Table 4.11: The classification error for User 4 for all descriptors. The number of settings in this user's collection is 16.

Descriptor	Support Vector Machine	Multiclass Linear Perceptron	K-Nearest Neighbour
SIFT	**0.008**	0.0159	0.0425
U-SURF64	**0.0086**	0.0146	0.1947
U-SURF128	**0.01**	0.0219	0.1874

Table 4.12: The classification error for User 5 for all descriptors. The number of settings in this user's collection is 10.

Descriptor	Support Vector Machine	Multiclass Linear Perceptron	K-Nearest Neighbour
SIFT	**0.0434**	0.1228	0.1329
U-SURF64	**0.0583**	0.101	0.2356
U-SURF128	**0.057**	0.1206	0.318

Table 4.13: The classification error for User 4 for all descriptors. The number of settings in this user's collection is 24 and the database was split into 10% training data and 90% testing data.

Descriptor	Support Vector Machine	Multiclass Linear Perceptron	K-Nearest Neighbour
SIFT	**0.0451**	0.078	0.0941
U-SURF64	**0.0551**	0.0739	0.1837
U-SURF128	**0.0556**	0.0726	0.2171

Table 4.14: The classification error for User 4 for all descriptors. The number of settings in this user's collection is 24 and the database was split into 30% training data and 70% testing data.

User	SIFT Error	U-SURF64 Error	U-SURF128 Error
User 1	0.5085	**0.4939**	0.5593
User 2	**0.6682**	0.7069	0.6835
User 3	**0.5441**	0.5697	0.5581
User 4	**0.5267**	0.5557	0.5403
User 5	**0.4817**	0.5309	0.5316
Average	**0.5458**	0.5714	0.5745

Table 4.15: The classification error for all users for each descriptor using the alternative approach.

Database Split	SIFT Error	U-SURF64 Error	U-SURF128 Error
10%:90%	0.925	**0.8859**	0.9249
30%:70%	**0.5864**	0.6078	0.6217

Table 4.16: The classification error using the alternate approach with different divisions of the database between training and testing data for User 4.

recall figures for each experiment can be found in Appendix B. In addition, in order to demonstrate the impact of changing the percentage of training and testing data used with this approach, the classification error for different divisions of the database is shown in Table 4.16. In a similar fashion to the first approach used, the database was split into 10% training data and 90% testing data, and 30% training data and 70% testing data.

Overall, there is a clear drop in performance using this approach compared to the Bag-of-keypoints approach. The lowest error achieved was 0.4817 for *User 5* with the SIFT descriptor. This means that just over 50% of the images were classified correctly using SIFT on this user's collection. These are similar performance levels to the worst version of the Bag-of-keypoints approach (U-SURF128 using KNN on *User 2's* image collection). Unsurprisingly, as the volume of training data decreases, the performance level drops further.

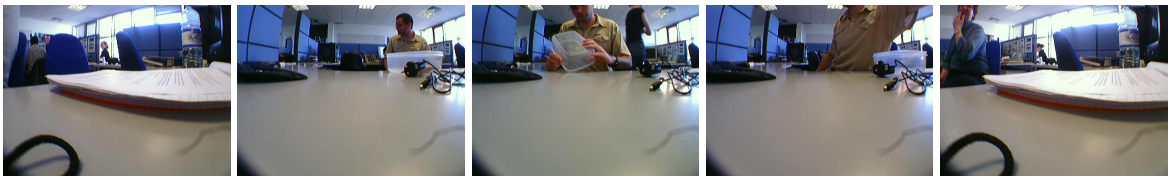
4.6 Discussion

When one considers the challenging nature of the dataset, the results obtained are very encouraging as they provide a significant improvement in performance over the baseline algorithm. The images used contain significant viewpoint, lighting, blur, and affine changes. Notwithstanding this, using the Bag-of-keypoints approach, the system was able to find matches for all settings with high rates of precision and recall under almost all conditions. The only exception to this was using the KNN classifier for Users 1, 3, & 4. For these users, the KNN classifier did not detect all of the available settings, however, since the KNN classifer was the worst performing of the three classifiers used, this is not surprising. The remaining classifiers detected all settings for all users.

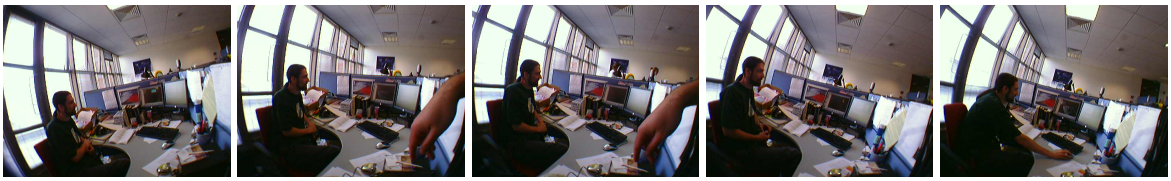
Although the overall classification errors were good for the SVM and MLP classifiers, the levels of precision and recall naturally varied from setting to setting for each user and each classifer. Some settings recorded precision and recall figures of 100%, whilst for others, the levels were also extremely high (see Figure 4.13). An inspection of these settings gives an insight into the situations where the algorithm works well. The images shown in Figure 4.13 all have a relatively uniform background. Foreground objects, which tend to dominate the scene for reasons discussed in Section 3.1, change slightly during these images, with some objects appearing and disappearing. However, the uniformity of the background means that the algorithm matches the images correctly. For other settings, the levels of precision and recall were much lower (e.g. precision of 100% and recall of 29.16% for setting 21 for *User 1* using the MLP classifier; and precision of 39.5% and recall of 33.68% for setting 39 for *User 2* using the SVM). A sample of images where the algorithm struggled can be seen in Figure 4.14. An inspection of these images reveals a much greater variation in the images annotated by the user as being in a single setting. In particular, there are significant changes in viewpoint, differing foreground and background objects, and very challenging lighting changes. It seems clear that the algorithm struggles with such large variations in the images and that a relatively static background is a key requirement for robust setting detection. However, it's worth noting that in certain situations where low rates of precision and recall were achieved with one version of the algorithm (e.g. precision of 39.5% and recall of 33.68% for setting 39 for *User 2* using the SVM classifier with SIFT features), the rates improved significantly with another version (e.g. precision of 86.48% and recall of 67.37% for setting 39 for *User 2* using the SVM classifier with U-SURF64 features). Therefore, a combination of features, although computationally extremely expensive, may generate further improvements in results, and this area will be further investigated in future work.

As previously stated, the performance of the MLP was reasonably close to the SVM, however, overall the SVM appears to be the most suitable classifier to use. Another very encouraging aspect of this approach was the performance of the algorithm as the level of training data decreased. Even at 30% training data, the level of performance was not significantly below that of the original choice of 50%. At 10% training data, the performance suffered, but was still acceptable, particularly for the SVM and MLP classifiers.

The performance of the second approach used was disappointing when compared to the first approach, however, it also provided a significant improvement over the baseline algorithm. The average classification error for the baseline algorithm ranged from 0.7197 for SIFT features to



(a) User 1 - Setting 2



(b) User 1 - Setting 24

(c) User 2 - Setting 28

(d) User 4 - Setting 10

(e) User 3 - Setting 20

Figure 4.13: Sample images where setting detection achieved high rates of precision and recall

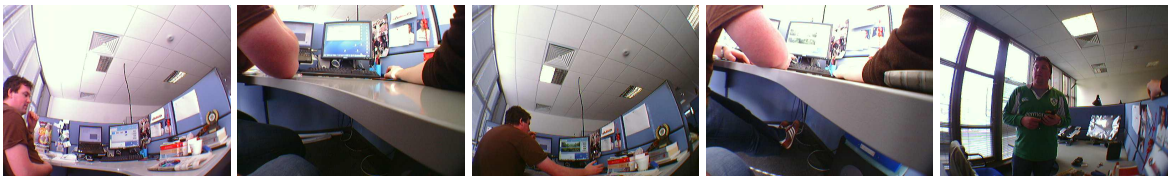
0.8383 for U-SURF128 features. The second approach developed for this thesis improved upon these figures, with average classification errors ranging from 0.5458 for SIFT to 0.5745 for U-SURF128. Indeed, the performance of the second approach appears more stable than our baseline algorithm across all descriptors due to the relatively small differences in classification error. When one examines the classification error for individual users using this approach, we can see that we have made significant performance gains for all users. However, the precision and recall figures for numerous settings were extremely poor using this algorithm, with a number of settings achieving figures of 0% for both precision and recall. Although this algorithm is similar to the baseline approach, these results justify the choice of K-means over X-means in the approaches developed in this thesis. Despite this, the algorithm's performance is still far below that of the Bag-of-

keypoints approach, and it's performance deteriorated rapidly as the volume of training data used decreased. It is clear that this approach is incapable of dealing with an increasingly large number of settings as would be required by a Visual Diary application.

(a) User 1 - Setting 15

(b) User 2 - Setting 25

(c) User 3 - Setting 2



(d) User 4 - Setting 10

(e) User 2 - Setting 39

Figure 4.14: Sample images where setting detection achieved low rates of precision and recall

4.7 Conclusion

In this chapter, we discussed interest point detectors before describing the decriptors used in our work. The SIFT and SURF algorithms were described and a detailed description of both is presented in Appendix E. We also presented two approaches to performing setting detection in lifelog images, as well as a simple baseline approach with which to judge the two approaches developed. The initial step in all approaches is for a user to annotate their images into settings. We developed

a simple annotation tool to allow the user to quickly and efficiently perform these annotations.

In the baseline approach, keypoints are extracted from user annotated settings and these are subsequently clustered using the X-means algorithm. The EMD was used as a distance measure between test images and each setting. The baseline approach allowed us to evaluate the results obtained using the other approaches developed in this thesis, as well as influencing the overall design of these algorithms.

Using the bag-of-keypoints approach, we created an image descriptor for each of the training images of each setting and then learned a classification model using three different multiclass classification algorithms. The classification algorithms used were KNN, MLP, and SVM. Using the learned model, we subsequently classified the test images.

In the second approach, we modelled user annotated settings and extracted keypoints in an attempt to create a descriptive model of each setting. We then attempted to match images against this model. Clustering was performed using the K-means algorithm and the EMD was used as a distance measure between images.

Finally, we described the experiments performed and presented the results of our analysis. The bag-of-keypoints approach using an SVM classifier was found to perform best, whilst the second, alternative, approach used was the poorest performing method. In terms of the descriptors used, there was little difference in performance between SIFT, U-SURF64, and U-SURF128. In general, U-SURF128's performance was above that of the other two descriptors, but the difference was not significant.

The overall goal is to use a technique to construct a Visual Diary in order to faciliate the management and organisation of passively captured lifelog images. By analysing the different settings a user encounters during their daily life, we may be able to make more meaningful assumptions about each individual day and, hence, assist the user in managing their collection. When settings are analysed across an extended time period, groups of similar activities within the collection can be grouped together. What may be considered unique in the context of a single day's activities might turn out to be more mundane when analysed over an extended period of time. An analysis of this kind will lead to a more meaningful visual representation of the user's life in the form of a Visual Diary. By detecting settings highlighted by the user as being important to them for some reason, we should be easily able to link events together across numerous days, weeks, and months. At that point we would be in a position to easily identify recurring and unique events from a large collection of SenseCam photographs, as well as facilitating easier searching and browsing of the

collection. In the following chapter, we discuss an implementation of a Visual Diary, and we present our initial findings relating to some user's experiences with this application.

CHAPTER 5

My Places: An Implementation of a Visual Diary

5.1 Introduction

In the previous chapter, we briefly discussed interest point detectors before describing the three decriptors used in our work. The SIFT and SURF algorithms were described in detail and we also identified the versions of SURF we use in our work, namely U-SURF64 and U-SURF128. We also developed a number of approaches which enable us to detect settings in a Visual Diary. In particular, the first approach developed, based on a bag-of-keypoints algorithm, achieved excellent results with a large volume of data and a large number of classes. The performance of the second approach, based on clustering the keypoints in each setting independently, degraded quickly as the number of settings to be classified increased.

In this chapter, we present an implementation of a Visual Diary application. This application leverages the results obtained using the bag-of-keypoints algorithm with SIFT image features and an SVM classifier, as this was the most successful approach developed. In Section 5.2, we discuss the motivation behind the design of the user interface in the Visual Diary. A key challenge is to facilitate the user to quickly browse and retrieve the images of interest from their Visual Diary and these considerations have been taken into account in the design. In Section 5.3, we describe a number of approaches to user evaluation that can be used in order to validate the approach taken in this thesis. In Section 5.4, we describe some user experiments carried out in order to investigate the utility, or otherwise, of the Visual Diary application. The results of this analysis are presented in Section 5.5. This evaluation is based on five real users, and will be qualitative and quantitative

in nature.

5.2 Interface Design

My Places is a web-based image browser designed to facilitate the simple and effective management of a large collection of lifelog images. On its web interface (see Figure 5.1), keyframes from a users uploaded photos are displayed in column format, with the groupings of photos automatically formed based on the date the image was captured. In this case, the images have all been captured using the Microsoft SenseCam.

On opening the application, a single screen is presented with a week's worth of images displayed. This is the main photo-collection page and is the only page in the system. There are two key areas on this page. The calendar, in the top right, allows the user to select which day's images they wish to view. Images from the selected date are displayed on the left hand side of the screen, below the calendar, with subsequent day's images being presented in the additional columns, from left to right. Days for which there are no image links currently available are greyed out.

The main focus of the system is on the displayed images. Images displayed here are keyframes selected from events which have been detected using an offline process. The events have been detected using a combination of MPEG-7 features [166] and SenseCam metadata using a technique developed by others in our research group. Briefly, the aceToolbox was used to extract low-level MPEG-7 features from the SenseCam images. A more detailed description of the aceToolbox can be found in [166]. A brief description of the descriptors used is provided here and more detailed information can be found in [79].

- Scalable Colour generates a colour histogram in the Hue Saturation Value (HSV) colour space that is encoded using a Haar transform, thereby providing a scalable representation.

- Colour Layout is designed to capture the spatial distribution of colour in an image or region by clustering the image into 64 blocks and deriving the average colour of each block. These values are then transformed into a series of coefficients by performing an 8×8 Discrete Cosine Transform (DCT).

- The Edge Histogram captures the spatial distribution of edges, which are identified using the Canny algorithm, by dividing the image into 16 non-overlapping blocks and then calculating 5 edge directions in each block.

Figure 5.1: My Places Image Browser. In the image shown, images from the week beginning April 19th are shown.

- Homogeneous Texture describes directionality, coarseness, and regularity of patterns in images by partitioning the image's frequency domain into 30 channels and computing the energy and energy deviation of each channel and then outputting the mean and standard deviation of the frequency coefficients.

The algorithm used to detect events is based on the agglomerative hierarchical clustering approach [218]. This approach starts by considering each individual image as a cluster, and the sequence is then formed by successively merging clusters. The merging is performed based on the nearest distance between images, where the distance calculated is the Euclidian distance based on a feature vector containing the normalised low-level features and SenseCam metadata for each image. The data was normalised between values of 1 and 0. Time constraints are also imposed on the clustering process based on an algorithm proposed in [227]. This is implemented by considering the time each photo was taken and penalising photos taken further away from each other (in time) using a cost function, thus increasing the distance measure. The cost function is calculated

based on the average squared distance of the data set.

Once the clustering process is complete, keyframes are selected from each event to display in the image browser. The choice of keyframe is important as we would like it to be the most representative image of the event in question. Many keyframe selection strategies exist, based on global features or heuristic strategies (such as selecting the middle keyframe). However, we believe a representative keyframe can be extracted using local features. Each event consists of many images. Some of these images will be similar to each other and some very different. This broadly depends on the activity being depicted in the sequence of images in question. A sequence of images representing a setting will have numerous images which are extremely similar, with perhaps only a few depicting other objects or scenes. An event where the user is constantly on the move will have a much higher degree of variation amongst the images. Using local features, we can match each image in an event to all of the other images in the same event. For each image, a certain number of keypoints will match between it and the test image. The closer the candidate image is to the test image, the higher the number of matching keypoints. By calculating the number keypoint matches between an image and all others in the event, and then working out the average value, we can calculate a similarity value for each image in the event. To select the most representative image from a sequence of images in an event, we simply select the image with the highest similarity value. Intuitively, the image that has the highest average number of matches to all others in the event will be the closest to all other images in that event and, therefore, the most representative.

The first step in this process is to extract SIFT features from the images in each event. Given a test image, each one of its keypoints is compared with the keypoints of every image present in the event. The Euclidean distance between each invariant feature descriptor of the test image and each invariant feature descriptor of the remaining images in the event is computed at first. However, two keypoints with the minimum Euclidean distance (the closest neighbours) cannot necessarily be matched because many features from an image may not have any correct match in the training database (either because of background clutter, noise, or perhaps because the feature was not detected at all in the training image). Therefore, a more effective method than simple matching using Euclidean distance is to use the distance ratio test. To examine whether a point from the 1^{st} image has a match in the 2^{nd}, its two most similar descriptors in the 2^{nd} image are found. If the ratio of the nearest distance to the second nearest distance is less than 0.7, a match is declared. The number of matches between an image and all other images in the event are summed,

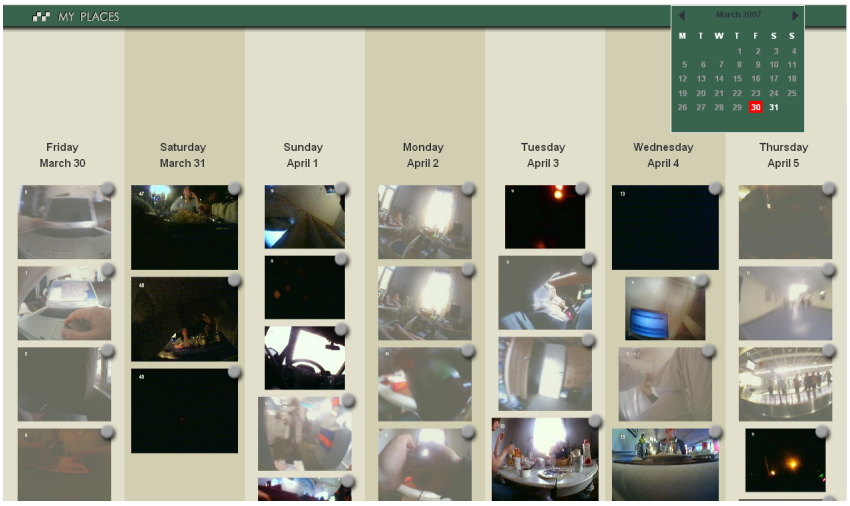
and then the average number of matches is calculated. The image which has the highest average is deemed to be the most similar to all other images in the event and, hence, is selected as the keyframe for that event.

This measure performs well because correct matches need to have the closest neighbour significantly closer than the closest incorrect match to achieve reliable matching. For false matches, there will likely be a number of other false matches within similar distances due to the high dimensionality of the feature space. We can think of the second-closest match as providing an estimate of the density of false matches within this portion of the feature space and at the same time identifying specific instances of feature ambiguity. By rejecting all matches in which the distance ratio is greater than 0.7, we can eliminate 90% of the false matches while discarding less than 5% of the correct matches [83].

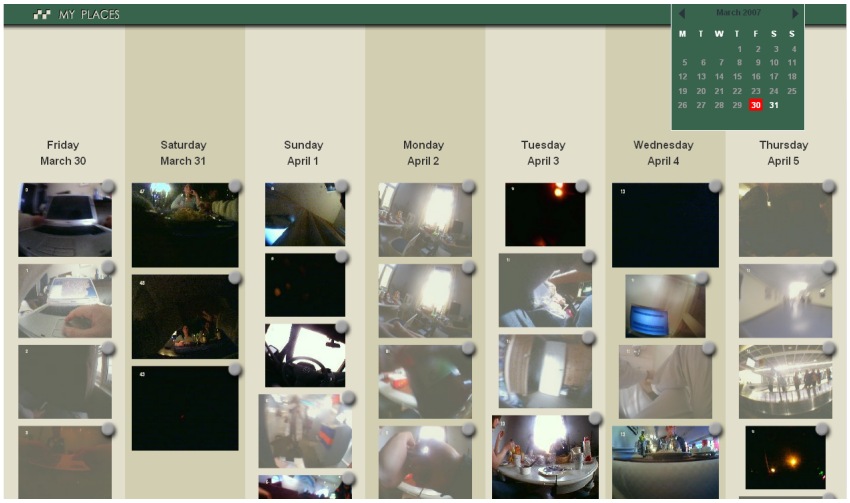
As discussed in Section 2.7, this event detection process is simply utilised to enable us to recognise settings which are of importance to the user. As mentioned above, the event detection process was developed in conjunction with other individuals in our research group. We simply leverage this event-based structuring of the data in our user interface to provide an initial structuring of the image collection. This enables us to demonstrate the benefits of the detected settings. As such, a more detailed description of the event detection process is beyond the scope of this thesis. However, the keyframe detection process described above is part of our program of work, but is utilised solely in the context of selecting appropriate keyframes to display in the user interface of this thesis.

By scrolling down, users can view all the keyframes for each event in a particular day. Users will note that some images are initially shaded (see Figure 5.2(a)). These are images which have no additional links associated with them (i.e. they are keyframes from events which were not considered to be a setting by the user). By mousing over, or clicking an image, the image will appear clearly for easier viewing (see Figure 5.2(b)). Images which are not shaded when the system is first loaded (but remain clear during normal use) represent those for which additional links are available (i.e. they were considered to be part of a particular setting). Mousing over one of these images will display a red border around the image and all of the other images which were deemed to be in the same setting (see Figure 5.3). By moving the mouse around the screen, the user can visualise all of the links occurring in their collection very quickly and easily. By clicking one of these images, the red border is again displayed around the image, and other linked images are also highlighted by a red border. However, clicking the image holds the links, so that

the user may peruse the collection for other images which occur in the same setting as the clicked image. Clicking another image will remove the linking information. The red border represents those images which are a strong match to the selected image (i.e. they occur in the same setting). Other images may be highlighted by a yellow border around the image. These are weak links, and represent the setting which was found to be second closest to the selected image.



(a)



(b)

Figure 5.2: (a) The clear images represent those with additional links. Those without links remain shaded. (b) A shaded image, such as the image in the top left, can be made clearer to facilitate easier viewing by mousing over the image.

The links are based on an analysis of particular locations, or settings, detected in the SenseCam images, using the best approach described in Section 4.5. Therefore, images taken at similar locations should be linked together. Providing these links allows a user to very quickly determine when and where they spent time in certain locations throughout a large collection of images. The exact nature of these linked images is based on the definition of a setting, outlined in Sec-

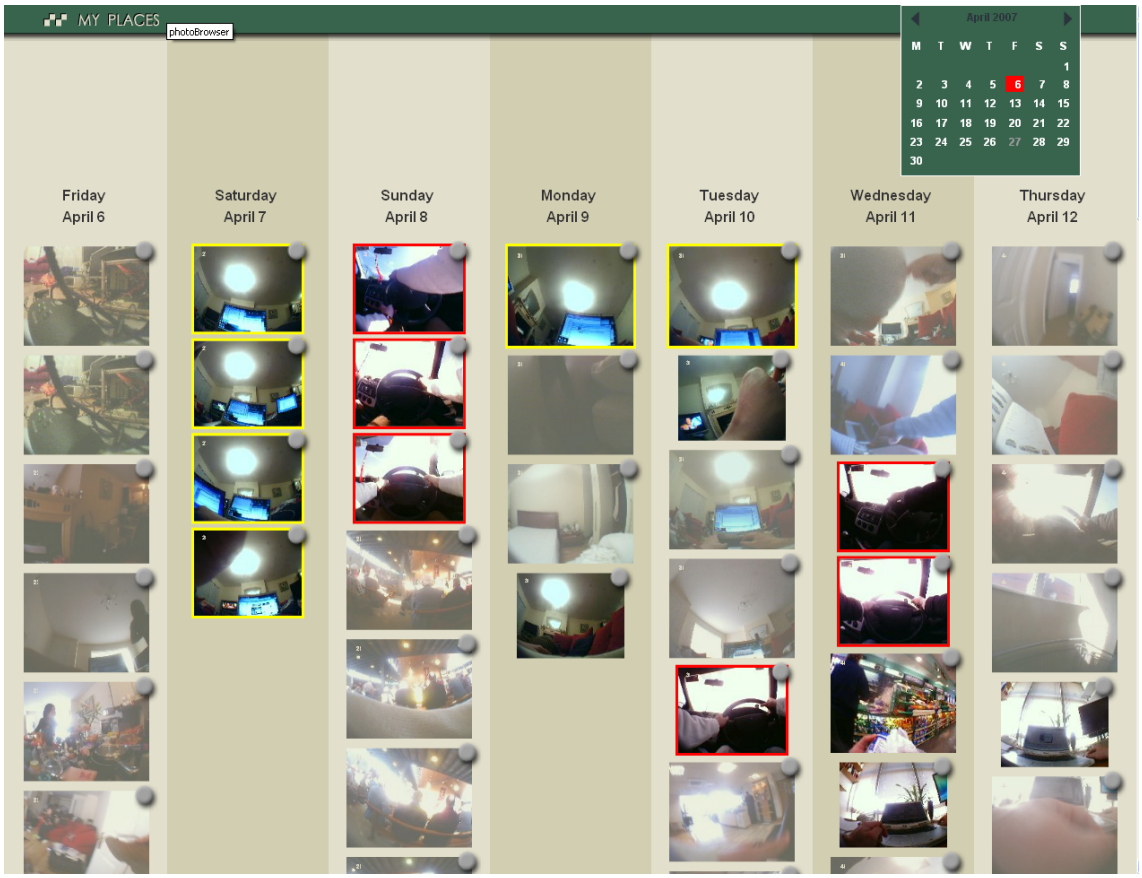


Figure 5.3: In the image shown, images from a number of settings are linked together. The images with the red outline all link images from settings where the user was driving. Weak links are also displayed, using a yellow outline around the images. In this case, the weak links show a setting where the user was working on his laptop at home.

tion 3.1. In addition, certain images are displayed in different sizes. The size is related to the perceived importance of the event in question, and three different sizes are used. The smallest images relate to events deemed to be of least importance and larger images to those deemed to be of most importance. The importance information was acquired during the annotation process as described in Section 4.3.3.1. The system is freely accessible online for demonstration purposes (http://www.cdvp.dcu.ie/SenseCam/michael.html).

5.3 User Evaluation

There are two main reasons for carrying out evaluations: to ensure that the system developed meets user needs, and to assess whether the original objectives were achieved [228]. Evaluation provides a mechanism to measure objectively what has been achieved through performing setting detection. It provides an opportunity for us to discover how users would like to use these large collections of

images. We make many assumptions about our users and their needs, often basing our expectations of digital usage patterns on existing usage of analog or other digital resources [229]. However, at best, these can only be educated guesses, since access and usage can change completely in a Visual Diary. Tasks which users may like to perform with standard photo collections may be irrelevant in a visual lifelog.

At a more practical level, giving the target user group the chance to evaluate and test the system, and in particular the interface, enables us to take advantage of feedback and suggestions for future improvements [230]. In particular, the questions we would like to address include:

- How do users use the interface?
- Do they make use of all facilities available to them in the interface?
- What is the effect of the interface on the users?
- Are they satisfied with the service?
- Does the system trigger interest in new topics, or challenge existing perceptions?
- What are the formal and informal learning outcomes?
- Are settings a useful method of structuring a Visual Diary?

In this thesis, we are interested in exploring whether or not the interface facilitates the user in locating images of interest in their collections and whether it allows them to make some sense of their collections. This analysis will indicate whether the detection of settings, as defined in Section 3.1, is useful in the management of a Visual Diary.

In large scale assessments, it would be best practice to run the evaluation with users from a variety of age groups, educational levels, backgrounds, and interests. The needs and expectations of different individuals will differ considerably and it would be normal to consider such issues in a large scale evaluation [231]. However, in these experiments, we are limited by the number of users available. This limit is imposed by the number of users who have gathered image data using the SenseCam [1]. One benefit of using a limited number of expert users at this stage is that they can critically evaluate the system based on their knowledge of the overall domain. In addition, they can also provide some valuable insights in relation to future directions for the system.

[1] In turn dictated by the number of SenseCams available

5.3.1 Evaluation Methods and Tools

There is no single golden method for evaluating digital programs and measuring their effectiveness and impact. Experience has shown that it is better to combine several methods in order to verify and combine data, relating quantitative with qualitative results. The dichotomy between the quantitative and qualitative approaches is usually artificial, since they work best in complementary ways, illuminating different aspects of a complex phenomenon [232].

A number of techniques are commonly used in traditional evaluation work and some are appropriate for our work. For example, Computer logging of user interaction facilitates the automated logging of a users interaction with a system and provides a reliable way of recording user's choices and the path they selected through the website or program. This is generally an easy and objective way of obtaining a large set of data. However, the results are not very meaningful on their own, but can be useful when combined with interviews, focus group discussions and observation [231].

Electronic questionnaires provide an easy way to obtain feedback from end users, although the results generally pose problems for valid statistical analysis, as the sample is self-selected [230]. Interviewing and focus group discussions with a small number of targeted users provides an effective method of evaluation and offers the opportunity for both structured and open-ended data collection. If an application is intended for specific groups, discussions with focus groups can be very useful during the planning and development stages.

When testing a prototype, interviewing and observation can take two forms, often described as 'cued' and 'uncued' testing [229]. Cued testing involves explaining to the users what the program is about and asking them to perform specific tasks or to answer questions. Another possibility is engaging users in conversation and encouraging them to 'think aloud' as they go through the program, while recording their responses. With uncued testing, users are observed unobtrusively as they use the program and are then asked questions about their experience.

5.4 Evaluation Experiments

In this thesis, our evaluation is constrained by the number of users available to perform the necessary testing. This constraint makes the significance of the quantitative results calculated questionable. Similarly, one could question the validity of the qualitative analysis provided. However, as a first step in the development of an effective Visual Diary application, we believe the results are useful and that they do provide some interesting insights into how users might realistically use

such an application. As described in Section 5.3, the goal of user evaluation is to determine if the system developed meets users needs and if our own original objectives have been achieved. The experiments described in this section are designed to help us determine if we have successfully achieved these goals.

In terms of meeting users' needs, we have proposed in this thesis that the detection of different settings in a visual lifelog of SenseCam images is a useful way to structure and subsequently browse through a Visual Diary. Therefore, one of the main goals of the user evaluation experiments is to determine the usefulness of the availability of the setting information. By building an application which allows the user to visualise the setting information via the highlighting and linking of various images displayed in the image browser, we can determine how useful this information is in browsing through a portion of each users Visual Diary. This process will also answer the second goal of user evaluation, in that it will determine whether we have achieved our objectives in this thesis regarding setting detection. From a quantitative point of view, we have already demonstrated this to be the case with the results presented in Section 4.5. These results demonstrate that we can successfully detect various user annotated settings in lifelog images.

In order to perform the evaluation, we have used two of the approaches described in Section 5.3.1, namely 'Computer Logging of User Interaction' and 'Electronic Questionnaires'. The tasks performed by the users are dictated by a questionnaire, presented in Appendix C. All of the images clicked on or selected by the users during the experiments were logged to an SQL database. During the tasks, users were also asked to select any images relevant to the task they were currently performing. Images were selected by simply clicking on a grey dot in the top right corner of each image. Once selected, the dot turned orange. This recorded the image name, as well as other information such as a timestamp and the image owner, to the database in order to analyse the images selected.

Regarding the questionnaires, a form of cued testing was used. The initial step in cued testing involves explaining to the users what the system is about. A document was created with some background information about the system, followed by a number of tasks for the user to complete (see Appendix C). Each task involves browsing through the collection in order to identify images which relate to the particular question currently being asked of them. In order to complete each task, the user needs to navigate through their collection using the application in order to locate the relevant images. It is hoped that users will find the setting information useful in completing the tasks. An ancilliary benefit of performing the evaluation in this fashion is that we can also

analyse users' thoughts on the user interface. Although this is not a core concern at this stage, users' thoughts on the user interface provide us with some useful directions for future work in this area.

The questionnaire then followed and this included questions which could be analysed quantitatively and qualitatively. The questionnaire is designed to determine the overall usefulness of the system, as well as examining specific aspects of interest in the system. Specifically, users are asked how useful they find the linking together of the images in different settings. In addition, they are also asked how useful they find the different sizes of images, relating to their importance. Other questions were more general in nature, relating to their overall satisfaction with the system and its user interface. The user was requested to provide as much feedback as possible on problems or improvements they would like to see in a future version of the application. All of the information presented to the user before the experiment began, as well as the tasks and questionnaire, is presented in Appendix C. Finally, usefulness ratings were sought in relation to specific aspects of the system (as well as the reasons behind these ratings) with some basic quantitative analysis capable of being performed using these.

5.5 User Feedback

In order to analyse the results of these experiments, a number of different metrics were used, based on the specific information gathered during the evaluation experiments. Firstly, the usefulness ratings were calculated for each question posed during the evaluation, and the results of these can be further analysed. Secondly, the average number of clicks (i.e. the amount of times a user selected a setting during the evaluation) can be calculated from the information logged to the database. This gives an indication of how much the user used this feature when performing the experiments. Finally, the overall user satisfaction (expressed as a percentage of the number of users) with different key aspects of the system, can be calculated, thereby giving a further indication of the success, or otherwise, of the evaluation process.

The usefulness scores for all questions posed can be found in Table 5.1. In terms of usefulness scores, where 1.0 indicates 'not useful' and 7.0 indicates 'very useful', the average user rating for the overall system is 5. Most users' ratings were high, however, one user (User 4) only gave the system a rating of 2. It was interesting to note that this particular user had the least amount of images gathered and gained least from the current implementation due to the small collection

involved.

Bearing the goals of the evaluation process in mind, the usefulness scores indicate that the application has met the users' needs in terms of structuring a Visual Diary using setting information. Of particular importance for this thesis was our user's thoughts on the linking together of images from the same setting. A number of questions were asked, specifically related to this feature. All users stated that they found the image linking useful in completing the tasks and finding the images in question, and the average score for this feature was 5.4. Another question related to the image linking asked users how well they thought the information was presented. The average score here was 5.0. Similarly, the users found the system easy to use (5.6) and easy to learn (5.4). In addition, users also enjoyed using the interface (5) and found it effective in helping them complete their tasks (4.8). In particular, users commented on the visualisation of a week's images on a single screen as being an attractive element of the user interface. Overall then, users found the representation of the settings, via the image linking, to be a useful feature in completing the tasks, and an important feature to have in a lifelog image browser.

The one area which users didn't find useful was the information relating to the importance of different settings. This was implemented by representing different keyframes using three different sizes. Users were asked how useful they found this feature and the average score was 3.4. Only one user found this feature useful and the comment of one in particular is informative: "once I started performing the tasks, I hardly noticed the different sizes". Another user commented that it would be useful to "make important settings even bigger". Overall, users didn't find the feature useful in completing the requested tasks, although some users felt that it might be useful if they could adjust the importance themselves, rather than have the ratings fixed at the annotation stage.

Question	**Average usefulness scores**
How useful is the system overall?	5
How useful is the image linking?	5.4
How well is the information presented?	5
How useful are the different image sizes?	3.4
How easy is it to use the system?	5.6
Was it easy to learn how to use the system?	5.4
Is it easy to find the information you needed?	4.6
Was the information provided about the system easy to understand?	5.2
Was the interface effective in helping you complete the tasks?	4.8
Is the organisation of the information on screen clear?	4.6
Is the interface of the system pleasant?	5
Did you enjoy using this interface?	5

Table 5.1: Usefulness scores for all questions in user evaluation

Besides the usefulness ratings discussed above, the information logged to the database was also analysed. By examining the amount of images clicked during each task (i.e. the amount of times the user selected a particular setting), we can get an idea of how often this feature was used during the experiments. The results of this analysis can be seen in Table 5.2.

User	Number of clicks	Percentage of overall activity
User 1	22	25.58%
User 2	45	15.2%
User 3	13	13.99%
User 4	3	5.55%
User 5	13	20.63%

Table 5.2: Number of clicks users made during evaluation experiments

These results show the total number of times a user selected a setting during the experiments, as well as showing the number of settings clicked as a percentage of the overall activity logged to the database during the experiment. This is a crude measure of the amount of times a user used the setting information during the experiments. For example, User 1 used it a little over 25% of the time, while User 4 only used it 5.55% of the time. Again, it's interesting to note that User 4 had the least amount of images used in these experiments. On their own, these results are not particularly informative, but in conjunction with the analysis provided from the usefulness scores, they back up the hypothesis that the setting information is indeed a useful feature for browsing through a Visual Diary.

Finally, in terms of quantitative analysis, the users were asked directly whether the image linking and image sizing was useful in finding the information requested in each task. All of our users found the image linking useful in these experiments, however, only 20% of our users found the image sizing useful. These results, when viewed in conjunction with the other results previously described, confirm our conclusions.

5.6 Discussion

From a qualitative point of view, the users made a number of comments about their experiences with the system. These comments reinforce the findings found in the quantitative analysis described above. All users found the linking of images from the same setting together to be a useful feature. Also, the ability to visualise multiple days' images at a glance (and view the connections between them) was consistently cited as an attractive feature of the system. This is extremely

encouraging as it justifies the work in this thesis relating to setting detection. Some users didn't like the amount of scrolling involved and the fact that they couldn't see all of the images linked together. One user commented: "Lots of scrolling! And this makes it difficult to use the linking effectively". However, these comments are encouraging as they show that users felt the linking of the settings would be more useful if certain user interface issues were resolved. The mouse over effect, where the links between settings are highlighted as the user moves the mouse around the screen, was also highlighted as a nice effect as it allows for an extremely fast visualisation of the links between settings in a week's images.

Nothwithstanding the positive results presented in this evaluation, a number of issues arose which are worth highlighting. In the main, these relate to the user interface. Some users felt that certain images should have been linked together, but weren't. This is not a deficiency in the algorithms used to perform setting detection. Rather, it is the case that these images simply weren't annotated by the user during the annotation phase. Others found the calendar difficult to use and were frustrated by the fact that a large volume of related content was off the screen. One user commented that the setting information "could have been much more useful", but found it difficult to make full use of the feature because of the problems described above, and also due to the fact that it was difficult to determine what time in the day an item occurred. Users liked the "strip style" interface, but complained about the amount of scrolling involved. It is difficult to see how this issue could be resolved, although one user suggested using a much larger screen to reduce the amount of scrolling.

Although users didn't find the sizing of images (relating to their importance) to be particularly useful in this version of the Visual Diary, some of their comments indicate that a similar feature would be useful. One user commented that they would like to "list the most interesting events", whilst another would like to "make important settings even bigger". There is a contradiction here in that users indicated that they didn't find the feature useful, but would like important settings to be highlighted in some fashion. An analysis of the importance scores provided in the questionnaires seems to indicate that the important settings are the ones that don't occur on a regular or routine basis (see Table 5.3). For example, events such as working, eating, or on the bus, generally received a low score, but those which occurred infrequently, such as a going to a restaurant for dinner, received a high score. A more appropriate solution to this issue may be to simply allow the user to indicate their favourites in the Visual Diary itself and to highlight these favourites using a larger size.

Finally, the shading of images was also consistently mentioned as a feature users didn't like as it made images "difficult to see". In terms of new features, users would like to see the user interface issues previously discussed resolved. They would also like a number of other interesting features, such as the manual adjustment of importance, ability to add new links, playback of all images in an event, a list of the most interesting/important events in a separate panel, and the ability to save favourite events. The ordering of images by time (in order to arrange each day) was also highlighted by a number of users as a feature which would make it much easier to use the interface.

Event	**Importance score**
Going to restaurant for dinner	5
Watching football in the pub	5
Chatting to a colleague at a conference	5
Chatting to a colleague in work	4
Chatting with students in office	2
Chatting over lunch	1
Meeting a colleague over coffee	3
Driving home	1
Relaxing at home	1
Shopping	2
Meeting friends for dinner	5
Going to the shop	3
Returning home after weekend away	5

Table 5.3: Importance scores for different events in user evaluation

The overall purpose of this evaluation is to determine whether the setting information is valuable in the context of a Visual Diary application. Settings were defined in Section 3.1 as images captured at the same location in the real world that have been flagged by a user as being important. Given that definition, we structured the Visual Diary based on the detection of these settings and found that users liked having this information available to them. Therefore, the evaluation demonstrates that the detection of settings, as currently defined, is a useful feature for users to have available in the management and organisation of a large volume of lifelog images. In addition, although certain issues regarding the identification of important images were raised by users in the user interface evaluation, they did indicate that this information was important, further validating the detection of settings as defined in Section 3.1. A more efficient strategy to alleviate this issue is discussed in Section 6.3.2. Although we acknowledge that the evaluation performed here is not exhaustive, mainly due to the limited number of users gathering data with the SenseCam, it is nonetheless proposed that the conclusions remain valuable and provide some interesting in-

sights into the future directions of this work. Users consistently agreed that the setting information was useful, and although certain user interface problems were highlighted, these were generally mentioned because they limited the user in fully utilising this feature.

5.7 Conclusion

In this chapter, we presented an implementation of a Visual Diary application. We discussed the design and implementation of the user interface and also outlined the techniques available to perform an evaluation of such an application. This information was provided in order to justify the evaluation approaches used in this thesis. The approaches used, Computer logging of user interaction and Electronic Questionnaires, were used in conjunction with a qualitative analysis of information provided by users, in order to determine whether the system meets user's needs and whether our own goals in this thesis have been achieved.

The experiments performed using the methods mentioned above were then described and the feedback gathered from users was presented. Overall, users found the interface pleasant to use, they enjoyed the ability to visualise a large quantity of lifelog images on a single screen in their visual diaries, and they found the setting information useful in completing the requested tasks. Therefore, we concluded that the experiments showed that the use of setting information is an effective method to structure a Visual Diary.

The evaluation also raised a number of issues which are key to ensuring that the Visual Diary remains an effective tool over a longer time period. Certain user interface issues detracted from the user's overall experience and from their ability to take advantage of the setting information. In the next chapter, we outline steps designed to overcome some of these issues. In particular, we focus on the ability of the Visual Diary to change over time. By analysing the different settings a user encounters during their daily life, we may be able to make more meaningful assumptions about each individual day and, hence, assist the user in managing their collection. When settings are analysed across an extended time period, groups of similar activities within the collection can be grouped together. What may be considered unique in the context of a single day's activities, might turn out to be more mundane when analysed over an extended period of time. An analysis of this kind will lead to a more meaningful visual representation of the user's life in the form of a Visual Diary. By detecting settings highlighted by the user as being important to them for some reason, we should be easily able to link events together across numerous days, weeks, and

months. At this point we will be in a position to easily identify recurring and unique events from a large collection of SenseCam photographs, as well as facilitating easier searching and browsing of the collection. In the following chapter, we discuss all the issues necessary to ensure that the Visual Diary dynamically changes and remains relevant over time. This will include the addition and detection of new settings in the Visual Diary, and an analysis of the annotated settings to determine what characteristics annotated settings have. This work should provide the ability of the Visual Diary to dynamically change as more and more images are added to the collection, ensuring the application remains relevant to user's needs.

CHAPTER 6

Analysing a Changing Visual Diary

6.1 Introduction

In the previous chapter, we introduced *My Places*, a web-based Visual Diary designed to facilitate a user in quickly browsing through a lifelog of passively captured images. We carried out a user evaluation of this lifelog to determine if the detection of settings, as proposed in this thesis, was in fact a useful way to assist users in making sense of the large volume of information in the Visual Diary. Our experiments indicated that setting detection was indeed a useful feature, and in fact some users were frustrated by the fact that certain flaws in the user interface design limited their use of the feature. Regarding the user interface, the experiments revealed that users were very pleased with the novel interface provided as it allowed them to quickly visualise a week's worth of images on a single screen.

Despite the success of the evaluation, a number of issues were raised which require further analysis. The experiments conducted so far were static in nature. By static, we mean that we asked users to annotate their data, we showed that we can detect annotated settings across a number of users image collections, and we built, and evaluated, a Visual Diary application to allow users to use the setting detection feature to determine if the objectives of the thesis have been met. The question remains as to how does the technology developed translate to a more realistic scenario? How do we detect new settings, or do we simply keep detecting the original settings annotated by the user? Do certain settings disappear over time, or do others appear and disappear at regular intervals? How long do settings last? What times do they occur during the day and are any of them common across multiple users?

The answers to these questions will allow us to create a Visual Diary which can grow and

change over time, as the user adds more and more images to their collections. In Section 6.2, we analyse the different settings annotated by the users, in the hope of gathering further insights into what constitutes a setting. In particular, we will focus on the times particular settings occur during the day, analysing whether patterns occur across days, and examining what this can tell us about the user's activity during the time period in question. It is hoped that this analysis will allow us to provide a more meaningful representation of the user's life in the Visual Diary.

In Section 6.3, we focus on the issues involved in managing the growth of the Visual Diary over time. It is worth highlighting that these are software engineering solutions to tackling this particular problem, and other solutions may be feasible. In particular, we will focus on the detection of new settings in the diary as they appear. We will outline how new settings will be detected and validated by the user, as well as addressing the issues raised by users in the evaluation concerning the importance of particular settings. We also introduce a much larger data set, gathered over an extended period of time. We describe how we detect new settings over a much larger collection than previously described in this thesis. This is important as it demonstrates that the setting detection approach is scalable, and can dynamically change, as the volume of images continues to grow. It also provides some pointers for future work in this area.

Finally, in Section 6.4, we outline how we can integrate the various features described in this chapter into the existing browsing tool, described in the previous chapter. We present the interfaces involved and demonstrate how they can be used to achieve the required functionality. We also describe how these additions to the main browsing tool fulfill the remaining requirements of the user application scenario outlined in Section 2.6.

6.2 Analysis of Settings

In order to detect new settings in the future, its important to analyse the settings initally annotated by the users and detected in the system using the setting detection approach, as described in Chapter 4. This type of analysis can provide us with two key pieces of information. Firstly, we can attempt to characterise a setting, in terms of the length of time it occurs for, what location it occurs in, and how often it occurs. Secondly, we can analyse the patterns which arise over a selected time period. This should reveal insights into the way the user goes about his/her daily life. In particular, we can determine if a 'routine' exists for a user, if certain settings occur regularly or are just one-offs, and whether there are correlations between users. In order to achieve this, we can

User	Average length	Standard Deviation	Minimum	Maximum
User 1	00:55:45	01:11:12	00:07:00	05:44:00
User 2	00:47:44	00:50:04	00:01:00	04:03:00
User 3	00:45:45	01:33:24	00:01:00	08:33:00
User 4	00:57:54	00:49:15	00:02:00	03:18:00
User 5	01:47:55	01:30:58	00:05:00	06:11:00
Total	00:59:25	01:12:43	00:01:00	08:33:00

Table 6.1: The average length of time a setting occurs for.

analyse the annotated data provided by the users using the annotation tool (described in Section 4.4.1), as well as analysing the settings detected by our algorithm.

The first step is to attempt to characterise a setting in some fashion. Naturally, all settings are different, and will vary greatly over time. However, this initial analysis will give us an indication of what characterises the settings currently annotated by the users, as well as providing some pointers to assist us in automatically detecting settings in the future as new images are loaded into the Visual Diary. In Section 3.1, we defined a setting in terms of it's visual characteristics, namely, that they are images taken in the same real world location. Having detected the annotated settings provided by the user, we can also analyse them in terms of time. By analysing the detected settings we can establish the start and end time, average length of time, etc..

In Table 6.1, we calculated the average length of time a setting lasted for each user, as well as the total for all users. We also calculated the standard deviation to indicate whether or not any settings were significantly longer or shorter than the mean. The results indicate that for most users, the average setting length is between 45 - 60 minutes, and indeed the average for all users is just over 59 minutes. The one exception to this is *User 5*, whose average setting length is 01:47:55. *User 5* had the fewest settings, and tended to only annotate settings which had a significant amount of images associated with them and, therefore, covered large periods of time. The other users were less restrictive in their annotations, hence, their lower overal means.

The figures for the minimum and maximum length of a setting demonstrate that certain settings are significantly longer or shorter than the average. It is also interesting to see exactly what settings occur over a very short period of time and which settings occur over longer time periods. It's interesting to note that the longest settings across all users generally relate to either work, such as working on a computer, or travel, such as in a car or on a train. In addition, these settings occur regularly throughout all users collections. The shortest settings tend to be very varied across all users and encompass settings such as chatting to colleagues in a corridor, eating a sandwich, or

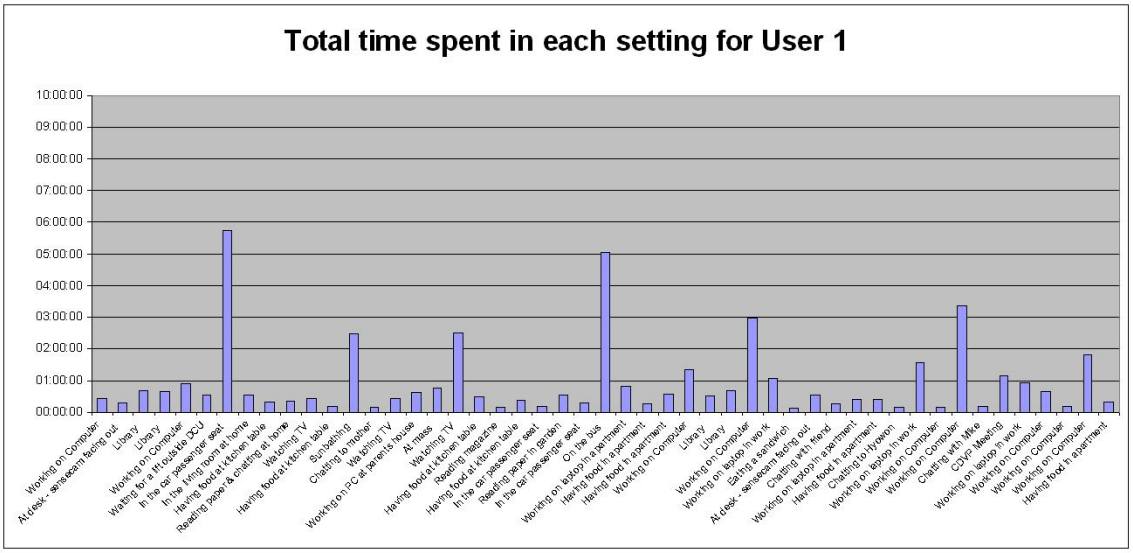


Figure 6.1: The length of time spent in different settings for User 1.

driving at night. The length of time users spent in various settings can be seen in Figures 6.1, 6.2, 6.3, 6.4, and 6.5. Analysing this information reveals that it is the routine events in daily life which occur regularly across all users (as expected). This encompasses activities such as eating, working, travel, and meeting friends and colleagues. Other activities are much more user specific, and include activities such as sunbathing, visiting a garage, giving a lecture, or making a presentation. Given the limited size of the collections analysed, these activities only occur once, however, they are activities one would expect to see arise in the future as more data is gathered.

Although these graphs give an indication of the amount of time a user spends in particular settings, on a day by day basis, it would also be useful to determine where the user spends most of their time over the entire collection of images analysed. We can obtain this information by summing the time spent in a particular setting over the time period analysed and expressing it as a percentage of the total time spent in all settings. This information can be seen in the pie-charts shown in Figures 6.6, 6.7, 6.8, 6.9, and 6.10. Again, these charts highlight the differences between regular (or routine) activities, and those which occur on a less frequent basis.

For example, if we examine Figures 6.8 and 6.10, we can see that both charts look extremely similar. For both of these users, an extremely large (65% for *User 5* and 73% for *User 3*) portion of their time was spent at work, working on their computer. The remaining settings for both users largely consist or routine events in a standard working week, such as reading in the evenings, watching TV, chatting with colleagues, etc.. Examining the images analysed for both of these users reveals that both users captured images over what we would term a 'normal' working week.

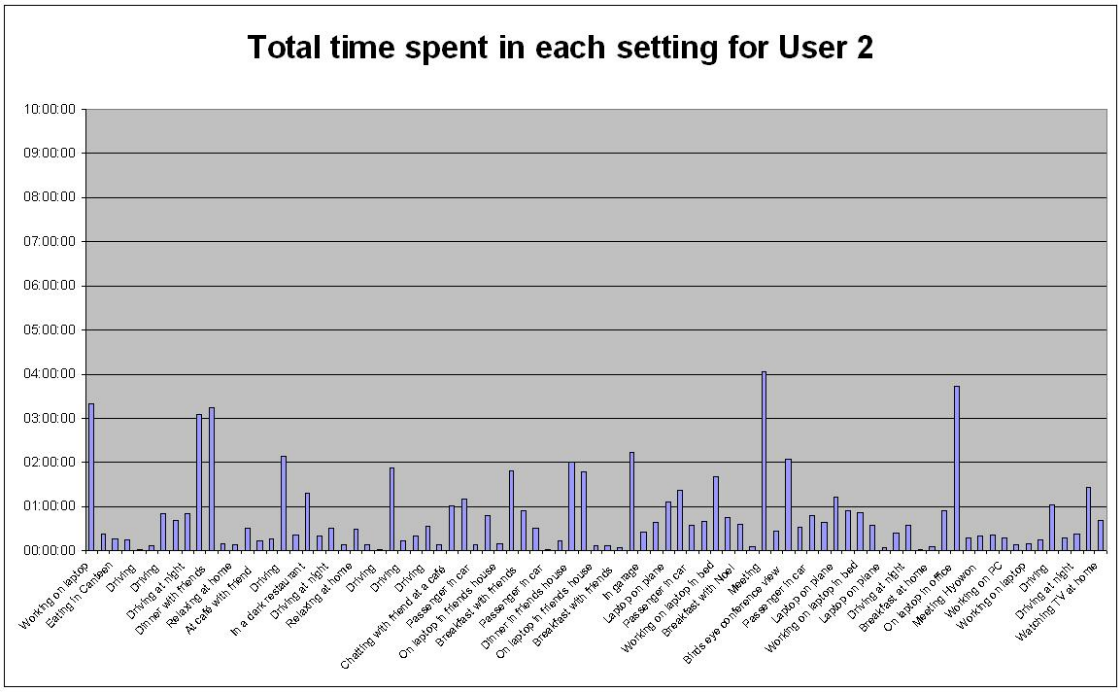


Figure 6.2: The length of time spent in different settings for User 2.

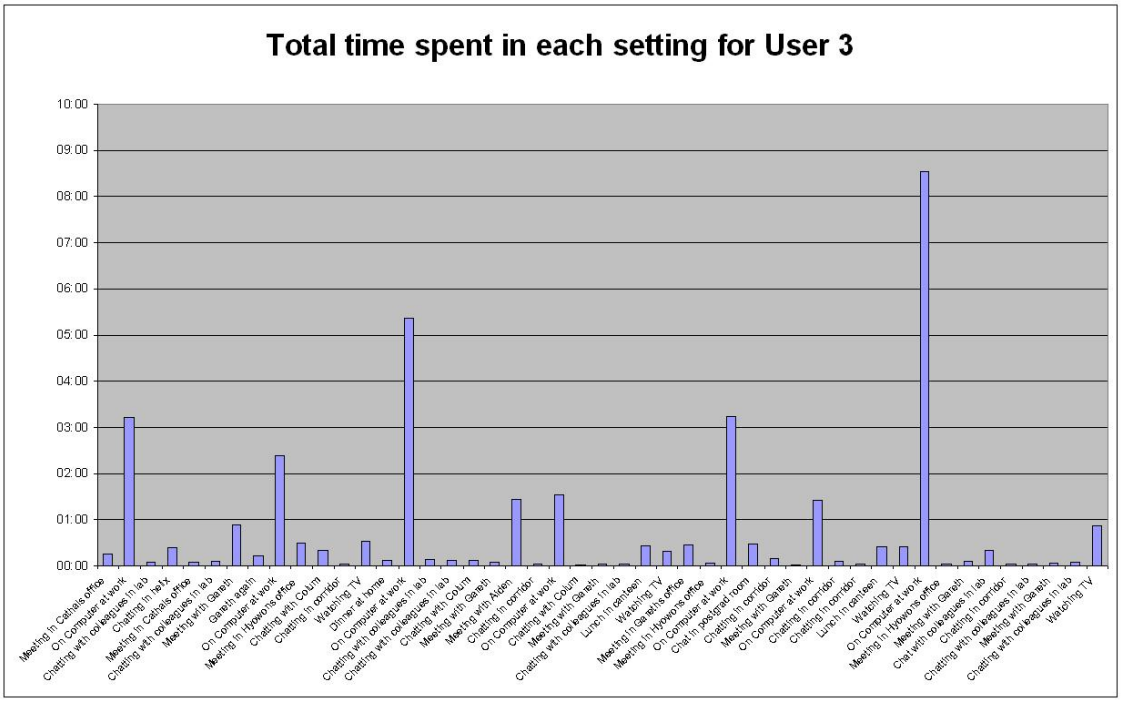


Figure 6.3: The length of time spent in different settings for User 3.

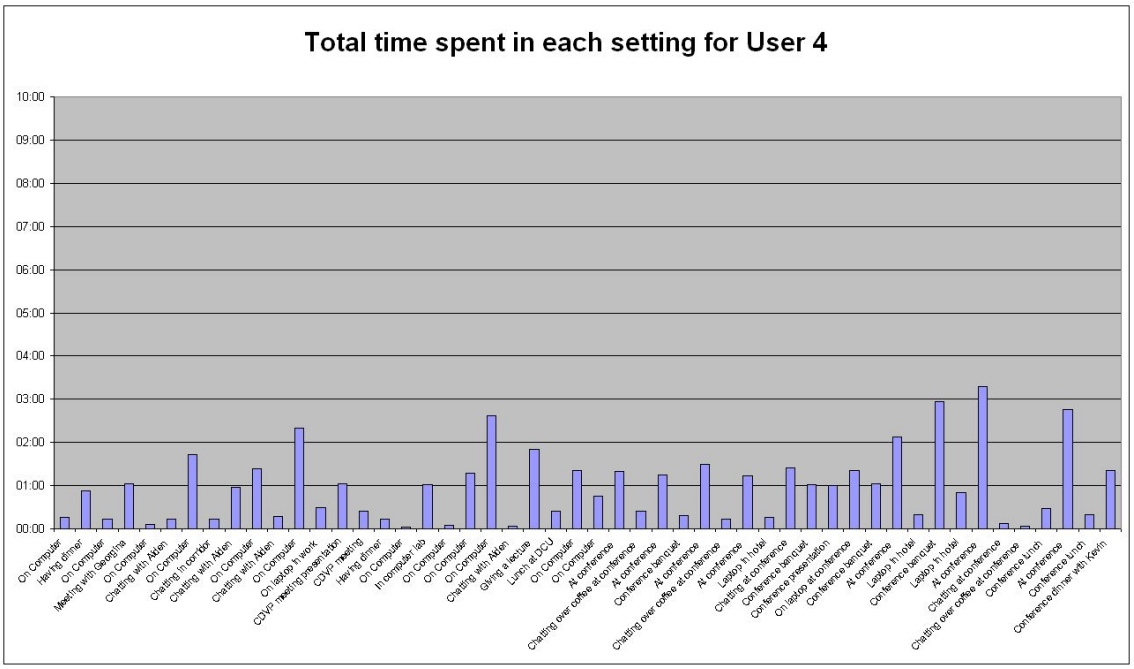


Figure 6.4: The length of time spent in different settings for User 4.

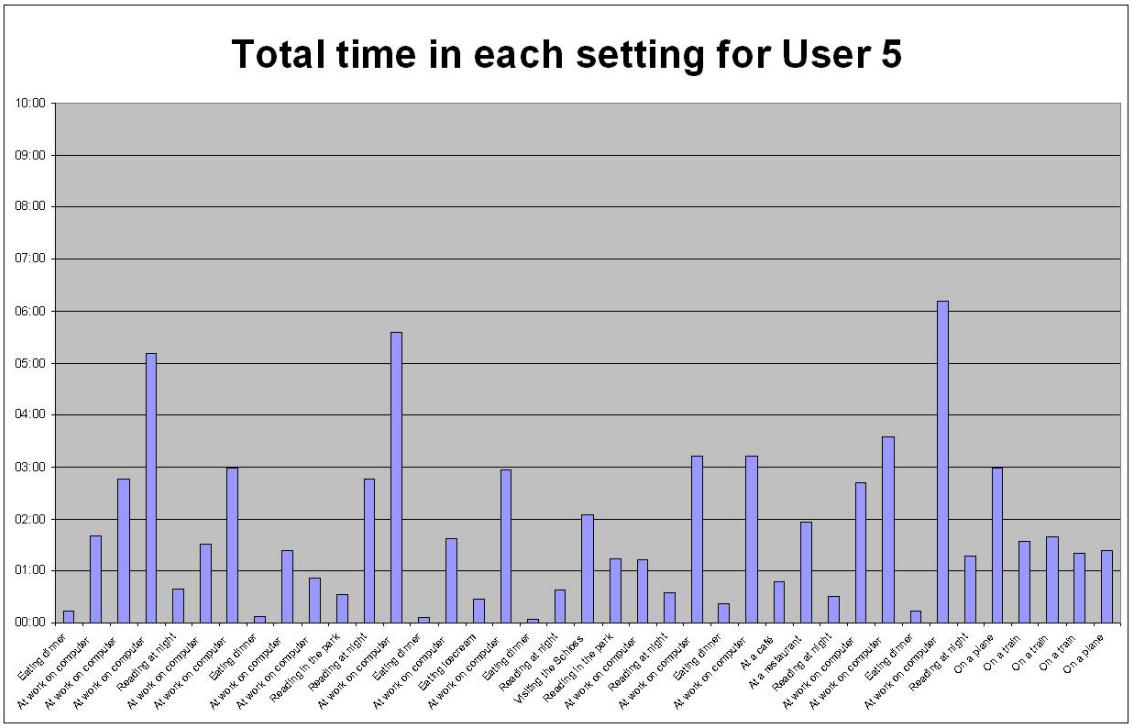


Figure 6.5: The length of time spent in different settings for User 5.

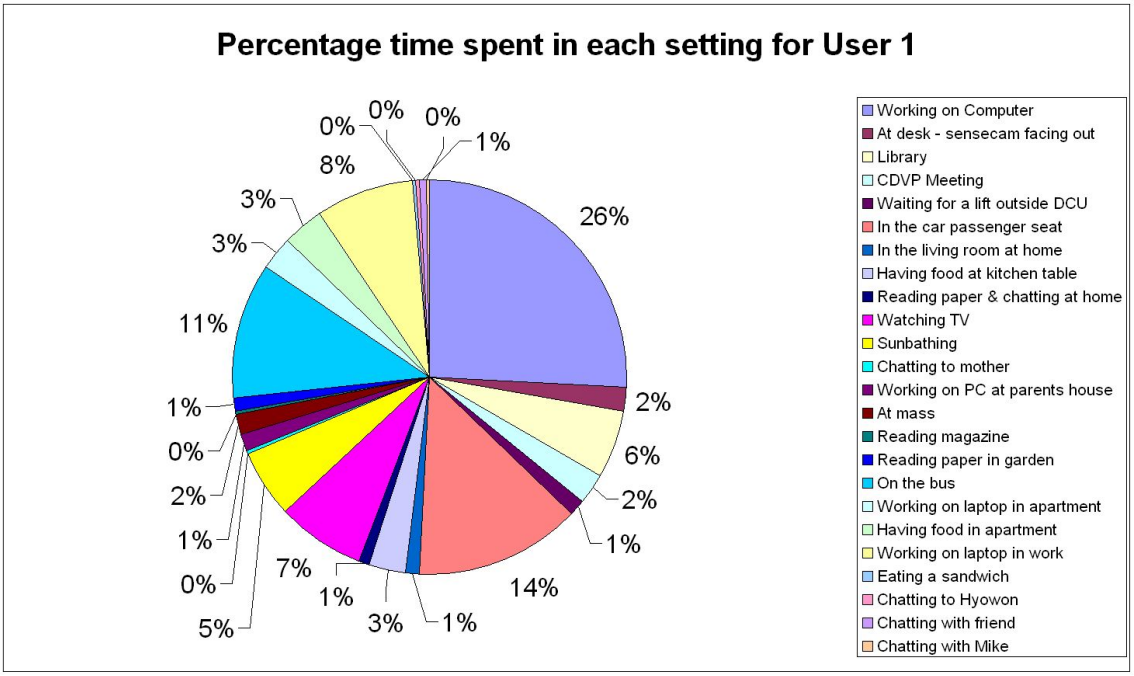


Figure 6.6: Percentage split between different settings for User 1

They got up, went to work, ate, worked, and went home. Very little activity outside of this routine occurred for both users over their collections. Indeed, if we look at Figures 6.13 and 6.15, we can clearly see the pattern of activity which occurred. These graphs show the starting and ending times for particular settings, in chronological order, over a period of one week. We only display a weeks worth of settings on these graphs in order to make them easier to view. The pattern for both users is very clear. *User 3* tends to start his day around 11:00, tends to lunch around 13:00-14:00 and tends to come back to work around 15:00 for the remainder of the evening. His day is interspersed with meetings and informal chats with work colleagues. He normally finishes work around 20:00 and his day normally ends around 22:00. *User 5* follows a similar pattern, although a different times. He tends to start his working day at around 8:00, lunches around 13:00, resumes work around 14:00, and tends to finish in the evenings between 17:00 and 19:00. His evening routine tends to involve reading, either in the park or at home, and he tends to turn off the camera between 22:00 and 23:00. For settings which occur on a routine basis, as is the case with these two users, the line graphs reveal the pattern of activity very effectively.

For other users, the pie-charts reveal a slighty different story. The images captured for all of these users was more varied in nature. Specifically, their collections involved periods at work, as well as periods in different locations, such as conferences, or visits to friends and family. The charts accurately demonstrate this. For example, in Figure 6.6 we can see that *User 1* spent most

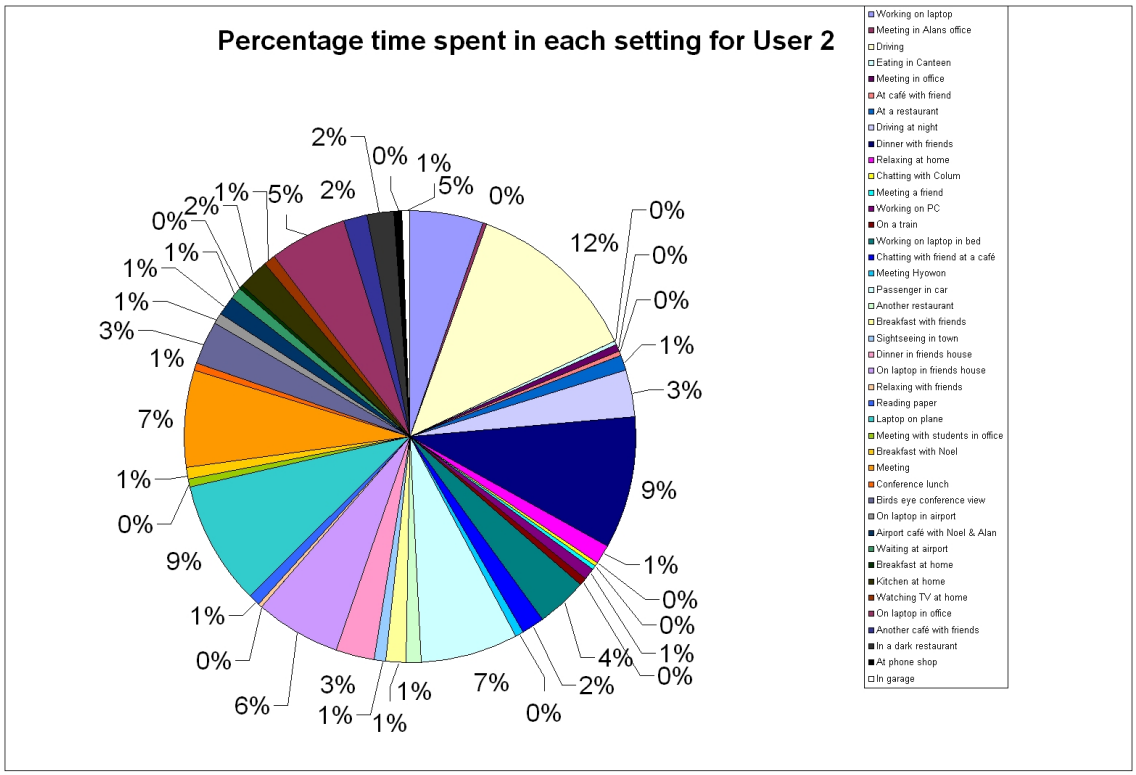


Figure 6.7: Percentage split between different settings for User 2

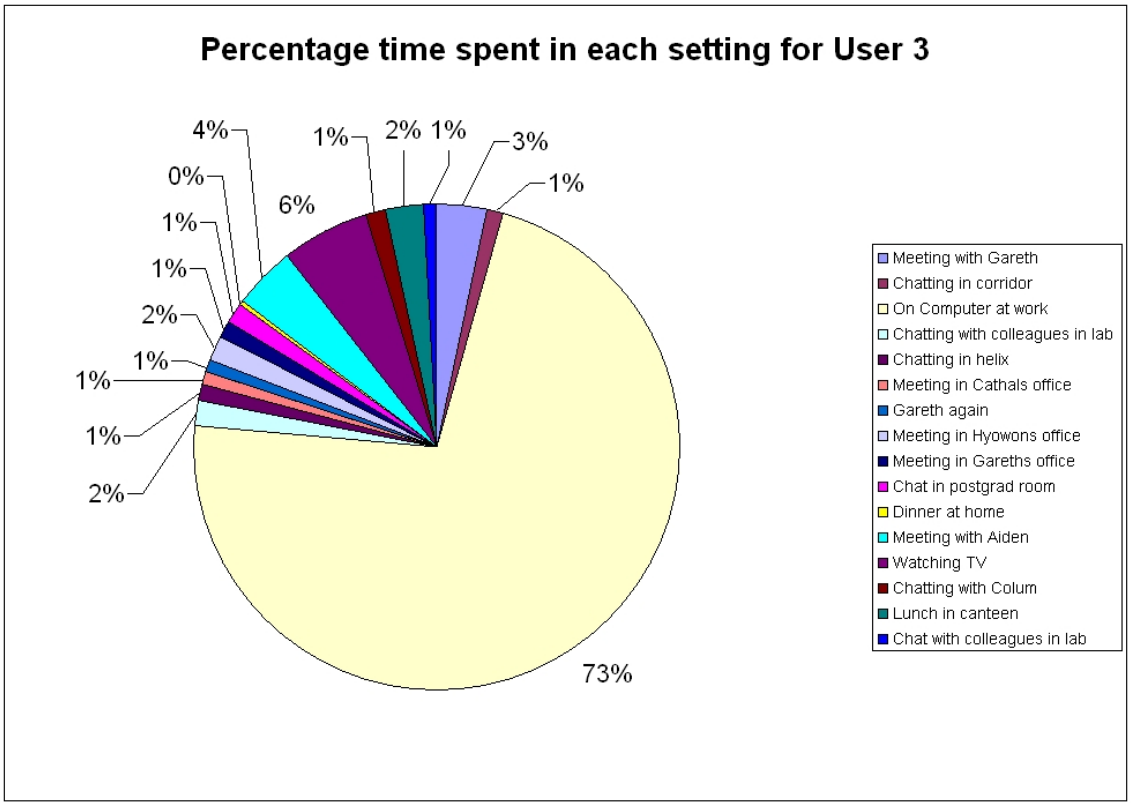


Figure 6.8: Percentage split between different settings for User 3

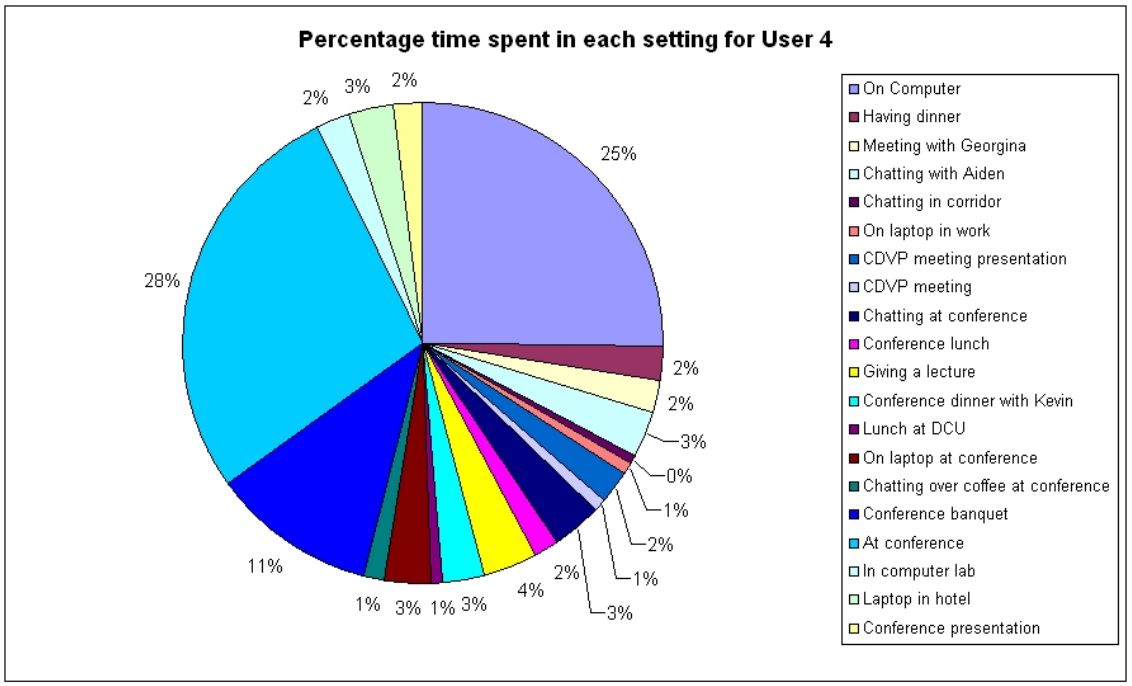


Figure 6.9: Percentage split between different settings for User 4

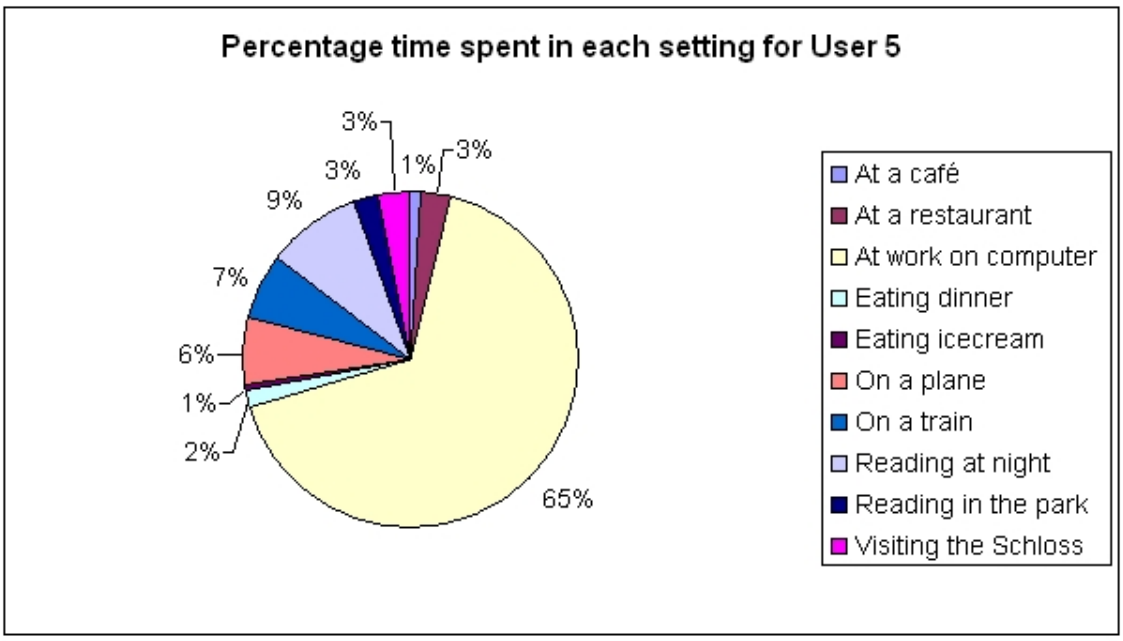


Figure 6.10: Percentage split between different settings for User 5

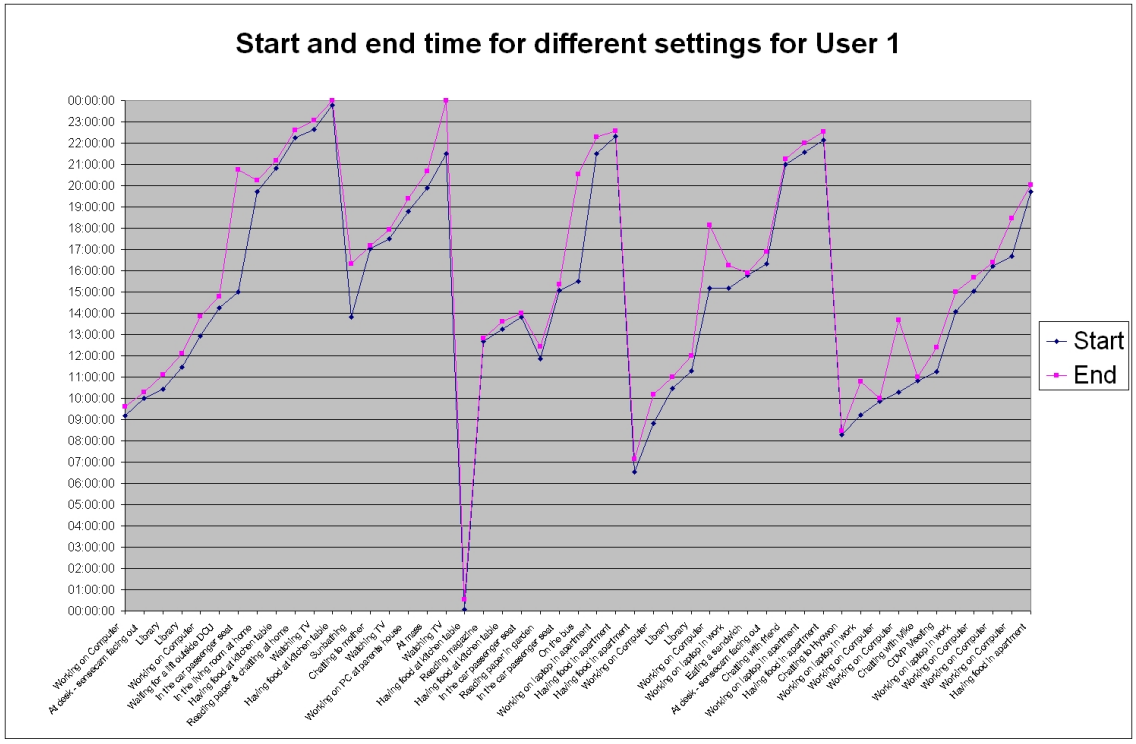


Figure 6.11: The starting and ending times for different settings for User 1

of his time at work (26%), or working on his laptop (8%), however, a significant amount of time was also spent travelling to and from his parents home. This time consisted of 14% in the car and 11% on the bus. In addition, much of the smaller settings (in percentage terms) were spent doing various activities at his parents home. These include attending church, sunbathing, watching TV, and chatting with parents. These settings only occur over the short period when the user was in this location. Again, as we mentioned previously, as this user's Visual Diary grows, these settings would be expected to crop up again.

For *User 2* and *User 4*, we can see something similar. Interestingly, *User 2* spent most of his time driving (12%) (shown in Figure 6.7). This appears to be due to the fact the *User 2* tends to work in many different locations due to the amount of travel he did during the time period analysed. For example, 9% of the time was spent using his laptop on a plane, 4% working on his laptop in bed, and 5% using the laptop in the office. *User 4* spent 25% of his time at work and 28% at a conference (shown in Figure 6.9). Again, the breakdown of other settings amongst these two users is indicative of the extremely varied nature of their lives during the image collections provided.

The line graphs showing the starting and ending times for *Users 1*, *2*, and *4* are shown in Figures 6.11, 6.12 and 6.14. Again, these show a general pattern of activity over the time period

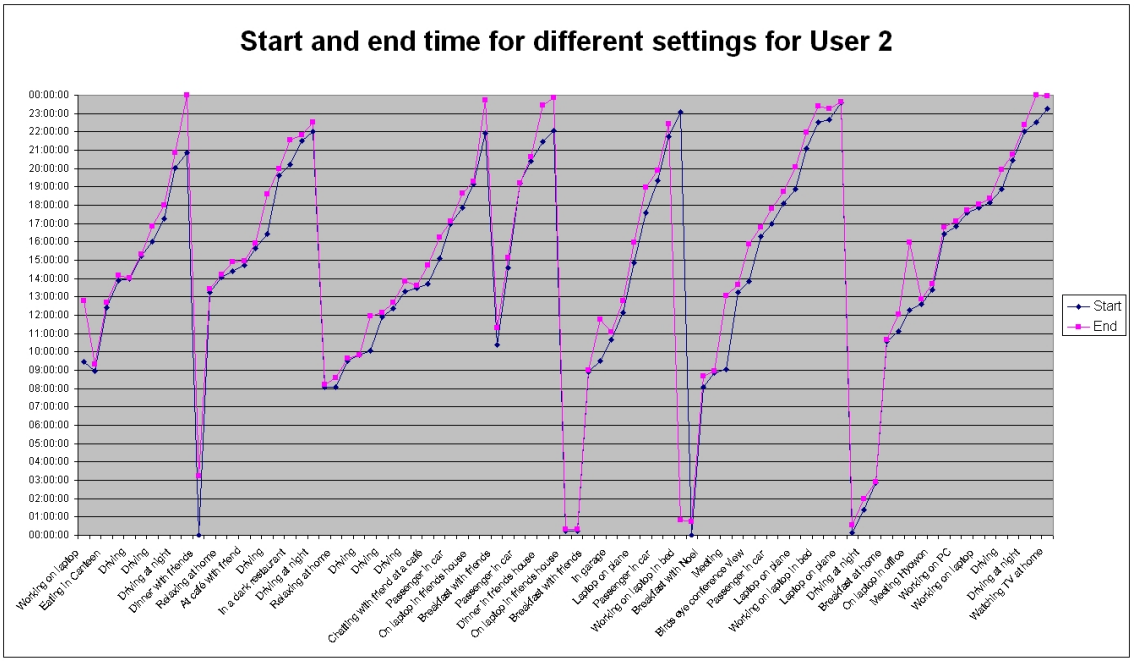


Figure 6.12: The starting and ending times for different settings for User 2

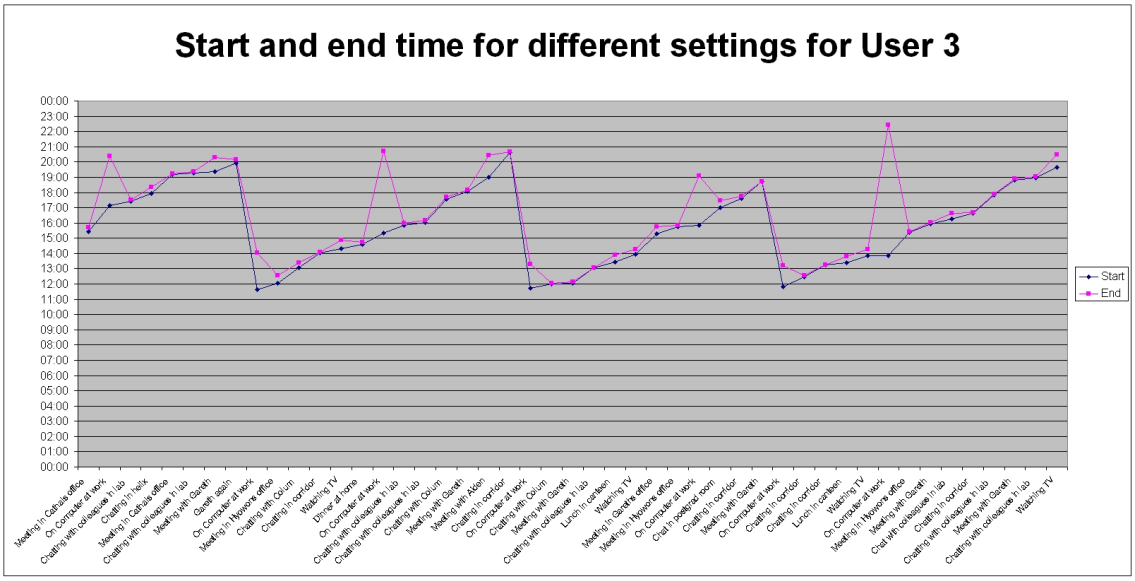


Figure 6.13: The starting and ending times for different settings for User 3

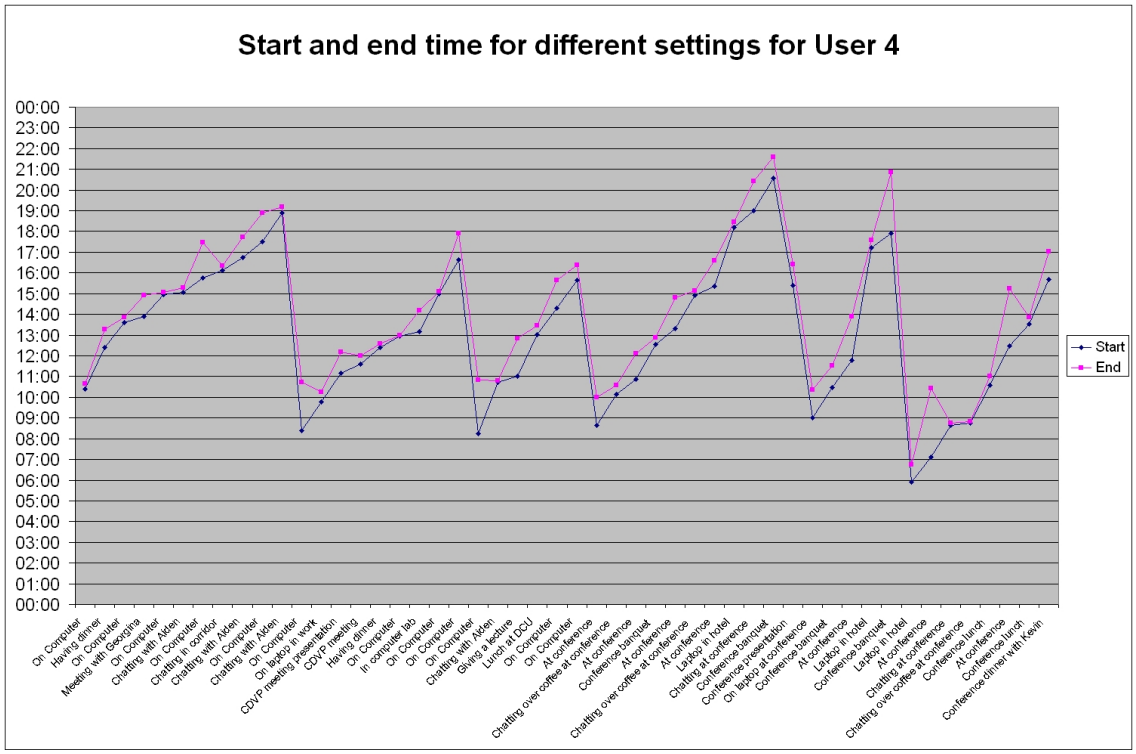


Figure 6.14: The starting and ending times for different settings for User 4

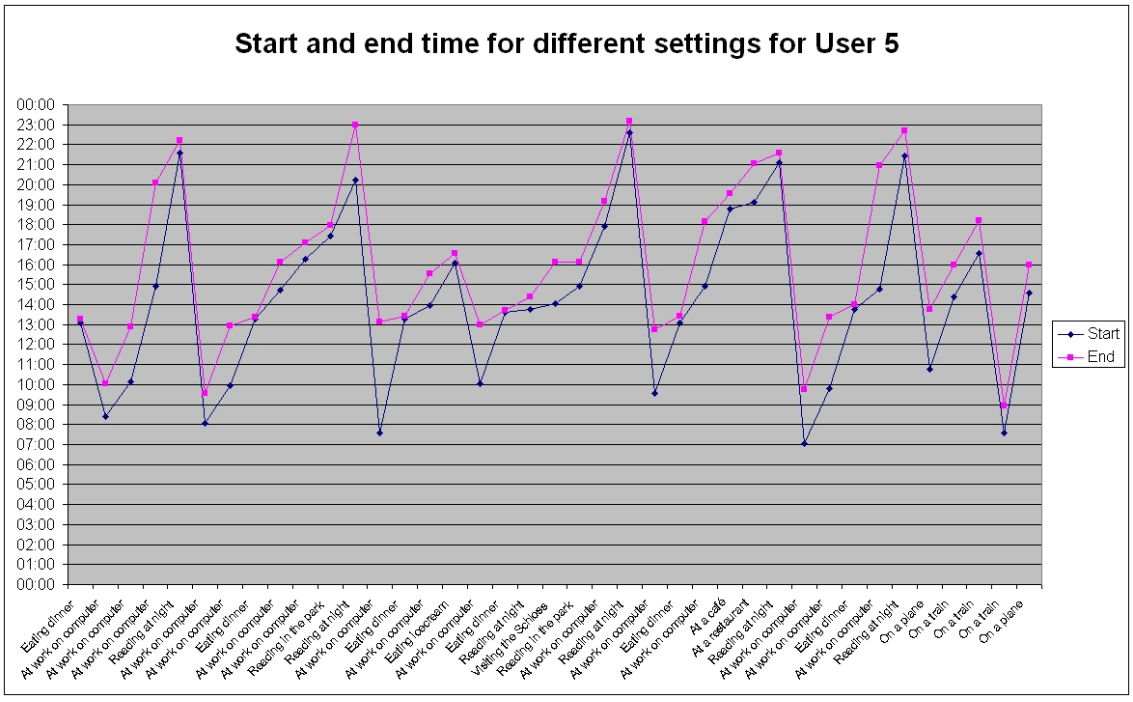


Figure 6.15: The starting and ending times for different settings for User 5

analysed, and certain trends do emerge. For example, *User 1* tends to start work early in the mornings between 6:00 and 8:00. *User 2* tends to keep very late hours, often dining with friends, or even working, after midnight. He also tends to drive at varying times throughout the day and night. This particular user did a lot of travelling during the collection analysed, and the unusual hours for certain activities (e.g. late meals, etc.) is indicative of a lifestyle outside of what might be considered the norm (and certainly at odds with the almost rigid routines displayed by *User 3* and *User 5*). *User 4* tended to engage in the same activities, even though he was travelling during this time. For example, he tended to get up at similar times, both at home and while travelling. He also tended to engage is similar activities in both locations (e.g. working, chatting to colleagues, etc.). This similarity in activity in probably due to the fact that the travel in this case was related to a conference the user attended. If it had been a vacation, or other leisurely activity, a different pattern may emerge.

6.2.1 Personalisation

The analysis described above facilitates the creation of personalised summaries of a user's day, week, month, or year. Rather than simply present each user with the same browsing interface, we can use the analysis of the different settings people experienced to present different summaries for each individual user. This would be an additional feature of the Visual Diary and not a replacement of the browsing interface.

For example, we present a summary of three different days for *User 5* in Figures 6.16, 6.17, & 6.18. These charts show that on the 9^{th} and 10^{th} April, *User 5* spent most of his time working (92% on the 9^{th} and 91% on the 10^{th}). The remainder of his day was spent eating (2% on both days) and reading (6% at home on the 9^{th} and 7% in the park on the 10^{th}). However, on the 12^{th} April, *User 5* only spent 47% of his day working. This may suggest that something else more interesting also occurred on this particular day and when we look at the remaining activities in that day we can see that he spent 24% of his time visiting a local castle and 28% of his time reading. This would suggest a more leisurely day for this particular user and the visit to the castle may be something the user wishes to mark as being important as it differs so much from the routine activities experienced in the other days activities. We discuss this further in Section 6.3.2.

By way of comparison, the daily summary for the 5^{th} April for *User 2* (shown in Figure 6.19) shows a very different pattern of activity than that for *User 5*. This user experienced a much larger variety of settings during this particular day, including periods travelling, meetings with

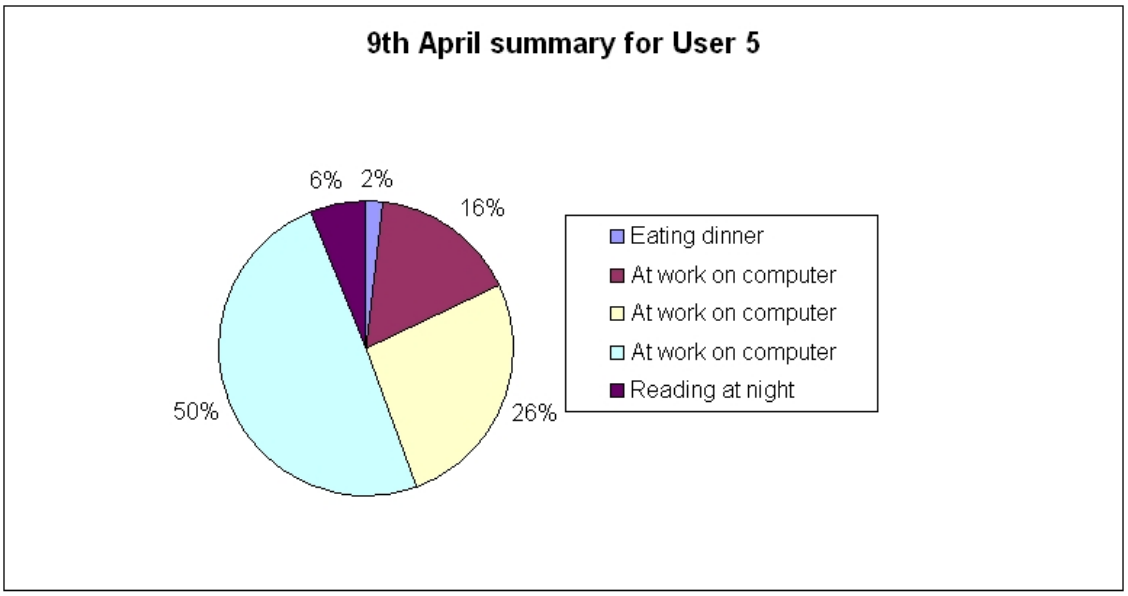


Figure 6.16: This image shows a summary of the settings experienced by User 5 on the 9^{th} April.

various colleagues, working in different locations, and shopping. As we discussed above, there is a significant difference between this user's summary and that of *User 5*. This would suggest that a more detailed analysis of this information can provide us with some real insight into a user's daily activities, thereby validating the importance of the detection of settings in a Visual Diary. In particular, it would be extremely useful if the user could use their personalised summaries to determine which settings are routine (and therefore not so interesting) and which settings are not routine (and therefore more interesting). It would also be useful to allow this information to dynamically change as the Visual Diary grows.

In order to achieve this, we need to examine the elements of a routine day (as shown in the analysis provided in Section 6.2 and in Figures 6.16 & 6.17). A routine day for *User 5* consists of three settings in the examples shown. These settings are working, eating, and reading. For the 9^{th} April, the first day the user has gathered images in this collection, we also know the start and end times of each settings. By calculating the median time, we can get a single representative time when each of these settings occurred. On the 9^{th} April, *User 5* spent 92% of their time working, 6% reading and 2% eating. The median time these settings occurred during the day was 09:13, 13:11 and 21:54. By examining the following days activities and comparing the statistics from the images during that day, we can determine if it was also a routine day. On the 10^{th} April, *User 5* spent 91% of their time reading, 7% reading and 2% eating. The median times these settings occurred at was 08:49, 13:20 and 21:36. This would suggest that these two days are extremely

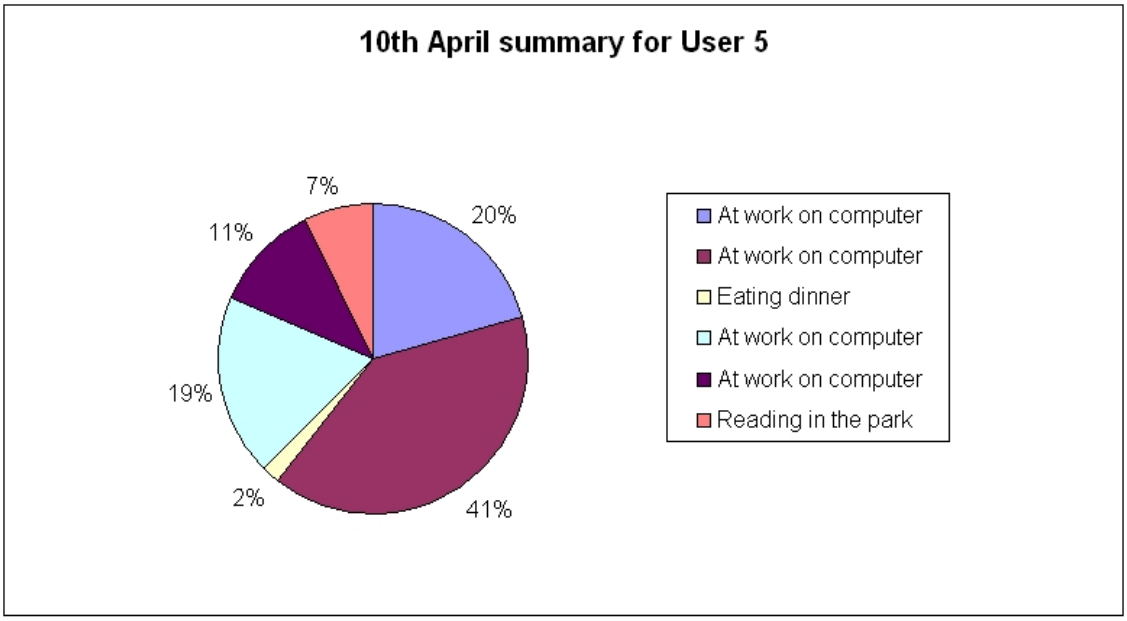


Figure 6.17: This image shows a summary of the settings experienced by User 5 on the 10^{th} April.

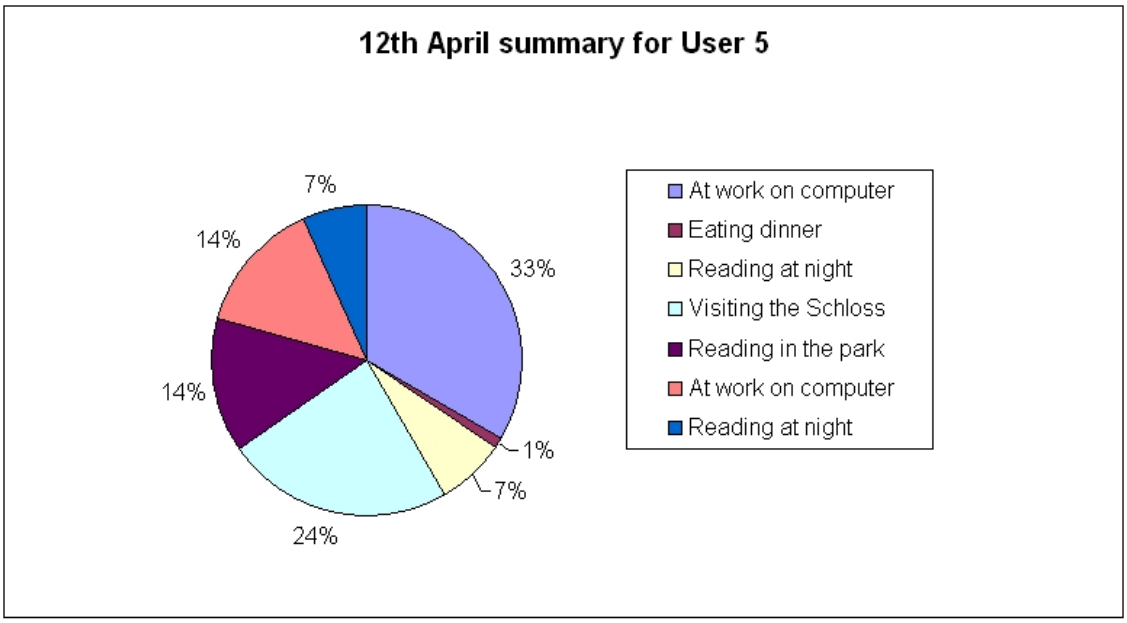


Figure 6.18: This image shows a summary of the settings experienced by User 5 on the 12^{th} April.

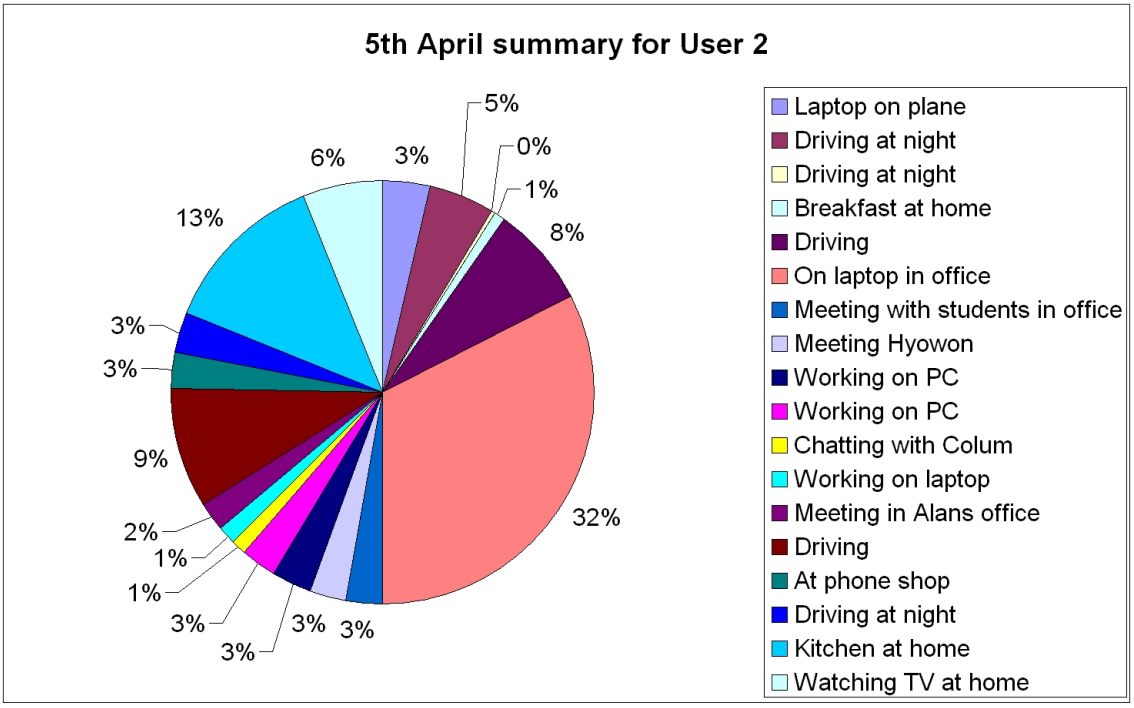


Figure 6.19: This image shows a summary of the settings experienced by User 2 on the 5^{th} April.

similar and could be classified as routine activites.

However, in order to definitively classify the settings experienced on the 10^{th} April as being similar to those on the 9^{th} April, we need to consider the times each setting occurred during the day. A similar combination of settings may arise on another day of the week, but this may not be part of a routine activity (e.g. working on the weekend). For this reason, we split the day into four categories. This allows us to allocate a setting to a particular category based on the median time. The four categories in question are: morning (6:00am - 11:59am); afternoon (12:00pm - 17:59pm); evening (18:00pm - 23:59pm); and night (00:00am - 5:59am). This will allow us to detect settings which occur at an unusual time. For example, in Figure 6.20 we can see the keyframes (see Section 5.2) from the summary of the 9^{th} and 10^{th} April. Clearly, these images are from the same settings, and both will be identified as routine (assuming the user has indicated they should be). However, if we examine keyframes from the 12^{th} April (see Figure 6.21), we can see that new settings have occurred which are not deemed to be part of the routine. Those settings which are different can then be flagged to the user to determine if they are important.

More formally, to determine whether a particular image is part of a routine setting or a non-routine setting, we can model the attributes outlined above. In order to determine whether an image is part of a routine event or not, we need to examine a number of attributes: setting, time,

(a) User 5 - 9^{th} April

(b) User 5 - 10^{th} April

Figure 6.20: Summary keyframes from two different days for User 5

Figure 6.21: This image shows a summary of the settings experienced by User 5 on the 12^{th} April.

and day. For a particular image to be considered routine, it should be part of a specific setting and occur at a particular time on a specific day. We can model the data in this fashion to determine whether specific settings are part of routine activities or not (see Table 6.2).

In order to determine if an image is part of a routine setting, we want to ask ourselves the question "if we observe an image of a user working on their computer on a Monday afternoon, is it likely to be part of a routine setting, based on the observed data sample? If so, in future, classify images of the user working on their computer on Monday afternoons as a routine setting." Given that we have already detected specific settings in the Visual Diary, we can use this information (as described above) to achieve this. In this first instance, we can automatically analyse the detected settings using the model outlined above. By analysing the date/time information in the database, we can easily determine which settings are routine across a given time period. This information can then be used to build models for each setting similar to that shown in Table 6.2. It's also important to note that these models can also be automatically updated by routinely performing this analysis as new images are uploaded to the Visual Diary. User feedback can also be incorporated into this process to validate the automatically detected information and to further refine the

Setting	Time	Day	Outcome
Computer	Morning	Monday	Yes
Computer	Morning	Tuesday	Yes
Computer	Morning	Wednesday	Yes
Computer	Morning	Thursday	Yes
Computer	Morning	Friday	Yes
Computer	Morning	Saturday	No
Computer	Morning	Sunday	No
Computer	Afternoon	Monday	Yes
Computer	Afternoon	Tuesday	Yes
Computer	Afternoon	Wednesday	Yes
Computer	Afternoon	Thursday	Yes
Computer	Afternoon	Friday	Yes
Computer	Afternoon	Saturday	No
Computer	Afternoon	Sunday	No
Computer	Evening	Monday	Yes
Computer	Evening	Tuesday	Yes
Computer	Evening	Wednesday	Yes
Computer	Evening	Thursday	Yes
Computer	Evening	Friday	Yes
Computer	Evening	Saturday	No
Computer	Evening	Sunday	No
Computer	Night	Monday	No
Computer	Night	Tuesday	No
Computer	Night	Wednesday	No
Computer	Night	Thursday	No
Computer	Night	Friday	No
Computer	Night	Saturday	No
Computer	Night	Sunday	No

Table 6.2: A portion of a single setting modeled for *User 5*. We only present one setting in this table as the information required to model all settings is too large to display. Using this information, we can work out the likelihood of particular settings occurring given specific scenarios. We can also use this information to determine whether specific images belong to routine or more interesting events.

detection of routine settings (see Section 6.4 for further discussion).

Once these models have been constructed, a classifier can be used to determine whether new images from detected settings are part of a routine event or not. In this instance, we restrict the classifier to the binary case, however, one could envisage more complex classes emerging as the data in the Visual Diary grows further. A simple method of performing this analysis is to allow each attribute to make contributions to the final decision that are equally important and independent of one another, given the class. One extremely effective method of achieving this type of classification is Naïve Bayes [218]. Despite it's simplicity, it has been shown to work extremely effectively for these types of problems [233]. Naïve Bayes does not require lots of observations

for each possible combination of the variables (see Table 6.2). Rather, the variables are assumed to be independent of one another and, therefore, the probability that an image of a user working on a computer, taken on a Monday afternoon, will be part of a routine setting can be calculated from the independent probabilities that the image is taken in that setting, that it was captured in the afternoon, and that it was also captured on a Monday. In other words, Naïve Bayes classifiers assume that the effect of a variable value on a given class is independent of the values of other variables. This assumption is called class conditional independence. It is made to simplify the computation and in this sense is considered to be naïve.

This assumption is a fairly strong assumption and is often not applicable. However, bias in estimating probabilities often may not make a difference in practice - it is the order of the probabilities, not their exact values, that determine the classifications. Studies comparing classification algorithms have found the Naïve Bayesian classifier to be comparable in performance with classification trees and with neural network classifiers. They have also exhibited high accuracy and speed when applied to large databases [233]. Therefore, we believe the use of Naïve Bayes is justified in this thesis.

As an example, by modelling the entire data available for *User 5* in this fashion, we can calculate the likelihood and probability of specific settings being routine or not, given the previously automatically detected and modeled data. For example, given an image of a computer which occurs on a Monday in the afternoon, it has a likelihood of being Yes (i.e. part of a routine) of 0.0357 and a likelihood of No (i.e. not part of the routine) of 0.0055. The probability of Yes is 86.65% and the probability of No is 13.35%. In comparison, given an image which occurs in a setting where the user is on an aeroplane on a Monday afteroon, the likelihood of this being part of the normal routine is 0.00987, which is a probability of 24.91%. As we can see, by calculating the likelihood or probabilities of different settings in this fashion, we can determine how likely it is that images from new settings are part of the normal routine. The results for other settings for this user, and for a different scenario, can be seen in Tables 6.3 & 6.4. However, these results are based on an analysis of the entire collection available for *User 5*. Given that, they are extremely difficult to comprehensively evaluate. We discuss this issue further in Section 6.2.1.2.

6.2.1.1 Bayes Theorem

Let X be the data record (case) whose class label is unknown. Let H be some hypothesis, such as "data record X belongs to a specified class C". For classification, we want to determine $P(H|X)$

Setting	Likelihood of yes	Likelihood of No	Probability of yes	Probability of No
Computer	0.0357	0.0055	86.65%	13.35%
Plane	0.0987	0.02976	24.91%	75.09%
Visiting Castle	0.0987	0.02976	24.91%	75.09%
Reading in bed	0.03571	0.00085	97.68%	2.32%
Eating ice cream	0.0987	0.02976	24.91%	75.09%
Eating dinner	0.02777	0.000295	98.97%	1.03%
Dining in a restaurant	0.00085	0.03571	2.32%	97.68%
On a train	0.0987	0.02976	24.91%	75.09%
Reading in park	0.00793	0.02819	21.95%	78.05%
In a cafe	0.0793	0.02819	21.95%	78.05%

Table 6.3: This table shows the likelihood and probability of a particular image being part of the routine or not, given that it was captured on a Monday afternoon

Setting	Likelihood of yes	Likelihood of No	Probability of yes	Probability of No
Computer	0.00056	0.02197	2.49%	97.51%
Plane	0.03571	0.01488	70.59%	29.41%
Visiting Castle	0.03571	0.01488	70.59%	29.41%
Reading in bed	0.000303	0.03571	0.84%	99.16%
Eating ice cream	0.03571	0.01488	70.59%	29.41%
Eating dinner	0.01587	0.01879	45.79%	54.21%
Dining in a restaurant	0.03571	0.00085	97.68%	2.32%
On a train	0.03571	0.01488	70.59%	29.41%
Reading in park	0.0555	0.000219	99.61%	0.39%
In a cafe	0.0555	0.000219	99.61%	0.39%

Table 6.4: This table shows the likelihood and probability of a particular image being part of the routine or not, given that it was captured on a Saturday evening

- the probability that the hypothesis H holds, given the observed data record X. $P(H|X)$ is the posterior probability of H conditioned on X. For example, the probability that an image taken by a user at work on their computer is a routine event, given the condition that it is part of a setting of images taken at work, and was captured at a specific time on a specific day. In contrast, $P(H)$ is the prior probability, or apriori probability, of H. In this example, $P(H)$ is the probability that any given data record is a routine setting, regardless of how the data record looks. The posterior probability, $P(H|X)$, is based on more information (such as background knowledge) than the prior probability, $P(H)$, which is independent of X.

Similarly, $P(X|H)$ is the posterior probability of X conditioned on H. That is, it is the probability that X is an image of the user working on their computer at work on a Monday afternoon given that we know that it is true that X is a routine setting. $P(X)$ is the prior probability of X (i.e. it is the probability that a data record from our set of images is of a computer and taken on a Monday afternoon). Bayes theorem is useful in that it provides a way of calculating the posterior

probability, $P(H|X)$, from $P(H)$, $P(X)$, and $P(X|H)$. Bayes theorem is:

$$P(H|X) = P(X|H)P(H)/P(X) \tag{6.1}$$

6.2.1.2 Experimental Results

Given the limited data available, we are somewhat restricted in performing our experiments and evaluating the results using this technique. In order to get around this problem, we will analyse different portions of the data provided by *User 5*. This will allow us to analyse a single weeks data, automatically detect the routine settings in that week (based on the known detected settings), and subsequently build a model for that weeks data. When new data is loaded to the Visual Diary (in this case, the subsequent weeks data for *User 5*), we can use Naïve Bayes to classify the detected settings as routine or not. The results can then be evaluated against a groundtruth of this data (i.e. the settings detected in the months collection have been annotated into routine and non-routine settings).

The first stage in this process is to automatically detect routine settings in the first weeks images. Given that the settings have already been detected, we analyse the summary keyframes used to represent each setting in the daily summary generated for each particular day's images (see Section 6.4). In this particular experiment, *User 5* encountered 19 different settings during this particular week. As an initialisation step, we need to automatically determine which settings within this weeks images are routine or not. This initialisation step is necessary in order to build the first model within the system for this image set. In order to determine if any of these settings are routine, we analyse them using the criteria for a routine setting previously described. In particular, we are interested in settings which reoccur at the same time during the day across this particular week. This provides an initial estimation of what might be considered routine in the users collection.

Given this criteria, the system detected a number of routine settings in this particular week. This would seem to tie in with the seemingly repetitive nature of the activities carried out by *User 5* whilst he was collecting images (see Figure 6.15). The results of this analysis can be seen in Table 6.5. This table shows the median time specific settings occurred during this particular week. Those which reoccur during the same time period have been automatically classified as a routine setting. In this particular example, most of the settings encountered reoccured at certain points throughout the week, and at similar times during the day. However, setting number 4 occurs in the evening on

Date	Time	Setting	Period	Outcome
09/04/2007	10:23	6	Morning	Yes
09/04/2007	13:21	2	Afternoon	Yes
09/04/2007	21:49	1	Evening	Yes
10/04/2007	12:52	6	Afternoon	Yes
10/04/2007	12:56	4	Afternoon	Yes
10/04/2007	13:44	2	Afternoon	Yes
11/04/2007	17:23	6	Afternoon	Yes
11/04/2007	19:12	4	Evening	No
12/04/2007	12:11	6	Afternoon	Yes
12/04/2007	12:53	4	Afternoon	Yes
12/04/2007	21:04	1	Evening	Yes
13/04/2007	11:06	6	Morning	Yes
13/04/2007	13:55	2	Afternoon	Yes
14/04/2007	10:53	6	Morning	Yes
14/04/2007	17:50	2	Afternoon	Yes
14/04/2007	19:27	3	Evening	No
15/04/2007	12:36	6	Afternoon	Yes
15/04/2007	16:07	4	Afternoon	Yes
15/04/2007	19:41	1	Evening	Yes

Table 6.5: Analysis of the settings detected during a single week for *User 5*. This table shows which settings the system has detected as being part of the routine, based solely on an analysis of the settings that reoccur during the same time period during this particular week.

the 11^{th} April, but occurs during the afternoon on other days. Perhaps more interestingly, setting number 3 only occurs once during the entire week. This may be a more interesting setting which can be highlighted to the user via the user interface (as described in Section 6.4).

Given this initial analysis of a week's data, we can subsequently build a model for that week similar to that shown in Table 6.2. We can see a portion of this model for setting 1 in Table 6.6. When new images are loaded to the Visual Diary (in this case, the subsequent data available for *User 5*), this model allows us to calculate the likelihood and probability of these images being part of the routine or not, using Naïve Bayes. Note, that these new images will have been initially processed to detect settings, generate summary information, etc., as outlined elsewhere in this thesis. The final stage of the analysis of new images is, therefore, the classification of these images as routine events using Naïve Bayes. Given a second week's images, the results from this analysis can be seen in Table 6.7. Although 19 settings were encountered during the first weeks images, this actually only constituted 5 unique settings. Therefore, we can only attempt to predict whether these particular settings are routine in the images analysed for week 2, and as the results show, only 4 settings reoccurred during this second week. Of these, settings two and six would be classified as routine settings, but settings one and three would not. Note, that setting number three

which only occurred once during the first week's images, did not occur again during the second week. Hence, it is excluded from the analysis.

Setting	Time	Day	Outcome
1	Morning	Monday	No
1	Morning	Tuesday	No
1	Morning	Wednesday	No
1	Morning	Thursday	No
1	Morning	Friday	No
1	Morning	Saturday	No
1	Morning	Sunday	No
1	Afternoon	Monday	No
1	Afternoon	Tuesday	No
1	Afternoon	Wednesday	No
1	Afternoon	Thursday	No
1	Afternoon	Friday	No
1	Afternoon	Saturday	No
1	Afternoon	Sunday	No
1	Evening	Monday	Yes
1	Evening	Tuesday	No
1	Evening	Wednesday	No
1	Evening	Thursday	Yes
1	Evening	Friday	No
1	Evening	Saturday	No
1	Evening	Sunday	Yes
1	Night	Monday	No
1	Night	Tuesday	No
1	Night	Wednesday	No
1	Night	Thursday	No
1	Night	Friday	No
1	Night	Saturday	No
1	Night	Sunday	No

Table 6.6: This table shows the model generated after the analysis of the data loaded for week 1 for setting 1

Given this information, we can subsequently update the initial model to include the settings analysed above, as well as the new settings introduced in the images loaded to the Visual Diary in week 2. This provides an updated model with which to calculate the probabilties and thus allows the process to dynamically change over time as images are loaded to the Visual Diary. The results of this analysis for images loaded in week 3 can be seen in Table 6.8. Again, only four settings are analysed for the images loaded in week 3 as these are the only settings which reoccurred during this weeks images. After updating our model (i.e. it now reflects the analysis performed on the first two weeks images), we can see that settings one and two are now classified as routine. Setting four and six are not (although the issue with setting six is marginal at this stage at 48.41%).

Setting	Likelihood of Yes	Likelihood of No	Probability of Yes	Probability of No
1	0.00175	0.03	5.51%	94.49%
2	0.02678	0.01785	60.00%	40.00%
4	0.001653	0.023809	6.49%	93.51%
6	0.020408	0.015306	57.14%	42.86%

Table 6.7: This table shows the likelihood and probability of new images loaded to the Visual Diary being part of the routine or not, given that they were captured on a Monday afternoon. The images loaded are from the 2^{nd} week of the user's collection.

Setting	Likelihood of Yes	Likelihood of No	Probability of Yes	Probability of No
1	0.0408	0.0102	80.00%	20.00%
2	0.03571	0.00487	88.00%	12.00%
4	0.001295	0.013605	8.69%	91.31%
6	0.01587	0.01691	48.41%	51.59%

Table 6.8: This table shows the likelihood and probability of new images loaded to the Visual Diary being part of the routine or not, given that they were captured on a Monday afternoon. The images loaded are from the 3^{rd} week of the user's collection.

When the user loads the images captured in week four, the model is updated once again to reflect the current collection. Once again, the images in week four are analysed to determine which images are part of the routine elements of this users collection. In Table 6.9 we can see the updated model for setting one at this stage of the analysis. This table is included to demonstrate the changes in the model for this particular setting since the first weeks images were loaded to the Visual Diary (shown in Table 6.6). At this stage of the analysis, images which are in settings one, two, four, and six, and which occur on a Monday afternoon, are all deemed to be part of the normal routine for this user when analysed over the entire month's collection. In addition, a new setting has occurred during this week, setting nine, which would not be considered routine. Note that many other scenarios could be considered (e.g. Saturday morning, Friday night, etc.). The scenario of an image occurring on a Monday afternoon was picked at random to demonstrate the process involved.

The final stage in this process is to evaluate the classifications provided by the Naïve Bayes classifier. In order to achieve this, the images provided by *User 5* were annotated to identify those which are considered to be part of the routine and those which are not. These annotations provide a ground truth with which we can objectively evaluate the classifier and we use the classification error, described in Section 4.3.1, to achieve this. The results can be seen in Table 6.11. The initial results after week 2 are relatively disappointing, however, it's worth remembering that this is based on a simple analysis of the date and time information of the settings involved. In subsequent weeks,

Setting	Time	Day	Outcome
1	Morning	Monday	No
1	Morning	Tuesday	No
1	Morning	Wednesday	No
1	Morning	Thursday	No
1	Morning	Friday	No
1	Morning	Saturday	No
1	Morning	Sunday	No
1	Afternoon	Monday	Yes
1	Afternoon	Tuesday	Yes
1	Afternoon	Wednesday	Yes
1	Afternoon	Thursday	Yes
1	Afternoon	Friday	No
1	Afternoon	Saturday	No
1	Afternoon	Sunday	No
1	Evening	Monday	Yes
1	Evening	Tuesday	No
1	Evening	Wednesday	No
1	Evening	Thursday	Yes
1	Evening	Friday	No
1	Evening	Saturday	No
1	Evening	Sunday	Yes
1	Night	Monday	No
1	Night	Tuesday	No
1	Night	Wednesday	No
1	Night	Thursday	No
1	Night	Friday	No
1	Night	Saturday	No
1	Night	Sunday	No

Table 6.9: This table shows the model generated after the analysis of the data loaded for week 4 for setting 1

the classification improves as the model itself improves. One would not expect the classification error to fall much further because as the data set grows, the number of settings analysed gets larger, thus increasing the overall complexity.

The results are encouraging as they demonstrate that a very detailed analysis of a user's pattern of behaviour can be determined using information which is readily available as a result of the setting detection process. As previously mentioned, the identification of routine settings can be displayed to the user on the user interface. We can also facilitate the user in refining these results further and we discuss this issue in more detail in Section 6.4. In future, one could envisage a more complex classifer, with different user profiles arising over time. This would allow settings to be classifed into numerous different categories, besides routine and other. For now, we focus on detecting routine settings and highlighting the remaining ones to the user. Once larger datasets

Setting	Likelihood of Yes	Likelihood of No	Probability of Yes	Probability of No
1	0.0408	0.0102	80.00%	20.00%
2	0.03571	0.00487	88.00%	12.00%
4	0.02232	0.01071	67.57%	32.43%
6	0.03571	0.00793	81.83%	18.17%
9	0.00595	0.02747	17.80%	82.20%

Table 6.10: This table shows the likelihood and probability of new images loaded to the Visual Diary being part of the routine or not, given that they were captured on a Monday afternoon. The images loaded are from the 4^{th} week of the user's collection.

User	Classification Error
Week 2	0.3888
Week 3	0.1875
Week 4	0.0952
Average	0.2238

Table 6.11: The classification error for the detection of routine settings as new images are loaded on a week by week basis.

are available, this analysis can be scaled over different time periods so that we can analyse the activities over a week, month, year, or over an entire image collection. This allows the user to quickly determine which settings occur regularly, which are infrequent, and which are important.

6.3 Managing the Growth of a Visual Diary

In order to allow the Visual Diary to evolve as new images are captured and loaded into the system, we need to find a method of flagging potential new settings to the user. Once these settings have been flagged, the system can then analyse them using the approach described in Section 4.3.3 to locate other images from the same setting in the Visual Diary. In order to detect potential settings, we rely on a number of features charateristic of settings as presented in this thesis (and briefly described in Section 6.2). The first is their visual characteristics. As images from the same setting should all come from the same location, there should be a high level of visual similarity between sucessive images in the Visual Diary. Furthermore, in order to flag potential new settings to the user when images are loaded, we can safely make the assumption that the images will also be temporally aligned. For example, if a user is loading a new day's images into their Visual Diary, we can assume that images from the settings they've experienced (e.g. eating, working, etc.) will all occur together - i.e. eating dinner images will be together, working images will occur together in batches throughout the day, etc.. We can also suggest that these images will not have undergone an extremely significant amount of transformations, as is possible with images gathered from

different settings gathered over an extended period of time. Subtle changes will of course occur (due to lighting, movement, etc.), but on the whole, the images should be visually very similar.

In conjunction with these properties, we have established in Section 6.2 that the average length of a setting across all users is 00:59:25, with a standard deviation of 01:12:43. While we acknowledge that many settings are both longer and shorter than the average, we can utilise this timing information when flagging new settings for the user. For the purposes of these experiments, we decided to only flag a potential setting to the user when it is at least x minutes in length. Given the high standard deviation outlined above, a fixed threshold seems inappropriate. Therefore, a more effective method is to leverage the underlying event structure of the images used in the user interface (described in Section 5.2) to determine a dynamic threshold. Based on the length of a particular event, we can determine an appropriate threshold, x, as part of the process of automatically detecting a new setting.

In order to detect and flag new settings, we use an approach similar to that described in Appendix D.0.3.1. This approach uses the SIFT features, which have been discussed in detail in Section 4.2.2. We previously discussed the merits of SIFT, USURF-64 and USURF-128 in Section 4.5 and found little to separate them in terms of performance in the experiments described in that section. Hence, it was felt an additional evaluation of the three descriptors would not provide any additional insights in this section, and for that reason we have only run these experiments using the SIFT descriptor. We briefly summarise the algorithm used in that section below:

- Compute SIFT features on the input image
- Match these features to the SIFT feature database
- Each keypoint specifies 4 parameters: 2D location, scale, and orientation
- To increase recognition robustness: use the Hough transform to identify clusters of matches that vote for the same object pose
- Each keypoint votes for the set of object poses that are consistent with the keypoint's location, scale, and orientation
- Locations in the Hough accumulator that accumulate at least three votes are selected as candidate object/pose matches

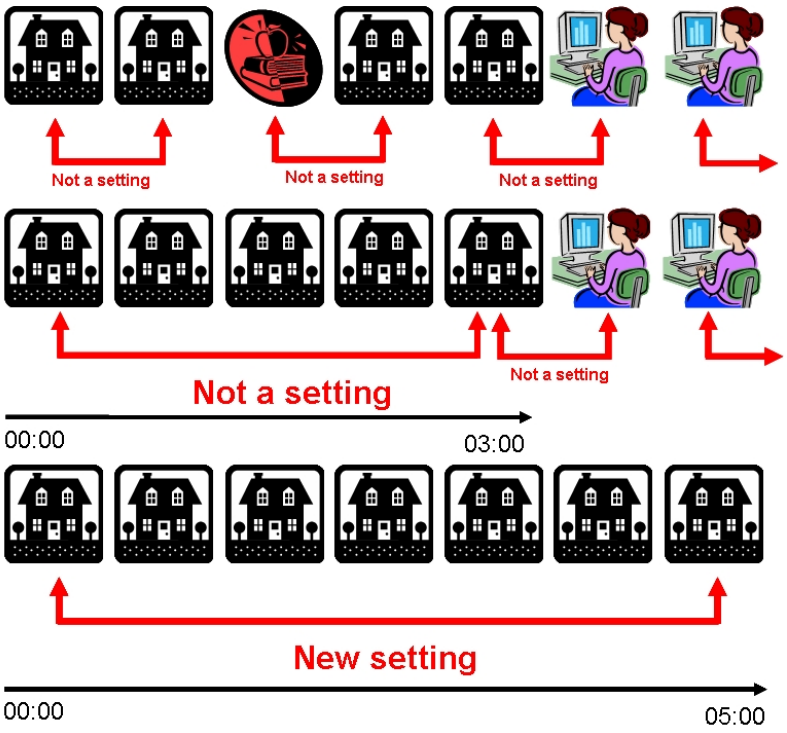


Figure 6.22: The process of automatically flagging potential new settings by the system. The images must be visually similar, temporally aligned, and occur for a time period of x minutes or greater. In this example, x is equal to 5 minutes.

- A verification step matches the training image for the hypothesised object/pose to the image using a least-squares fit to the hypothesised location, scale, and orientation of the object

In order to incorporate the information described above concerning the characteristics of particular settings into the algorithm, we make some minor modifications. Firstly, instead of searching the entire feature database for a match (note that the database in this case refers to the new collection of images being loaded), we only attempt to match an image to the image located next to it in time. This imposes a temporal constraint on the matching process. If the proceeding image contains a match, we then proceed to attempt to match the following image to determine if that image is also a match to the first image. This process continues until a group of matching images are detected which span a window of x minutes or greater. Once this scenario arises, we flag the entire group of images as a potential setting.

This process is graphically illustrated in Figure 6.22. In this example, the dynamically chosen time threshold, x, is equal to 5 minutes in length. In the first scenario in this image, the images loaded by the user change frequently, hence no setting is found. In the second scenario, a group of images which match have been detected by the system. However, they only cover a time period of

User	Classification Error
User 1	0.0839
User 2	0.1789
User 3	0.0843
User 4	0.1391
User 5	0.0494
Average	0.1071

Table 6.12: The classification error for each user, as well as the overall average, for the algorithm to detect new settings.

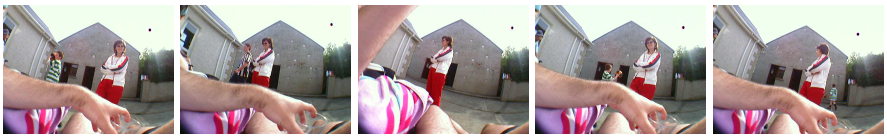
3 minutes (less than x). Therefore, no setting is detected. Finally, in the final scenario, a group of images has been located whose SIFT features match using the criteria outlined above. The group also encompasses a time period of 5 minutes, or greater, and hence they are flagged to the user as a potential setting.

6.3.1 Experimental Results

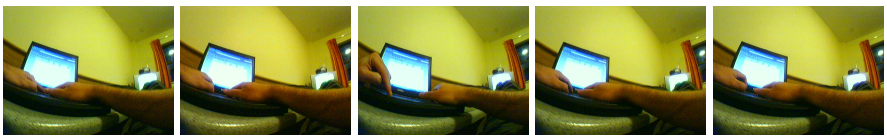
Due to the restricted dataset available for analysis, these experiments were carried out on the same images previously analysed in Section 4.4. This collection consists of a total of 207,580 SenseCam images gathered by five different users over varied periods of time. The number of images gathered by each user is shown in Table 4.2 while the number of images annotated as being in a setting is shown in Table 4.3. Using these images provides a groundtruth with which to evaluate the results of the algorithm described above (in Section 6.3).

The algorithm is designed to propose new settings to the user as the Visual Diary evolves and new images are added. By running the algorithm over the existing annotated collection, we can determine if it detects those settings already annotated by the user. This provides an initial evaluation of the algorithm. In addition, we can also determine if the algorithm detects new settings, not previously annotated by the user. These images will be qualitatively analysed to ensure they meet the criteria for a setting outlined in the previous section. The experiment was carried out using the SIFT features on each user's collection, giving a total of five different experiments. In order to evaluate the performance of this approach, we use the performance measures described in Section 4.3.1, namely Precision / Recall and the overall Classification Error.

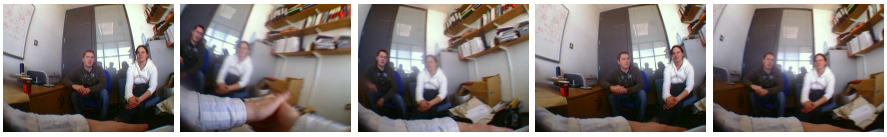
The classification error for each user, and the average across all users, can be seen in Table 6.12. Overall, we can see that the algorithm performed very well. It detected most settings previously annotated by the users, and most of those that were not detected were below the dynamically chosen threshold. However, not all settings annotated by the users were detected. In particular,



(a) User 1 - Setting 1



(b) User 1 - Setting 2



(c) User 2 - Setting 1

(d) User 2 - Setting 2

(e) User 3 - Setting 1

(f) User 3 - Setting 2

(g) User 4 - Setting 1

(h) User 4 - Setting 2

(i) User 5 - Setting 1

(j) User 5 - Setting 2

Figure 6.23: Sample images detected as potential new settings for each of the 5 users

Setting	Precision	Recall
Class 1	93.72%	86.83%
Class 2	78.81%	76.22%
Class 3	94.63%	80.18%
Class 4	98.01%	85.54%
Class 5	97.24	98.6%
Class 6	96.69%	96.28%
Class 7	98.27%	87.69%
Class 8	97.21%	96.91%
Class 9	93.75%	95.74%
Class 10	98.73%	98.73%
Class 11	98.61%	98.96%
Class 12	100%	100%
Class 13	64.93%	94.34%
Class 14	99.15%	89%
Class 15	82.05%	74.42%
Class 16	94.73%	82.89%
Class 17	98.7%	96.45%
Class 18	65.74%	97.26%
Class 19	74.84%	78.81%
Class 20	75.72%	84.78%
Class 21	89.47%	72.34%
Class 22	60.27%	97.77%
Class 23	60.21%	96.66%
Class 24	40.95%	97.72%

Table 6.13: Precision and Recall figures for each setting for User 1

User 2 had annotated a number of short settings, as well as a number of settings which the algorithm would not have considered a setting. These were sequences of images with a break (i.e. a sudden change of scene), before the user returned to the original activity. The user deemed this to be a setting, but the algorithm does not. The algorithm performed best on *User 5*'s images as this user had a few well defined settings which were longer than the threshold time period. Overall though, the performance is encouraging, and the average classification error of 0.1071, as well as the precision / recall figures shown in Tables 6.13, 6.14, 6.15, 6.16, and 6.17, confirm this. As with the algorithms described in Chapter 4, it's worth noting that the quality of the annotations do have an impact on overall performance. However, we are satisfied that the algorithm can detect the majority of the interesting settings available in a collection of lifelog images.

A sample of the images proposed as new settings for each of the users can be seen in Figure 6.23. These images match those previously annotated by the users. Perhaps more interestingly, a number of new settings were detected which were not annotated by the users. The most likely reasons for their omission is that they were either not interesting for the user, or more likely, that

Setting	Precision	Recall
Class 1	78.77%	72.87%
Class 2	77.19%	78.57%
Class 3	85.4%	87.72%
Class 4	72.41%	77.77%
Class 5	35.82	39.34%
Class 6	93.75%	95.23%
Class 7	68.06%	61.36%
Class 8	71.26%	75.9%
Class 9	77.11%	88.9%
Class 10	88%	68.57%
Class 11	90.56%	92.31%
Class 12	89.13%	89.13%
Class 13	88.17%	92.13%
Class 14	78.21%	100%
Class 15	67.12%	34.26%
Class 16	83.15%	92.43%
Class 17	98.48%	83.33%
Class 18	57.64%	61.32%
Class 19	75.96%	77.77%
Class 20	78.32%	76.71%
Class 21	100%	84.75%
Class 22	80.08%	88.73%
Class 23	89.23%	95.53%
Class 24	100%	90.47%
Class 25	72.72%	72%
Class 26	75.51%	69.17%
Class 27	93.54%	97.75%
Class 28	87.87%	90.63%
Class 29	90.82%	92.96%
Class 30	92.16%	94.94%
Class 31	100%	100%
Class 32	62.5%	42.37%
Class 33	53.98%	42.07%
Class 34	53.45%	64.58%
Class 35	90%	86.53%
Class 36	84.61%	73.33%
Class 37	91.36%	87.58%
Class 38	87%	89.92%
Class 39	84.39%	77.24%
Class 40	94.03%	81.81%
Class 41	92.13%	90.11%
Class 42	96.61%	79.16%

Table 6.14: Precision and Recall figures for each setting for User 2

Setting	Precision	Recall
Class 1	89.26%	94.23%
Class 2	84.65%	89.35%
Class 3	94.31%	94.31%
Class 4	78.1%	51.19%
Class 5	91.66	39.28%
Class 6	48.14%	47.27%
Class 7	90.56%	88.88%
Class 8	86.66%	93.6%
Class 9	91.01%	95.01%
Class 10	88%	90%
Class 11	94.86%	98.04%
Class 12	97.61%	97.61%
Class 13	95.36%	96%
Class 14	84.93%	86.71%
Class 15	89.55%	91.39%
Class 16	95.23%	93.11%
Class 17	93.84%	92.81%
Class 18	91.93%	89.06%
Class 19	100%	95.15%
Class 20	100%	97.93%

Table 6.15: Precision and Recall figures for each setting for User 3

Setting	Precision	Recall
Class 1	93.37%	39.27%
Class 2	100%	37.14%
Class 3	88.93%	92.78%
Class 4	72.05%	65.33%
Class 5	67.92	50.34%
Class 6	66.66%	58.82%
Class 7	97.46%	100%
Class 8	89.76%	97.96%
Class 9	60.26%	99.46%
Class 10	99.25%	100%
Class 11	100%	100%
Class 12	100%	99.53%
Class 13	75.33%	70.87%
Class 14	93.83%	91.94%
Class 15	69.51%	85%
Class 16	66.1%	80.13%

Table 6.16: Precision and Recall figures for each setting for User 4

Setting	Precision	Recall
Class 1	78.54%	98.91%
Class 2	88.48%	89.13%
Class 3	89.94%	98.58%
Class 4	92.85%	93.69%
Class 5	100	90.15%
Class 6	100%	100%
Class 7	100%	90.14%
Class 8	100%	81.33%
Class 9	96.28%	100%
Class 10	97.85%	96.47%

Table 6.17: Precision and Recall figures for each setting for User 5

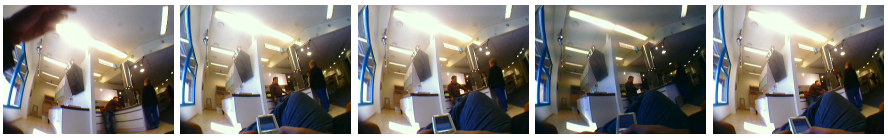
they were simply missed during the annotation phase. Although the annotation tool used in this thesis was designed to reduce the effort required by the user, the annotation process still requires time out of people's busy schedules, so certain settings may easily have been missed. Sample images from some of the proposed new settings can be seen in Figure 6.24. The detection of these new settings highlights the effectiveness of this approach and confirms its suitability as a method to reduce the annotation burden on the user and to allow the Visual Diary to continue to grow over time.

6.3.2 Validation & Importance of Proposed Settings

The final requirement to allow the Visual Diary to evolve over time is to allow the newly proposed settings to be integrated into the existing diary structure. This will require some level of validation from the user to indicate whether or not the proposed settings are correct and are of interest to them. In addition, we would also like to determine the importance of the settings once the user has confirmed that they are indeed correct.

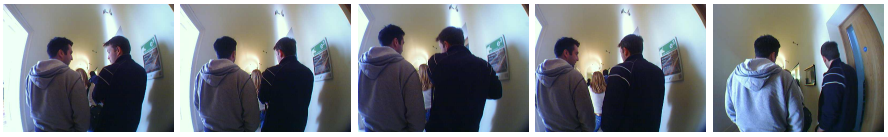
The simplest way to achieve this is via the user interface. Regarding the importance of particular settings, during the user evaluation experiments, described in Section 5.4 and Section 5.5, the users indicated that they didn't find the importance information all that useful. Many of the users "hardly noticed the different sizes" and "didn't think they helped at all". However, what did emerge was a requirement to mark particular settings of interest as "favourites". For example, one user commented that they would like to "save my favourite settings" while another would like to highlight their "most interesting settings". This would give the user more control over this process as they would have to ability to adjust the importance at any time by assigning or removing images from their favourites. The previous approach required them to assign importance at the annotation

(a) User 1



(b) User 2

(c) User 3



(d) User 4

(e) User 5

Figure 6.24: Potential settings detected by the algorithm which were not previously annotated by the users

stage, and this was then locked into the system. The approach proposed here is more flexible and more suitable when using the automatic setting detection approach outlined in Section 6.3. This can be acheived very easily and efficiently on the user interface and we discuss these issues further in Section 6.4.

6.3.3 Analysis of a Large Collection of Images

The amount of images used in these experiments is significant, however, given that the SenseCam gathers approximately 3,000 images per day, the entire collection currently analysed only represents a little over 69 days worth of lifelog data. In lifelog terms, this is a relatively small amount of information. Therefore, as a final experiment, we decided to investigate how we could analyse an extremely large collection of unannotated SenseCam images. The collection consists of 1,864,149 images gathered from four different users (the same users as our previous experiments). This represents a little over 621 days worth of lifelog data, or approximately 21 months of data.

The breakdown of images between different users can be seen in Table 6.18 and, as we can clearly see, the collection is dominated by images gathered by *User 2*. As the collection has not been annotated by the individual users, the data in this collection of images had to be annotated by a single individual. Although this compromises the results somewhat, we believe that this analysis is important as it demonstrates the ability of the Visual Diary to scale over a significant period of time (in lifelog terms), whilst also providing interesting insights for possible future work.

User	**Images**	**Keyframes**
User 1	$92,387$	$1,181$
User 2	$1,686,424$	$19,994$
User 3	$40,715$	504
User 5	$44,440$	440

Table 6.18: The number of images for each user in the larger, unannotated, collection, as well as the number of keyframes extracted from each users collection.

The goal of our analysis of these images is to determine an efficient method to analyse the collection in order to propose new settings to the user. Once new settings have been proposed, and confirmed by the user, the bag-of-keypoints approach (described in Section 4.3.3) can be run as normal to match the new settings to other settings located in the Visual Diary. In order to achieve this, we propose a method broadly similar to that outlined in Section 6.3. The key difference here is that we do not analyse the entire 1.8 million images. Instead, we leverage the event structure described in Section 5.2 to extract keyframes from the larger collection of images. This keyframe extraction process is also described in Section 5.2. Instead of analysing the entire collection of 1.8 million images, we only analyse the keyframes extracted from this collection. This process results in the extraction of 22,125 keyframes from the entire collection. Although analysing the keyframes in this fashion may not provide an exhaustive analysis of the collection, it does provide an initial starting point with which we can begin to structure a Visual Diary containing an extemely large quantity of images. In order to determine exactly how effective the proposed technique is, we initially evaluate the method on the smaller image collection used in this thesis. This collection has been annotated by the users and therefore provides a groudtruth for evaluation purposes.

As previously mentioned, the algorithm used is a variation of that described in Section 6.3. In that algorithm, we analysed the entire collection of images, and imposed a time threshold as a minimum length of time we would like a setting to last. However, when analysing keyframes extracted from the collection, this threshold is no longer relevant. Instead, we simply want to match two keyframes images which are temporally aligned. If the two keyframes are a strong match, it

would suggest that the collections of images they represent, also match. For example, Figure 6.25 shows two sequences of keyframe images extracted from the 1.8 million strong collection. In the first row, we would like the algorithm the identify the images where the user is working on his computer and chatting to a work colleague as potential settings. Similarly, in the second row of images, we would like the algorithm to identify the images where the user is driving and working on his laptop as potential new settings to be flagged for the users attention.

Figure 6.25: Potential settings detected by the algorithm from keyframes extracted from a large collection of 1.8 million images.

The first step is to evaluate the effectiveness of this approach on the annotated collection of images (i.e. the collection of 207,580 images gathered by five users). The keyframes for each user's collection have already been extracted in order to build the user interface (described in Section 5.2). We now process these keyframes, using the algorithm described above, in order to detect any new settings. In order to evaluate the method, we use the Classification Error, described in Section 4.3.1.

The results of the analysis can be seen in Table 6.19. As we can see, a large amount of settings have not been detected through a simple analysis of the keyframes. Given that a number of settings cover a relatively short timespan, significantly less than the average length of approximately one hour, this is not a surprising result. Naturally, a closer examination of the settings which have been detected reveals that they tend to be those events which occur frequently and last a reasonably significant length of time. Other settings detected are those which occur less frequently, but again, last a significant period of time. This in itself is a useful starting point for structuring this large collection. Using this method, we can detect approximately 50% of the settings annotated by the users. These can then be further analysed and placed into the Visual Diary. However, this technique omits approximately 50% of the settings originally annotated by the users in the smaller image collection. In order to analyse these images, we can utilise the setting detection techniques described in Chapter 4. By removing the events detected using the process described here, we can

User	Annotated Settings	Detected Settings	Percentage
User 1	24	13	0.4583
User 2	42	25	0.4048
User 3	20	10	0.50
User 4	16	6	0.625
User 5	10	4	0.60
Average	22.4	11.6	0.4821

Table 6.19: The number of settings annotated by each user and the number of settings detected by analysing the keyframes from each users collection.

then perform an analysis on the reduced set of images to detect the remaining settings annotated in the collection.

Focusing back on the larger collection of 1.8 million images, we can now hypothesise that by analysing the 22,125 keyframes from this collection, we can detect approximately 50% of the settings contained within the collection. This is extremely beneficial for a number of reasons. Firstly, it provides an extremely fast method to flag potential settings to the user from a large image collection, with a speed-up over an exhaustive analysis of the entire collection of several orders of magnitude. The obvious benefit of this is that the user doesn't have to wait for a long time while the system processes the images before they can begin to use the Visual Diary. Secondly, we can then remove these images from our further more detailed analysis of the remaining images, in order to detect the remaining settings. Naturally, the processing of this reduced set of images also provides an improvement in the time taken to analyse, and propose, new settings from a large collection of lifelog images.

In order to evaluate the results, the keyframes were annotated in order to determine where the potential settings were located. This provided a groundtruth for the evaluation and the Classification Error was used to analyse the results. The annotation process simply involved counting the number of settings, based on a visual inspection of keyframes, in each users collection. A setting in this context consists of two matching keyframes, temporally aligned. In total, 945 unique settings were annotated from the entire collection. Naturally, many settings repeat themeselves, so we exclude these from the overall total.

The results of this analysis can be seen in Table 6.20. In total, the algorithm detected 819 unique settings in the collection, meaning that 86.67% of the annotated settings in the entire collection of 1.8 million images were detected by analysing the 22,125 keyframes. Although these results appear to indicate that the algorithm performed better on the larger collection of images than the smaller one, this is not the case. The annotation process used in each example was differ-

ent, hence, the difference in results. The results in Table 6.19 were based on the users annotating the entire set of images in the original collection. The results in Table 6.20 are based on an annotation of the keyframes from the collection of 1.8 million images. Hence, we can hypothesise that the results presented in Table 6.20 represent approximately 50% of the settings in the entire collection (based on our findings in the analysis of the smaller collection of 207,580 images).

User	Annotated Settings	Detected Settings	Percentage
User 1	39	32	0.1795
User 2	855	745	0.1286
User 3	28	23	0.1786
User 5	23	19	0.1739
Total	945	819	0.1333

Table 6.20: The number of settings detected by analysing the keyframes extracted from the larger collection of 1.8 million images

6.4 Visual Diary Toolkit

In Section 2.6, we outlined a user application scenario involving a Visual Diary application. The core of this application involved the detection of settings and the browsing of the images in the collection. These elements were discussed in detail in Chapter 4 and Chapter 5. In addition, approaches designed to automate the detection of new settings were described in Section 6.3. Besides these core facilities of the Visual Diary, a number of other requirements were outlined in the user scenario. These included the ability to highlight settings as a favourite and the personalisation of the Visual Diary.

In Section 6.2.1, we analysed the detected settings and described how we would generate personalised summaries of a user's day. In order to allow the user to generate these summaries, we propose an addition to the user interface, as shown in Figure 6.26. By clicking on the 'Generate Summary' link in the browsing interface, a new window appears. This allows the user to select the time period they would like summarised using the calendar tool. A summary of activities then appears, consisting of the keyframe from the displayed setting and the total percentage time the setting occurred for during that day. We can see this summary in Figure 6.27.

Besides the simple generation of a daily summary, the user can also provide feedback on the summary information. This allows the system to determine whether the settings which occurred during that particular day represent a particular pattern of activity. One can imagine (as we've demonstrated previously in Section 6.2.1) that certain combinations of settings represent routine

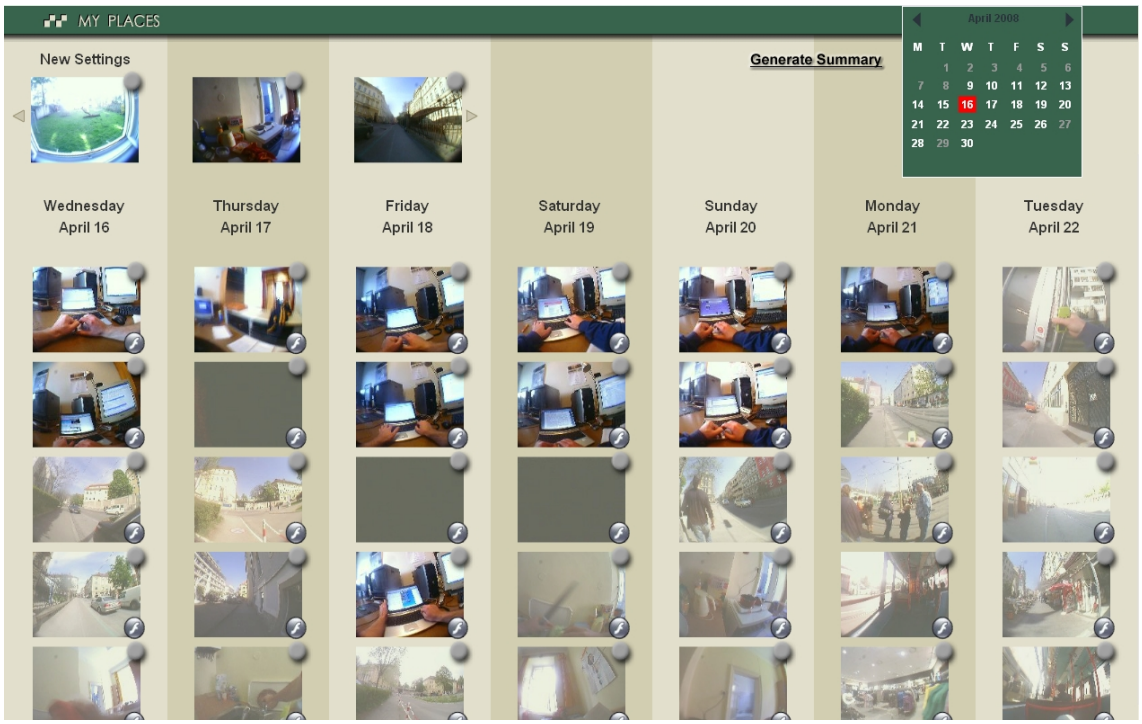


Figure 6.26: This image shows the image browser with a link available to generate a personalised summary for a particular user. By clicking on the link, a new window opens which displays the summary information.

daily activities. Others may represent more unusual activities. It is important to be able to distinguish between these patterns of activity and to allow them to develop dynamically as the Visual Diary grows. Initially, our analysis will provide a distinction between routine activites and other more interesting activities (which the user may subsequently wish to highlight as a favourite). By allowing the user to provide feedback on this process, we can adapt what is considered a routine setting over time, as new settings are introduced. This is useful, as what is routine one week, may not be considered so over an extended period of time. By allowing the user to interact with the system, and to control this process, we can faciliate the dynamic adaptation of the Visual Diary over time based on user feedback. By utilising the information already garnered via the setting detection process (i.e we know what setting an image is in and what time it occurred), we can combine this information with user feedback to generate a particular profile (in this case routine events) for each user. We can then easily locate deviations from this profile.

In Figure 6.27, we can see a daily summary of a user's activities. Based on the analysis described in Section 6.2.1, the system has already classified these images as either part of a routine setting or not. This is highlighted to the user by the use of a coloured border around the images. The orange border indicates that the images displayed are part of a routine setting. However, a red

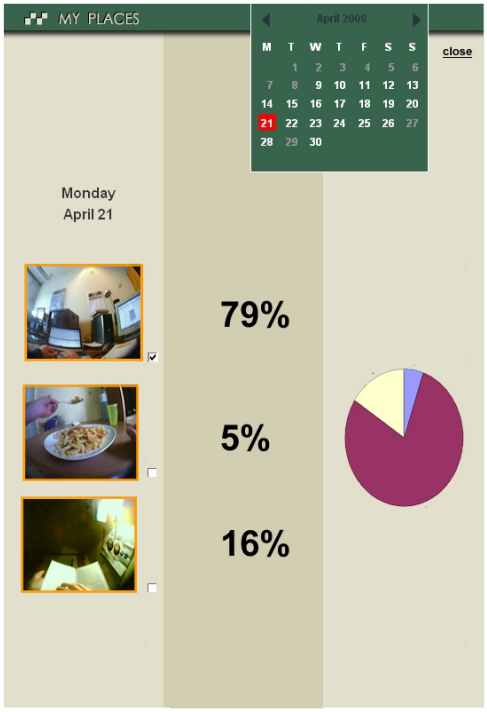


Figure 6.27: This image shows a daily summary of the settings experienced by the user. In this example, all images have been classified as routine events. However, by selecting the checkbox beside a particular image, the user can modify this classification, thus refining the automatically detected results.

border indicates that the image has not been classified as routine and this may represent something more interesting (see Figure 6.28). If the user is unhappy with the automatic classification provided, they can provide feedback here to update this information by simply selecting the checkbox. This sends information to the database indicating that this particular setting is, or is not, part of a routine series of events. This has the effect of updating the models described in Section 6.2.1, thus refining the analysis when images are loaded to the Visual Diary in the future. It also updates the existing information regarding routine settings which currently exists in the system, thereby allowing the defintion of routine settings to dynamically change over time. This simple method allows the user to generate a personalised summary of their activity and provides a powerful tool for them to gain an insight into how they conduct their daily lives. In order to continue browsing through their collection, they simply click on 'close' in the top right corner of the window to close the summarisation window.

In Section 6.3.2, we described how we can detect new settings and determine their importance. This is most effectively achieved in the user interface. In the image shown in Figure 6.29, the importance information from the original browsing interface (see Section 5.2) has been removed. Therefore, all images shown are the same size. In order to mark an image as a favourite, the user

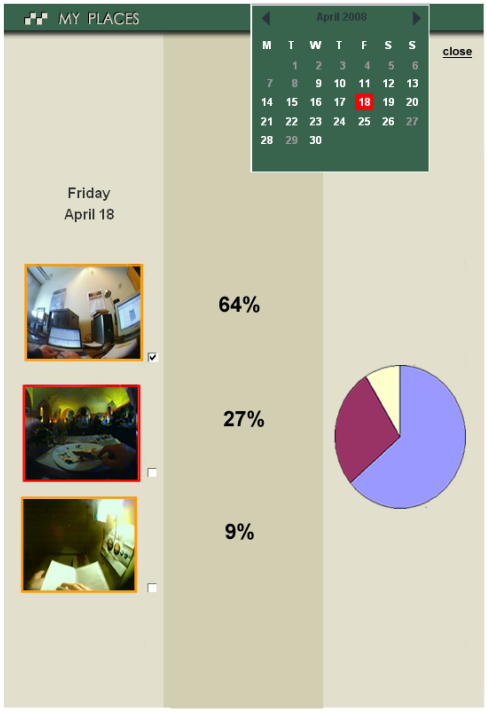


Figure 6.28: This image shows a daily summary of the settings experienced by the user. In this example, one image has been classified as a non-routine event, thus highlighting a deviation from the users normal pattern of activity. As before, by selecting the checkbox beside a particular image, the user can modify this classification, thus refining the automatically detected results.

simply needs to click the grey *f* icon in the bottom right corner of the image. Once clicked, the icon is replaced with a red *f* icon in the bottom corner of the image and the image is increased in size (as shown in Figure 6.30). The effect of the marking is to increase the size of the image in question, making it stand out more on the interface. If the image is not part of a setting, only that image is marked as a favourite. If the image being marked is part of a setting, all the images in that setting will be marked as a favourite, and their size increased accordingly. This simple approach allows the user to very quickly control what images they are most interested in, and they can easily be changed over time as new images are loaded to the system.

In order to validate potential new settings, a similar approach can be taken. As images are loaded into the Visual Diary structure, potential new settings are highlighted in a new section of the interface in the top left corner of the screen. This allows users to scroll through the images detected by the algorithm as potential settings, as shown in Figure 6.31. In order to confirm the system's suggestion, the user simply clicks the grey button in the top right corner of the image. The button will turn orange, indicating that that particular image is indeed a setting. This is shown in Figure 6.32, where two of the suggested images have been confirmed as a setting. Once a

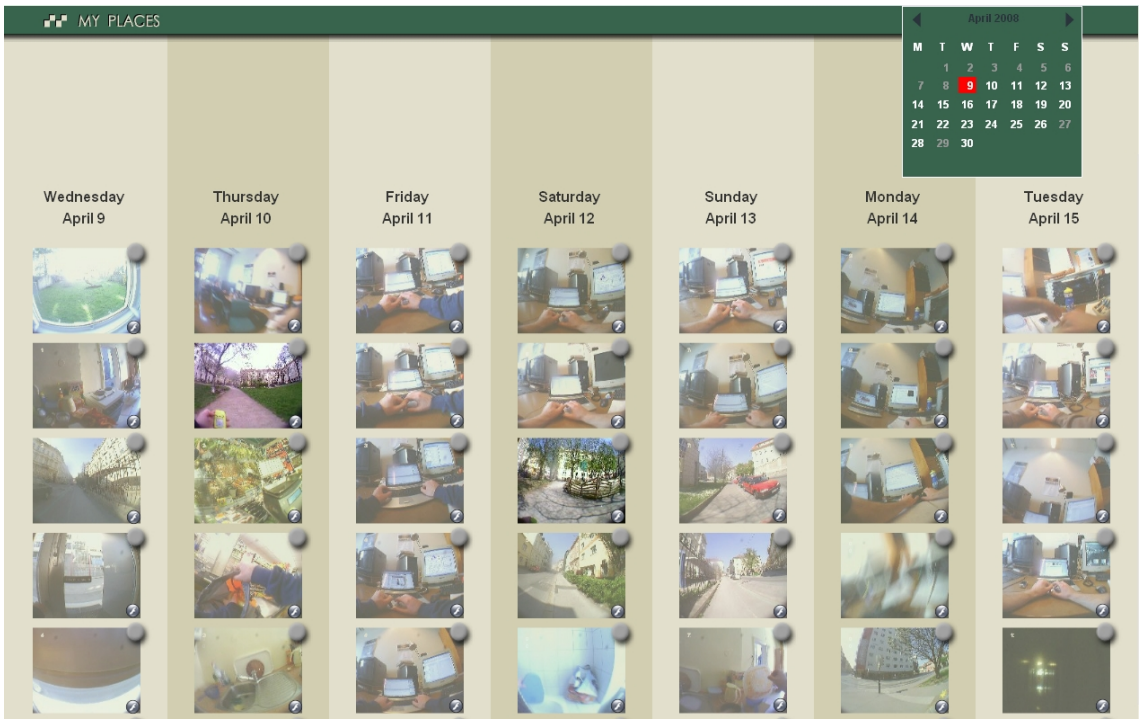


Figure 6.29: In the image shown, all images are the same size. The grey icon in the bottom right corner of the images allows the user to mark the image, or setting, as a favourite.

setting has been confirmed, the approach outlined in Section 4.3.3 can then be used to detect matching settings from the other images in the Visual Diary. If the setting is rejected, nothing more happens with that group of images (bar their display in the diary). As with the approach to selecting favourites, this method places a minimum amount of burden on the user, and effectively uses the algorithms developed to structure the Visual Diary.

6.5 Discussion

It is clear from the discussion above that the detection of settings facilitates detailed further analysis of the Visual Diary of an individual user. The experiments previously carried out in this thesis were designed to facilitate the detection of settings across a static collection of user images. In that regard, the experiments were successful, as the results presented in Section 4.5 demonstrate. However, a Visual Diary does not consist of a static collection of images, and the work described in this chapter is designed to overcome some of the issues involved in managing the growth of a Visual Diary over an extended period of time. In particular, a range of software engineering solutions to manage the growth of the Visual Diary were discussed in Section 6.3. The approaches described facilitate the automatic detection of new settings as they appear, as well as the identifi-

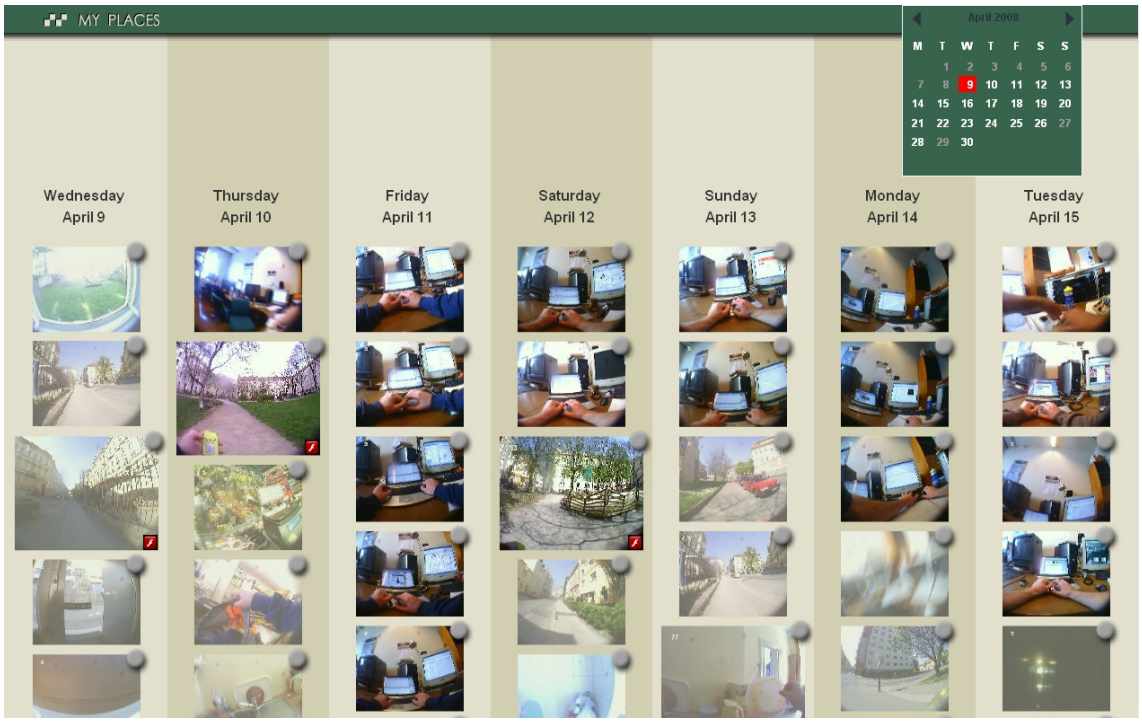


Figure 6.30: In the image shown, a number of images have been selected as favourites by the user. These are indicated by their increased size and a small icon in the bottom right corner of the image.

Figure 6.31: In the image shown, potential new settings are displayed in the top left corner of the screen. The user can scroll through the new settings by clicking on the left and right arrow icons.

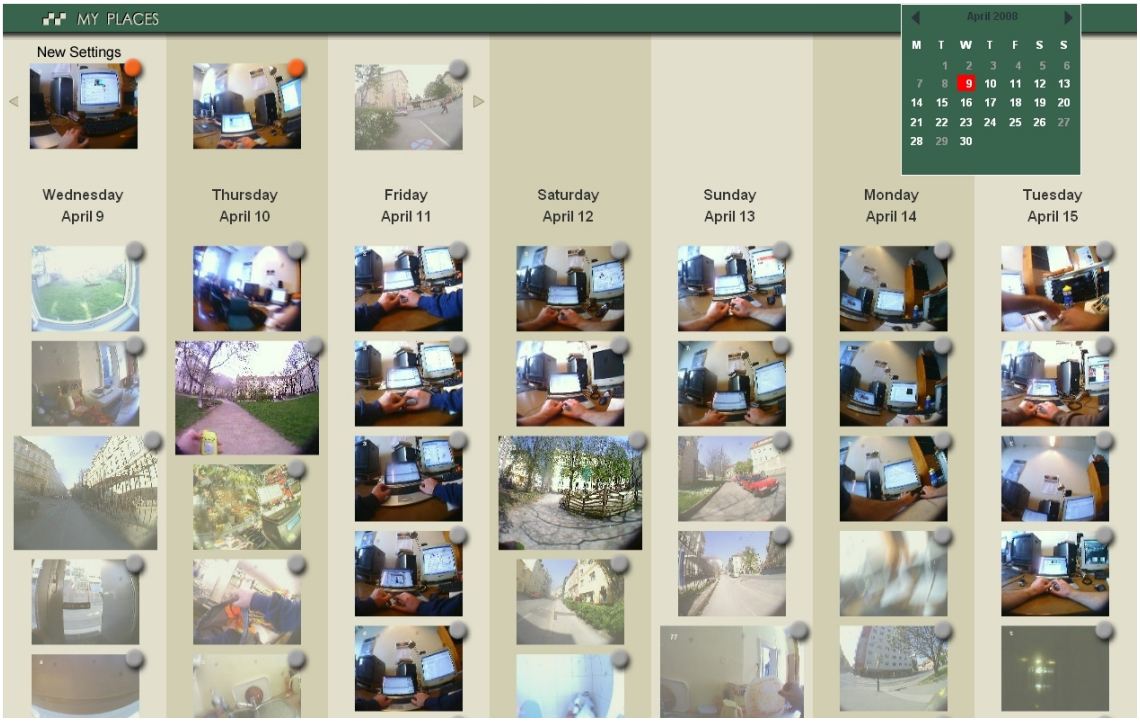


Figure 6.32: In the image shown, two new settings have been confirmed by the user. The user confirms a new setting by clicking on the grey icon in the top right of the image. For confirmed settings, the grey icon turns orange, as is the case with two of the new settings shown here.

cation of important settings. In addition, an approach to managing a significantly larger volume of images is presented. However, it is acknowledged that these are only one of a range of possible solutions to these particular problems.

Of more interest, from a research perspective, is the analysis of the detected settings described in Section 6.2. In this section a set of techniques is presented to exploit the temporal patterns of settings in order to try and gain a further understanding of the activities the user was engaged in when the images were captured. In particular, it was noted that some users followed relatively routine patterns of activity, whilst other users activities were more random. Therefore, by performing a relatively straightforward temporal analysis of the detected settings, we can compare the activity across users to determine whether they were engaged in routine activities, or something outside a routine (depending on the definition of routine). The analysis of these patterns facilitated the development of an approach to detect whether a new setting loaded to the diary was routine or not (for an individual user), thus providing additonal information to the user in the Visual Diary. The personalised summaries generated from this analysis could be a powerful tool in the management of a Visual Diary and, to the best of our knowledge, no other authors have generated similar summaries from user data.

6.6 Conclusion

In this chapter, we discussed the management of a changing Visual Diary over time. The previous work in this thesis described how the diary would initially be setup, how a user would annotate settings from their image collections, and how we would detect those annotated settings and structure the Visual Diary. What happens after that is the focus of this chapter.

The first issue addressed was an analysis of the settings themselves. We examined the characteristics of the annotated settings in order to determine what information this might provide concerning the individuals who had collected this data. We found that certain users followed very routine patterns during the daily activities, whilst some of our other users lives were more varied during this time. This had an impact on the types of settings gathered, with those following the normal day to day routine having a number of settings which occurred at around the same times throughout their collections, whilst also having very few settings which would stand out from the routine. The other users experienced many infrequent settings due to their constantly changing locations and places of work. We also used this information to demonstrate how we can create personalised summaries of the users activities over a particular time period and showed how this could be an extremely powerful feature in the Visual Diary.

The second issue addressed concerned the ability of the Visual Diary to adapt as new images were gathered by the user. This section also addressed some concerns raised in the user evaluation study conducted in Chapter 5. In particular, issues related to how the importance of images was utilised in the diary were addressed. We found that users didn't like the fact that they couldn't control this factor, but instead, they preferred the ability to mark images or settings as favourites themselves at any time. In terms of allowing the Visual Diary to cope with new images, we proposed a method to analyse new images before they are loaded into the Visual Diary. This technique proposes new settings for the user and the user can then validate the proposed settings in the user interface. Once the user has confirmed the new image to be part of a setting, the algorithms described in Chapter 4 can then detect matching settings from the other images in the diary.

The third issue concerned the ability of the Visual Diary to scale to a very large volume of images gathered over a significant period of time. We examined the feasibilty of analysing the keyframes extracted from such a large collection and found that we could detect approximately 50% of the available settings by analysing a fraction of the collection. This has important conse-

quences in terms of the overall processing times of the system and seems like a fruitful area for future work.

Finally, we outlined how the tools described in this chapter could be integrated into the existing browsing tool. We presented the user interfaces and described how each tool would operate in practice. Each of the tools presented fulfills the criteria outlined in the user application scenario, thus completing the requirements laid out in this thesis for a Visual Diary application. However, each tool also provides scope for future work in this area, particularly in the area of personalisation.

CHAPTER 7

Conclusions

In this thesis, a system designed to manage and organise a lifetime's worth of visual data is proposed. The Visual Diary application is designed to give added value to a user capturing vast numbers of images using a passive image capture device (such as Microsoft's SenseCam). During the early stages of this work (see Section 1.1), it was stated that the key challenge is to be able to manage, organise, and search large volumes of photos in order to present them to a user in a visually coherent manner. The use of passive capture devices (discussed in Section 1.2) means that a key element of meeting this objective is the ability to identify representative sample images in the first place, as they will typically need to be selected from extremely large image collections. In addition, it was stated that the system should be able to identify images which the user deems to be important for some reason.

In order to tackle this problem, we identified setting detection as a key enabling technology (see Chapter 3). By identifying images captured in the same real world locations, and then subsequently matching images from these locations across different days, we can begin to structure the Visual Diary in a way which provides real value to an end user. In order to perform setting detection, we outlined a number of challenges which needed to be overcome. Due to the fact that images were captured over extended periods of time, images captured at the same location would more than likely not be captured under the same conditions (i.e. there could be significant visual distortions). These distortions include changes in image scale and rotation, changes in illumination, noise, and minor changes in viewpoint. In addition, specific objects in images captured at the same location could be partially occluded. In order to develop an application which could perform reliable matching of images with these characteristics, a wide variety of approaches were discussed in Chapter 3. From this discussion, interest point detection algorithms were deemed to

be the most appropriate to deal with these specific problems (discussed in Chapter 4).

An approach to identifying specific settings was also presented in Chapter 4 and after a rigorous evaluation, it has been shown that the system meets the initial requirements. The best performing version of the system obtains a classification error of 0.0898, whilst also obtaining high values for precision and recall from numerous challenging settings from five different users. This indicates accurate performance of the proposed approach under a variety of scenarios.

7.1 System Assumptions, Limitations and Potential Issues

Perhaps the biggest assumption made in this thesis is that passive image capture is something users will accept and that these devices will become commonplace in the future. We have outlined a number of potential applications for this technology in Section 2.5, and have discussed in detail the significant challenges this technology presents in Chapter 2. However, although lifelogging applications are beginning to appear in the market place, the technology described in this thesis still remains in the research phase. Despite these assumptions, the setting detection technology developed in this thesis has been demonstrated to be a valid approach in managing large volumes of lifelog image data, and the techniques applied have relevance in other application areas.

Another potential limitation involves the annotation of a large volume of image data by users. The thoroughness and effectiveness of such evaluations can often be questionable, particularly when the annotation task takes up a significant portion of a user's time. Although the annotation tool developed in this thesis (discussed in Section 4.3.3.1) was designed for efficiency, a large number of images will still require a significant annotation effort. The work described in Section 6.3 begins to address this issue by allowing new settings to be automatically detected by the application. Future work will also focus on reducing the annotation burden further.

Finally, although the algorithm used to perform setting detection has been thoroughly analysed, it is also important to carry out a user evaluation of the application developed. This evaluation was discussed in Chapter 5. The evaluation results were positive overall, however, certain limitations do exist which were highlighted in this chapter. In particular, the limited volume of data provided by a very limited number of users leaves the significance of the evaluation results open to question. This is a difficult issue to overcome and given the constraints imposed, the evaluation provided is comprehensive and addresses the core issues involved. Future work will focus on gathering more data from a wider pool of users in order to further enhance and validate the

application developed.

7.2 Thesis Overview and Research Contributions

In Section 1.3, the objectives for this thesis were presented. At this point, we review those objectives to determine whether they have been met by the subsequent work described. The first objective of this thesis is to outline the current state of the art in lifelogging. This was discussed in detail in Chapter 2. Having presented a broad overview of lifelogging, we also discussed the area of content management in this chapter. The purpose of this discussion was to provide some perspective for the problems raised when managing lifelog image data.

The second objective is to investigate a new approach to help solve the content management problem as it pertains to lifelogging and a Visual Diary application. Setting detection was identified as an important technology in this regard and a number of approaches were presented (see Chapter 3). One of the key aims of this chapter was to determine the most appropriate techniques necessary to perform setting detection in a Visual Diary and this represents the first contribution of this thesis. Specifically, we presented a comprehensive evaluation of the most appropriate techniques available to perform setting detection.

The third objective is to present a number of approaches to setting detection in detail and to explore the robustness of these algorithms. In each case, each of the main parameters incorporated within the proposed techniques are rigorously examined. This work was discussed in Chapter 4, and represents another research contribution. Specifically, we: (a) determined the optimal parameters to use in the detection of settings; (b) provided a comparison between the major components of the system (i.e. SIFT or SURF, K-means or X-means, etc.). A user study was conducted to validate the proposed techniques and an application was developed to facilitate this evaluation (see Chapter 5). This represents another research contribution of this thesis. Specifically, we: (a) developed a novel web-based interface to facilitate the management and organisation of a Visual Diary; (b) discussed the most appropriate principles in application design used to develop a Visual Diary application, and hence provided a contribution to application design principles in this area; (c) demonstrated a technique for evaluating the accuracy and overall utility of a settings based Visual Diary application.

The fourth objective is to analyse the results obtained in order to facilitate the development of techniques which will allow the Visual Diary to dynamically evolve as it grows over time. We

discussed a number of techniques to facilitate this in Chapter 6. In particular, we focused on the ability of the Visual Diary to grow automatically (hence reducing annotation burden) and on allowing users to analyse their settings in more detail. This work constitutes the final research contribution. Specifically, we: (a) provided an insight into the way different users conduct their daily lives by analysing the settings detected in order to gain further insights; (b) developed methods to allow users to generate personalised summaries and to enable them to detect unusual settings over a period of time.

The final objective of this thesis is to indicate directions for further research. In particular, we consider possibilities for further improvement of the proposed algorithms and discuss ideas for additional improvements of the Visual Diary application itself. We discuss this in more detail in Section 7.3. Finally, a number of papers have been published in relation to this work, demonstrating that the research contributions discussed are relevant to the wider academic community.

7.3 Future Research

During the development and evaluation of the Visual Diary application, numerous avenues of potential investigation for future research were uncovered. Some of these directions may lead to further improvement of the proposed techniques, whilst others may be seen as the application of the proposed system as an integral element in a variety of application areas. In this section, a number of these future directions are described in further detail.

7.3.1 Algorithmic Improvements

The best performing algorithm was developed using a bag-of-keypoints approach with an SVM classifier. However, with a bag-of-keypoints approach, we are faced with a number of implementation choices. These include how to sample image patches, what visual patch descriptor to use, and how to classify images based on the resulting global image descriptor. In this work, we used the SIFT, U-SURF64, and U-SURF128 features to sample and describe the image patches. These features have been used in many applications for object detection and recognition [132] [212]. However, they have not been widely used as a tool to detect settings across the entire image. The very nature of SenseCam images themselves means that they are inherently of poor quality, with many blurry shots, significant changes in lighting, etc. Therefore, it was important that the training images used in these experiments provided a realistic data set with which to describe the

settings in question. This is in stark contrast to most object detection systems using these features, where the use of high quality training, or model, images is crucial [177]. We believe the use of these features is justified in our work due to the excellent results achieved and the large body of existing work in similar areas. Furthermore, the results presented in Section 4.6, indicated that a combination of features may generate further improvements in results, and this area should be further investigated. However, despite this, it would be naive to ignore other algorithms, such as GIST [234], which provides a global description of an image, and which are also tackling similar problems (i.e. managing and understanding large volumes of images [180]). In addition, it has been suggested that randomly sampled image patches are more discriminant than keypoint based ones and this should be further investigated in our work.

Another issue which can impact performance is the size of the descriptor generated using interest point detectors. This represents a significant bottleneck in the system in terms of overall speed and performance. At present, the techniques presented in this thesis could not be performed in real time. However, using techniques to reduce the dimensions of the descriptors could prove useful in this regard. Our initial research into this area produced inconclusive results. However, other authors have obtained interesting results using PCA on SIFT descriptors [217]. In their work, they used the first 50 components. However, no information is provided as to how this number was determined. Other authors have performed PCA on the 41×41 pixel patches that are passed through the SIFT interest point detector, instead of on the descriptor itself [152]. Again, the results achieved here using very low-dimensional descriptors (e.g. 20) were good. Further investigations are necessary to determine if PCA, or other similar techniques, can be successfully used with SenseCam images.

The algorithms used in this thesis utilised content-based information only as this was the only source of information which could be relied upon as a collection of lifelog data is gathered. However, as the technology used to capture additional sources of metadata improves, this assumption should be re-examined. In particular, location based metadata could yield significant improvements in the organisation of images in a Visual Diary application. In addition, biometric technologies are also advancing at pace and the metadata provided by these technologies could provide vital information regarding the users activity on a day to day basis. This can also greatly enhance the information and analysis provided in the Visual Diary application. Investigations into these areas should continue and new techniques will be evaluated as and when they appear.

7.3.2 Application Improvements

The key to the success of the Visual Diary application is it's ability to evolve as more and more images are uploaded. Much future work will focus on ehancements to the current core browsing application and additional tools described in Chapters 5 & 6. Although the evaluation of this application was successful, it would be useful to gather more data from new users in order to continually evaluate the application. As previously discussed, one of the key limitations in this work has been the limited quantity of data gathered by a core group of users. If these limitations could be overcome, a much richer browsing tool could be developed based on the additional experimentation and feedback from other users.

Besides the core browsing tool, a toolkit of additional features was developed to facilitate the evolution of the Visual Diary over time. Perhaps the most exciting element of this is the ability for each user to generate personalised summaries of their data. Further experiments are necessary to determine how we can further enhance this type of feature. In addition, a single user has gathered approximately 1.8 million images, and continues to gather data using the SenseCam. The techniques presented in Section 6.3.3 provided a starting point in the analysis of these images, but further work is necessary to provide a comprehensive analysis of this collection and to upload it to the Visual Diary application. This will allow a much more detailed investigation into the settings occurring over approximately 18 months of user data and constitutes a significant research challenge.

Finally, a number of additional applications using interest point detectors have been developed, and these were discussed in Appendix D. These applications leverage some of the tools developed in this thesis and work in these areas continues. In particular, the development of real-time matching systems which could work on mobile devices would represent a significant step forward in the museum information and tourist information systems. Other similar application areas will also be investigated using variants of the algorithms developed in future work.

APPENDIX A

Precision / Recall for Bag of Keypoints Method

Setting	Precision (SVM)	Recall (SVM)	Precision (MLP)	Recall(MLP)	Precision (KNN)	Recall (KNN)
Class 1	98.05%	99.54%	98.04%	99.54%	98.19%	99.39%
Class 2	100%	100%	100%	98.39%	100%	100%
Class 3	92.44%	89.37%	92.83%	86.04%	92.2%	43.19%
Class 4	100%	96.94%	100%	90.81%	100%	76.53%
Class 5	98.48	89.04%	100%	86.3%	100%	56.16%
Class 6	92.91%	97.65%	91.89%	92.28%	48.46%	100%
Class 7	97.14%	100%	94.44%	100%	100%	94.11%
Class 8	92.13%	97.62%	94.22%	97.02%	98.13%	62.5%
Class 9	93.88%	93.88%	95.74%	91.83%	33.85%	87.75%
Class 10	99.49%	99.49%	98.72%	97.72%	99.73%	95.7%
Class 11	98.66%	99.32%	98.65%	98.65%	97.01%	43.6%
Class 12	96.88%	100%	100%	100%	100%	100%
Class 13	95.83%	82.14%	69.69%	82.14%	100%	7.14%
Class 14	93.65%	90.77%	97.95%	73.85%	90.57%	73.85%
Class 15	50%	39.13%	56.25%	39.13%	0%	0%
Class 16	91.3%	78.75%	87.87%	72.5%	100%	16.25%
Class 17	97.48%	96.99%	96.68%	94.98%	100%	66.66%
Class 18	96.15%	98.68%	89.61%	90.79%	94.44%	89.47%
Class 19	87.95%	96.05%	80.68%	93.42%	93.33%	55.26%
Class 20	93.81%	89.65%	95.29%	79.8%	89.11%	64.53%
Class 21	100%	54.16%	100%	29.16%	100%	4.16%
Class 22	100%	91.66%	27.38%	95.83%	100%	41.66%
Class 23	90.32%	93.33%	75%	100%	100%	3.33%
Class 24	100%	100%	37.5%	95.45%	100%	95.45%

Table A.1: Precision and Recall figures for User 1 using the Bag-of-Keypoints approach for all classifiers with SIFT features

Setting	Precision (SVM)	Recall (SVM)	Precision (MLP)	Recall(MLP)	Precision (KNN)	Recall (KNN)
Class 1	97.76%	100%	98.79%	99.54%	92.29%	98.48%
Class 2	100%	98.39%	100%	93.54%	96.55%	90.32%
Class 3	92.02%	92.02%	94.21%	92.02%	70.93%	60.79%
Class 4	100%	97.96%	100%	92.86%	100%	65.31%
Class 5	100	97.26%	100%	93.15%	100%	49.31%
Class 6	95.32%	98.34%	92.84%	98.48%	95.12%	91.44%
Class 7	82.92%	100%	85%	100%	100%	85.29%
Class 8	94.28%	98.21%	97.51%	93.45%	90.44%	73.21%
Class 9	93.47%	87.75%	97.77%	99.79%	96.42%	55.1%
Class 10	100%	98.23%	100%	95.2%	100%	95.96%
Class 11	100%	100%	99.32%	99.32%	97.93%	95.3%
Class 12	96.87%	100%	96.87%	100%	96.55%	90.32%
Class 13	100%	64.28%	96.29%	92.85%	100%	46.43%
Class 14	98.38%	93.84%	79.49%	95.38%	76.36%	64.62%
Class 15	80%	52.17%	78.57%	47.82%	33.33%	4.34%
Class 16	88.52%	67.5%	75.38%	61.25%	100%	31.25%
Class 17	95.82%	97.74%	98.22%	96.99%	97.31%	81.45%
Class 18	80.89%	94.73%	77.08%	97.36%	79.51%	86.84%
Class 19	93.84%	80.26%	91.07%	67.1%	45.76%	35.53%
Class 20	95.47%	93.59%	96.74%	87.68%	34.17%	100%
Class 21	90%	75%	77.41%	100%	100%	25%
Class 22	95.24%	83.33%	95.65%	91.66%	100%	70.83%
Class 23	93.75%	100%	100%	93.33%	100%	73.33%
Class 24	95.65%	100%	43.14%	100%	100%	95.45%

Table A.2: Precision and Recall figures for User 1 using the Bag-of-Keypoints approach for all classifiers with U-SURF64 features

Setting	Precision (SVM)	Recall (SVM)	Precision (MLP)	Recall(MLP)	Precision (KNN)	Recall (KNN)
Class 1	97.03%	99.69%	99.85%	99.24%	89.83%	96.95%
Class 2	100%	100%	100%	77.42%	100%	95.16%
Class 3	90.22%	95.01%	95.03%	89.03%	65.59%	47.51%
Class 4	100%	96.93%	100%	94.89%	100%	39.8%
Class 5	100	100%	94.66%	97.26%	100%	49.31%
Class 6	96.11%	99.03%	94.92%	95.45%	95.43%	89.38%
Class 7	91.89%	100%	96.29%	76.47%	100%	73.53%
Class 8	98.76%	94.64%	96.34%	94.04%	93.07%	72.02%
Class 9	87.5%	85.71%	90%	91.83%	100%	55.1%
Class 10	99.47%	95.71%	97.98%	98.23%	100%	94.95%
Class 11	98.68%	100%	97.98%	97.98%	99.28%	92.61%
Class 12	96.88%	100%	100%	100%	100%	96.77%
Class 13	95%	67.85%	56.81%	89.28%	100%	39.28%
Class 14	84.05%	92.31%	100%	80%	100%	61.53%
Class 15	90.9%	43.48%	66.66%	52.17%	0%	0%
Class 16	90.62%	72.5%	90.16%	68.75%	100%	30%
Class 17	97.02%	98.24%	98.2%	95.74%	99.39%	82.46%
Class 18	87.34%	90.79%	71.28%	94.74%	77.77%	82.89%
Class 19	92.75%	84.21%	78.57%	72.36%	31.11%	18.42%
Class 20	91.86%	94.58%	87.59%	90.14%	26.86%	99.51%
Class 21	100%	62.5%	84.61%	45.83%	0%	0%
Class 22	91.66%	91.66%	85.71%	100%	100%	29.16%
Class 23	100%	96.66%	76.92%	100%	100%	60%
Class 24	100%	95.45%	30.98%	100%	100%	77.27%

Table A.3: Precision and Recall figures for User 1 using the Bag-of-Keypoints approach for all classifiers with U-SURF128 features

Setting	Precision (SVM)	Recall (SVM)	Precision (MLP)	Recall(MLP)	Precision (KNN)	Recall (KNN)
Class 1	87.13%	82.67%	92.44%	72.44%	96.43%	31.88%
Class 2	95%	65.52%	100%	48.27%	0%	0%
Class 3	85.91%	85.09%	92.56%	79.27%	26.73%	94.71%
Class 4	73.91%	62.96%	85.71%	22.22%	0%	0%
Class 5	65	41.93%	100%	48.39%	0%	0%
Class 6	83.87%	78.78%	83.33%	75.75%	0%	0%
Class 7	77.55%	52.77%	91.89%	47.22%	100%	1.38%
Class 8	73.02%	94.86%	88.02%	89.25%	76.41%	75.7%
Class 9	74.82%	85.71%	78.02%	85.17%	51.94%	75.74%
Class 10	90.63%	82.85%	100%	74.29%	100%	54.28%
Class 11	100%	96.15%	100%	69.23%	100%	19.23%
Class 12	100%	91.3%	100%	82.61%	100%	47.83%
Class 13	70.59%	81.81%	90.91%	45.45%	100%	9.09%
Class 14	86.66%	86.66%	96.42%	90%	93.75%	50%
Class 15	89.23%	80.55%	94.74%	75%	97.56%	55.55%
Class 16	53.37%	66.92%	64.61%	56.15%	0%	0%
Class 17	97.29%	92.3%	100%	87.18%	93.93%	79.49%
Class 18	61.37%	72.93%	74.73%	56.96%	23.86%	51.64%
Class 19	65.38%	53.96%	75.75%	39.68%	0%	0%
Class 20	40.74%	30.14%	78.57%	30.13%	100%	6.84%
Class 21	91.49%	72.88%	97.61%	69.49%	0%	0%
Class 22	72.97%	49.09%	75.67%	25.45%	75%	5.45%
Class 23	88.93%	93.79%	88.06%	86.13%	97.57%	58.76%
Class 24	100%	95.23%	100%	47.61%	95.45%	100%
Class 25	50%	44%	62.5%	20%	0%	0%
Class 26	61.34%	63.62%	53.64%	70.58%	84.35%	32.9%
Class 27	84.9%	95.74%	83.33%	95.74%	100%	40.42%
Class 28	92.54%	96.87%	83.56%	95.31%	100%	28.13%
Class 29	87.94%	84.95%	88.3%	81.56%	93.77%	47.88%
Class 30	78.26%	35.29%	58.33%	13.72%	0%	0%
Class 31	98.68%	100%	86.21%	100%	100%	100%
Class 32	76.92%	66.66%	70.68%	68.33%	100%	8.33%
Class 33	74.19%	63.01%	78.68%	65.75%	0%	0%
Class 34	62.74%	65.31%	50.79%	65.31%	0%	0%
Class 35	96%	92.31%	95%	73.07%	100%	38.46%
Class 36	62.69%	70%	44.44%	60%	50%	1.66%
Class 37	91.94%	78.08%	74.82%	78.08%	88.71%	75.34%
Class 38	74.21%	86.16%	52.25%	85.29%	94.03%	46.32%
Class 39	39.5%	33.68%	42.64%	30.52%	0%	0%
Class 40	80.36%	78.26%	50.24%	90.43%	100%	8.69%
Class 41	76.19%	69.57%	22.22%	69.56%	0%	0%
Class 42	65.21%	41.66%	11.72%	77.77%	0%	0%

Table A.4: Precision and Recall figures for User 2 using the Bag-of-Keypoints approach for all classifiers with SIFT features

Setting	Precision (SVM)	Recall (SVM)	Precision (MLP)	Recall(MLP)	Precision (KNN)	Recall (KNN)
Class 1	86.52%	78.34%	81.69%	72.04%	97.05%	38.98%
Class 2	69.56%	55.17%	37.93%	37.93%	100%	6.89%
Class 3	84.4%	90.92%	80.34%	83.06%	86.87%	49.32%
Class 4	62.16%	85.18%	58.97%	85.18%	0%	0%
Class 5	59.52	80.65%	51.11%	74.19%	0%	0%
Class 6	82.35%	84.84%	62.22%	84.84%	100%	24.24%
Class 7	60.46%	72.22%	57.5%	63.88%	68%	23.61%
Class 8	68.13%	86.91%	62.05%	73.36%	10.03%	99.06%
Class 9	80.9%	86.79%	76.9%	84.36%	9.45%	6.47%
Class 10	64.28%	51.43%	60%	51.43%	20.33%	34.28%
Class 11	82.14%	88.46%	81.81%	69.23%	100%	46.15%
Class 12	100%	91.3%	100%	91.3%	0%	0%
Class 13	90%	81.81%	83.33%	45.45%	80%	9.09%
Class 14	90.9%	100%	90.9%	100%	100%	43.33%
Class 15	57.14%	27.77%	56.25%	25%	0.7%	1.38%
Class 16	78.38%	89.23%	77.33%	89.23%	100%	49.23%
Class 17	80%	82.05%	74.19%	58.97%	96.15%	64.1%
Class 18	75.31%	75%	62.73%	67.62%	72.22%	10.65%
Class 19	87.09%	85.71%	67.24%	61.91%	89.47%	26.98%
Class 20	63.16%	49.31%	59.02%	49.32%	75%	16.43%
Class 21	97.36%	62.71%	97.36%	62.71%	100%	32.2%
Class 22	52.54%	28.18%	34.06%	28.18%	1.98%	4.54%
Class 23	76.59%	91.97%	70.78%	91.97%	61.47%	51.82%
Class 24	95.24%	95.24%	95.23%	95.23%	100%	80.95%
Class 25	56.41%	44%	47.82%	44%	0%	0%
Class 26	65.99%	64.27%	60.17%	61.87%	33.69%	40.09%
Class 27	95.74%	95.74%	94.73%	76.59%	100%	38.29%
Class 28	84.93%	96.87%	84.93%	96.87%	88.46%	35.93%
Class 29	88.01%	90.25%	81.26%	79.02%	82.97%	24.78%
Class 30	88.23%	58.82%	82.35%	27.45%	0%	0%
Class 31	98.68%	100%	98.36%	80%	92.21%	94.66%
Class 32	88.37%	63.33%	73.68%	23.33%	100%	33.33%
Class 33	85.48%	72.6%	84.21%	65.75%	86.66%	17.81%
Class 34	59.7%	81.63%	25%	34.69%	100%	18.36%
Class 35	83.33%	76.92%	76.92%	76.92%	0%	0%
Class 36	71.15%	61.66%	57.14%	60%	42.85%	5%
Class 37	85.93%	75.34%	85.93%	75.34%	95.83%	31.5%
Class 38	87.02%	83.82%	68.79%	71.32%	94.28%	24.26%
Class 39	86.48%	67.37%	82.05%	67.36%	0%	0%
Class 40	86.41%	77.39%	85.57%	77.39%	31.94%	60%
Class 41	95%	82.61%	94.28%	71.73%	100%	30.43%
Class 42	69.23%	75%	43.54%	75%	100%	2.77%

Table A.5: Precision and Recall figures for User 2 using the Bag-of-Keypoints approach for all classifiers with U-SURF64 features

Setting	Precision (SVM)	Recall (SVM)	Precision (MLP)	Recall(MLP)	Precision (KNN)	Recall (KNN)
Class 1	80.52%	73.22%	93.13%	74.8%	94.23%	38.58%
Class 2	63.63%	72.41%	100%	51.72%	0%	0%
Class 3	85.71%	93.49%	86.11%	87.39%	91.13%	39.02%
Class 4	78.12%	92.59%	100%	48.14%	0%	0%
Class 5	60.53	74.19%	80%	25.8%	33.33%	3.22%
Class 6	85.29%	87.87%	100%	81.81%	100%	15.15%
Class 7	57.35%	54.16%	92.68%	52.77%	80%	11.11%
Class 8	71.86%	88.31%	58.89%	69.62%	7.67%	100%
Class 9	81.31%	93.8%	78.19%	84.09%	6.17%	2.96%
Class 10	75%	51.43%	88.88%	45.71%	11.21%	34.28%
Class 11	76.92%	76.92%	100%	80.76%	100%	15.38%
Class 12	85.71%	78.26%	100%	78.26%	0%	0%
Class 13	84.78%	88.63%	94.11%	72.72%	75%	6.81%
Class 14	93.75%	100%	93.75%	100%	100%	43.33%
Class 15	41.66%	20.83%	78.26%	25%	0%	0%
Class 16	84.52%	100%	72.66%	83.84%	100%	16.92%
Class 17	85.71%	61.53%	95.65%	56.41%	100%	56.41%
Class 18	66.02%	70.08%	71.73%	67.62%	80%	16.39%
Class 19	80.59%	85.71%	74.64%	84.12%	100%	11.11%
Class 20	69.49%	56.16%	79.59%	53.42%	0%	0%
Class 21	83.33%	59.32%	100%	71.18%	100%	27.11%
Class 22	60.71%	46.36%	75.36%	47.27%	0.78%	0.9%
Class 23	88.85%	95.98%	90.67%	88.68%	63.59%	47.81%
Class 24	95.23%	95.23%	100%	85.71%	93.33%	66.66%
Class 25	58.33%	56%	67.64%	46%	0%	0%
Class 26	69.52%	67.1%	68.89%	65.14%	41.47%	35.51%
Class 27	91.66%	93.61%	95.23%	85.1%	93.75%	31.91%
Class 28	84.5%	93.75%	90.62%	90.62%	95.83%	35.93%
Class 29	89.87%	94.06%	89.1%	88.34%	90.32%	11.86%
Class 30	73.07%	37.25%	72.97%	52.94%	0%	0%
Class 31	98.66%	98.66%	98.63%	96%	97.05%	88%
Class 32	78.57%	55%	66.17%	75%	100%	30%
Class 33	76.66%	63.01%	67.46%	76.71%	76.92%	13.69%
Class 34	79.59%	79.59%	43.47%	81.63%	0%	0%
Class 35	71.27%	65.38%	68.96%	76.92%	0%	0%
Class 36	58.97%	38.33%	66.66%	43.33%	0%	0%
Class 37	79.1%	72.6%	78.12%	68.49%	100%	32.87%
Class 38	76.59%	79.41%	75.31%	89.71%	30.55%	8.08%
Class 39	72.15%	60%	42.17%	65.26%	100%	4.21%
Class 40	80.58%	72.17%	63.35%	72.17%	24.63%	44.34%
Class 41	79.48%	67.39%	43.75%	91.3%	100%	6.52%
Class 42	62.5%	41.66%	20.12%	88.88%	0%	0%

Table A.6: Precision and Recall figures for User 2 using the Bag-of-Keypoints approach for all classifiers with U-SURF128 features

Setting	Precision (SVM)	Recall (SVM)	Precision (MLP)	Recall(MLP)	Precision (KNN)	Recall (KNN)
Class 1	84.64%	97.21%	88.35%	95.62%	61.55%	95.75%
Class 2	56.62%	43.11%	73.91%	15.59%	25.17%	33.94%
Class 3	81.48%	50%	75.75%	56.81%	100%	25%
Class 4	78.63%	87.61%	83.33%	85.71%	94.93%	71.42%
Class 5	88%	78.57%	100%	39.28%	100%	50%
Class 6	100%	82.14%	100%	78.57%	100%	78.57%
Class 7	90.9%	49.38%	97.56%	49.38%	100%	43.21%
Class 8	95.08%	92.06%	98.03%	79.36%	92.18%	93.65%
Class 9	67.33%	59.06%	74.57%	51.46%	100%	13.45%
Class 10	73.77%	45%	80.48%	33.33%	57.14%	8%
Class 11	82.82%	92.13%	86.91%	93.25%	95%	85.39%
Class 12	77.41%	88.88%	85.14%	78.83%	74.07%	74.07%
Class 13	78.57%	84.61%	88.52%	69.23%	45.67%	47.43%
Class 14	73.77%	62.5%	64.78%	63.88%	80%	50%
Class 15	63.33%	77.86%	42.24%	80.32%	80%	52.45%
Class 16	66.26%	68.27%	63.73%	72.75%	71.83%	29.56%
Class 17	77.76%	77.76%	81.54%	74.77%	62.57%	89.71%
Class 18	70.27%	40.62%	43.24%	75%	30.76%	12.5%
Class 19	91.35%	88.09%	66.66%	88.09%	100%	73.81%
Class 20	100%	34.69%	39.39%	79.59%	97.91%	95.91%

Table A.7: Precision and Recall figures for User 3 using the Bag-of-Keypoints approach for all classifiers with SIFT features

Setting	Precision (SVM)	Recall (SVM)	Precision (MLP)	Recall(MLP)	Precision (KNN)	Recall (KNN)
Class 1	85.66%	94.3%	93.76%	91.77%	96.13%	79.17%
Class 2	50.54%	43.12%	55.07%	34.86%	31.81%	12.84%
Class 3	91.3%	95.45%	97.29%	81.81%	85%	38.63%
Class 4	83.16%	80%	86.17%	77.14%	95.23%	38.09%
Class 5	87.09%	96.43%	100%	82.14%	41.66%	17.86%
Class 6	48%	85.71%	95.24%	71.42%	52.38%	39.28%
Class 7	88%	54.32%	92.45%	96.07%	100%	43.21%
Class 8	95%	90.48%	100%	93.65%	100%	82.53%
Class 9	77.54%	84.79%	80.92%	81.87%	84.51%	35.09%
Class 10	92.47%	86%	89.47%	68%	100%	8%
Class 11	92.43%	96.06%	92.26%	93.82%	47.29%	78.65%
Class 12	94.24%	95.23%	90.09%	96.29%	97.9%	74.07%
Class 13	75.67%	71.79%	71.71%	91.02%	100%	29.49%
Class 14	72.22%	54.16%	75.43%	59.72%	100%	23.61%
Class 15	82.57%	89.34%	79.26%	87.7%	94.87%	60.65%
Class 16	87.5%	86.56%	82.62%	86.02%	83.07%	48.38%
Class 17	86.87%	84.18%	89.29%	82.96%	45.01%	91.37%
Class 18	65.85%	42.18%	39.82%	70.31%	54.16%	20.31%
Class 19	86.36%	90.47%	45.65%	100%	97.01%	77.38%
Class 20	100%	95.92%	100%	93.87%	100%	95.92%

Table A.8: Precision and Recall figures for User 3 using the Bag-of-Keypoints approach for all classifiers with U-SURF64 features

Setting	Precision (SVM)	Recall (SVM)	Precision (MLP)	Recall(MLP)	Precision (KNN)	Recall (KNN)
Class 1	84.09%	94.69%	92.67%	92.31%	92.24%	77.32%
Class 2	65.21%	55.04%	86.95%	36.69%	57.14%	3.66%
Class 3	91.11%	93.18%	100%	90.9%	100%	13.63%
Class 4	85.57%	84.76%	98.78%	77.14%	94.44%	32.38%
Class 5	81.48	78.57%	100%	42.85%	0%	0%
Class 6	86.66%	92.85%	68.29%	100%	93.33%	50%
Class 7	98%	60.49%	94.82%	67.9%	100%	48.15%
Class 8	93.22%	87.3%	100%	90.47%	100%	82.53%
Class 9	79.38%	90.06%	71.02%	88.88%	94.82%	32.16%
Class 10	91.95%	80%	84.4%	92%	100%	3%
Class 11	93.47%	96.63%	92.73%	93.25%	30.04%	80.34%
Class 12	96.33%	97.35%	91.32%	94.71%	96.12%	65.61%
Class 13	90%	92.31%	91.55%	83.33%	100%	16.66%
Class 14	61.9%	54.16%	55.43%	70.83%	100%	36.11%
Class 15	77.53%	87.71%	72%	88.52%	100%	27.86%
Class 16	90.65%	86.91%	90.49%	85.31%	91.15%	18.46%
Class 17	87.28%	85.84%	89.39%	87.61%	37.98%	85.29%
Class 18	77.14%	42.18%	42.99%	71.87%	6.25%	1.56%
Class 19	93.9%	91.66%	66.66%	92.86%	97.91%	55.95%
Class 20	100%	95.91%	100%	95.92%	100%	95.92%

Table A.9: Precision and Recall figures for User 3 using the Bag-of-Keypoints approach for all classifiers with U-SURF128 features

Setting	Precision (SVM)	Recall (SVM)	Precision (MLP)	Recall(MLP)	Precision (KNN)	Recall (KNN)
Class 1	100%	65.55%	100%	51.11%	85.98%	51.11%
Class 2	92.85%	72.22%	100%	69.44%	33.33%	2.77%
Class 3	96.64%	98.8%	97.31%	98.29%	90.29%	100%
Class 4	79.28%	98.23%	86.84%	87.61%	81.29%	100%
Class 5	94.66	98.61%	98.46%	88.88%	91.17%	86.11%
Class 6	97.95%	92.31%	95.45%	80.76%	87.87%	55.76%
Class 7	100%	94.87%	100%	94.87%	94.59%	89.74%
Class 8	89.81%	97.97%	100%	80.8%	75%	57.57%
Class 9	98.93%	100%	93.93%	100%	93.25%	89.24%
Class 10	100%	100%	95.71%	100%	96.92%	94.02%
Class 11	94.44%	77.27%	87.5%	63.63%	0%	0%
Class 12	93.87%	83.63%	94.25%	74.54%	80.59%	49.09%
Class 13	96.86%	96.26%	88.1%	96.88%	89.8%	71.33%
Class 14	69.56%	42.66%	49.35%	50.66%	71.42%	13.33%
Class 15	86.36%	98.27%	44.11%	98.27%	87.02%	98.27%
Class 16	98.64%	98.64%	96.05%	98.64%	95.52%	86.48%

Table A.10: Precision and Recall figures for User 4 using the Bag-of-Keypoints approach for all classifiers with SIFT features

Setting	Precision (SVM)	Recall (SVM)	Precision (MLP)	Recall(MLP)	Precision (KNN)	Recall (KNN)
Class 1	95%	52.77%	100%	36.66%	97.91%	52.22%
Class 2	91.17%	86.11%	100%	86.11%	80%	22.22%
Class 3	95.61%	98.74%	95.18%	98.19%	96.46%	92.82%
Class 4	85.6%	94.69%	96.26%	91.15%	68.81%	56.63%
Class 5	92	95.83%	94.59%	97.22%	100%	79.16%
Class 6	92%	88.46%	92.59%	96.15%	100%	30.76%
Class 7	100%	97.43%	100%	87.17%	100%	92.3%
Class 8	96.9%	94.94%	100%	87.87%	100%	74.74%
Class 9	98.86%	93.54%	100%	93.54%	100%	94.62%
Class 10	100%	97.01%	88%	98.5%	100%	67.16%
Class 11	100%	77.27%	100%	81.81%	100%	50%
Class 12	81.19%	86.36%	95.18%	71.81%	63.46%	30%
Class 13	91.83%	87.53%	94.23%	86.6%	36.91%	98.44%
Class 14	95.12%	52%	53.19%	66.66%	96.15%	33.33%
Class 15	83.58%	96.55%	80.29%	94.82%	84.52%	61.21%
Class 16	84.88%	98.64%	54.47%	98.64%	86.11%	41.89%

Table A.11: Precision and Recall figures for User 4 using the Bag-of-Keypoints approach for all classifiers with U-SURF64 features

Setting	Precision (SVM)	Recall (SVM)	Precision (MLP)	Recall(MLP)	Precision (KNN)	Recall (KNN)
Class 1	94.44%	56.66%	94.31%	46.11%	97.46%	42.77%
Class 2	96.87%	86.11%	96.87%	86.11%	90%	25%
Class 3	96.34%	98.67%	97.08%	97.48%	97.5%	91.59%
Class 4	88.13%	92.03%	87.03%	83.18%	77.14%	47.78%
Class 5	84.21	88.88%	95.16%	81.94%	100%	51.38%
Class 6	92.31%	92.31%	77.77%	94.23%	100%	25%
Class 7	100%	97.43%	100%	94.87%	100%	92.31%
Class 8	94.73%	90.91%	96.51%	83.83%	100%	74.74%
Class 9	100%	93.54%	96.59%	91.39%	100%	91.39%
Class 10	98.51%	98.51%	92.95%	98.51%	100%	73.13%
Class 11	95%	86.36%	79.16%	86.36%	100%	63.63%
Class 12	77.05%	85.45%	82.08%	79.09%	74.41%	29.09%
Class 13	92.92%	90.03%	95.57%	80.68%	29.88%	100%
Class 14	97.77%	58.66%	88.67%	62.66%	100%	17.33%
Class 15	81.56%	99.13%	55.61%	89.65%	81.25%	22.41%
Class 16	83.72%	97.29%	39.77%	97.29%	100%	24.32%

Table A.12: Precision and Recall figures for User 4 using the Bag-of-Keypoints approach for all classifiers with U-SURF128 features

Setting	Precision (SVM)	Recall (SVM)	Precision (MLP)	Recall(MLP)	Precision (KNN)	Recall (KNN)
Class 1	97.89%	98.93%	98.93%	100%	78.99%	100%
Class 2	98.68%	100%	100%	94.66%	97.26%	94.66%
Class 3	98.31%	99.43%	100%	97.15%	89.74%	99.43%
Class 4	100%	98.31%	100%	96.61%	100%	94.91%
Class 5	97.26	100%	93.42%	100%	100%	100%
Class 6	100%	100%	99.47%	100%	100%	100%
Class 7	99.37%	98.76%	100%	95.06%	100%	90.12%
Class 8	98.66%	98%	96.75%	99.33%	100%	79.33%
Class 9	99.67%	100%	99.01%	99.33%	96.19%	100%
Class 10	100%	98.21%	96.12%	99.55%	97.27%	95.98%

Table A.13: Precision and Recall figures for User 5 using the Bag-of-Keypoints approach for all classifiers with SIFT features

Setting	Precision (SVM)	Recall (SVM)	Precision (MLP)	Recall(MLP)	Precision (KNN)	Recall (KNN)
Class 1	94.89%	98.93%	98.93%	98.93%	37.75%	100%
Class 2	98.68%	100%	100%	100%	94.74%	48%
Class 3	97.72%	97.72%	99.41%	95.45%	37.5%	8.52%
Class 4	100%	100%	100%	100%	100%	98.31%
Class 5	98.61	100%	97.26%	100%	100%	97.18%
Class 6	100%	99.47%	98.45%	100%	100%	96.33%
Class 7	100%	100%	98.76%	98.76%	100%	96.91%
Class 8	100%	100%	100%	99.33%	100%	91.33%
Class 9	99.33%	99%	100%	97.35%	72.38%	96.04%
Class 10	100%	98.21%	93.47%	99.11%	100%	76.34%

Table A.14: Precision and Recall figures for User 5 using the Bag-of-Keypoints approach for all classifiers with U-SURF64 features

Setting	Precision (SVM)	Recall (SVM)	Precision (MLP)	Recall(MLP)	Precision (KNN)	Recall (KNN)
Class 1	95.92%	100%	100%	98.83%	33.09%	98.93%
Class 2	96.1%	98.66%	100%	97.33%	95.23%	80%
Class 3	98.27%	97.16%	100%	90.9%	52%	7.38%
Class 4	100%	98.31%	95.61%	100%	100%	98.31%
Class 5	100	98.59%	100%	94.36%	100%	97.18%
Class 6	100%	99.47%	100%	100%	100%	93.19%
Class 7	100%	99.38%	91.52%	100%	100%	95.67%
Class 8	100%	100%	100%	98%	100%	89.33%
Class 9	99.33%	100%	100%	98.02%	79.08%	97.35%
Class 10	98.66%	98.66%	93.69%	99.55%	99.41%	75%

Table A.15: Precision and Recall figures for User 5 using the Bag-of-Keypoints approach for all classifiers with U-SURF128 features

Setting	Precision (SVM)	Recall (SVM)	Precision (MLP)	Recall(MLP)	Precision (KNN)	Recall (KNN)
Class 1	99.19%	76.39%	100%	63.04%	96%	81.98%
Class 2	88%	69.84%	95.45%	33.33%	0%	0%
Class 3	96.4%	99.58%	99.51%	96.11%	86.33%	89.67%
Class 4	90.82%	92.61%	100%	46.31%	70.24%	70.93%
Class 5	95.04	89.84%	100%	20.31%	96.72%	92.18%
Class 6	96.05%	79.34%	100%	51.08%	100%	3.26%
Class 7	100%	95.65%	100%	55.07%	100%	82.61%
Class 8	88.58%	93.14%	96.92%	36%	98.38%	34.85%
Class 9	98.78%	96.42%	96.82%	72.61%	98.91%	54.16%
Class 10	100%	100%	72.89%	100%	100%	99.17%
Class 11	93.33%	35%	81.81%	22.5%	100%	2.5%
Class 12	95.32%	82.74%	96.62%	72.58%	100%	6.09%
Class 13	95.13%	97.75%	87.31%	96.2%	99.79%	83.07%
Class 14	85.86%	58.52%	21.69%	87.41%	94.23%	36.29%
Class 15	84.02%	89.32%	36.39%	90.29%	56.82%	86.89%
Class 16	98.41%	93.93%	66.27%	86.36%	100%	9.84%

Table A.16: Precision and Recall figures for User 4 using the Bag-of-Keypoints approach for all classifiers with SIFT features and data split into 10% training data, 90% testing data

Setting	Precision (SVM)	Recall (SVM)	Precision (MLP)	Recall(MLP)	Precision (KNN)	Recall (KNN)
Class 1	97.74%	67.39%	98.51%	61.49%	100%	61.49%
Class 2	89.06%	90.47%	91.43%	50.79%	0%	0%
Class 3	96.91%	99.09%	98.65%	96.56%	97.32%	90.37%
Class 4	78.4%	96.55%	93.38%	62.56%	32.32%	15.76%
Class 5	94.06	74.21%	100%	21.09%	100%	49.21%
Class 6	97.4%	81.52%	75.55%	73.91%	0%	0%
Class 7	98.46%	92.75%	98.14%	76.81%	100%	59.42%
Class 8	92.26%	95.42%	80%	96%	96.49%	31.43%
Class 9	99.39%	97.02%	100%	85.11%	98.46%	76.19%
Class 10	98.26%	93.38%	88.62%	90.08%	100%	61.15%
Class 11	100%	72.5%	82.5%	82.5%	100%	2.5%
Class 12	64.11%	80.71%	59.07%	77.66%	22.72%	5.07%
Class 13	87.86%	88.77%	83.79%	77.72%	26.18%	99.48%
Class 14	100%	65.92%	65.1%	92.59%	80%	8.88%
Class 15	92.11%	90.77%	53.11%	95.14%	94.66%	34.46%
Class 16	98.3%	87.87%	39.57%	97.72%	100%	8.33%

Table A.17: Precision and Recall figures for User 4 using the Bag-of-Keypoints approach for all classifiers with U-SURF64 features and data split into 10% training data, 90% testing data

Setting	Precision (SVM)	Recall (SVM)	Precision (MLP)	Recall(MLP)	Precision (KNN)	Recall (KNN)
Class 1	99.13%	70.81%	99.54%	68.63%	100%	52.48%
Class 2	90.76%	93.65%	95.55%	68.25%	0%	0%
Class 3	97.13%	99.16%	98.81%	95.45%	98.41%	81.98%
Class 4	77.68%	89.16%	89.71%	47.29%	11.53%	1.47%
Class 5	100%	60.93%	93.75%	11.72%	100%	31.25%
Class 6	90%	78.26%	50%	79.16%	0%	0%
Class 7	95.71%	97.1%	92.45%	71.01%	100%	60.86%
Class 8	95.83%	92%	95.65%	75.43%	100%	21.14%
Class 9	100%	94.04%	100%	85.71%	100%	76.78%
Class 10	98.07%	84.29%	97.43%	94.21%	100%	33.88%
Class 11	96.87%	97.5%	81.81%	90%	0%	0%
Class 12	66.66%	87.31%	52.04%	77.66%	0%	0%
Class 13	84%	90.67%	77.68%	82.38%	19.22%	99.65%
Class 14	99%	73.33%	69.71%	73.33%	0%	0%
Class 15	86.75%	91.34%	30.96%	94.23%	100%	10.09%
Class 16	100%	75%	71.91%	48.48%	100%	3.03%

Table A.18: Precision and Recall figures for User 4 using the Bag-of-Keypoints approach for all classifiers with U-SURF128 features and data split into 10% training data, 90% testing data

Setting	Precision (SVM)	Recall (SVM)	Precision (MLP)	Recall(MLP)	Precision (KNN)	Recall (KNN)
Class 1	99.47%	71.96%	100%	53.41%	100%	69.69%
Class 2	97.22%	70%	100%	44%	0%	0%
Class 3	96.97%	99.31%	98.61%	98.41%	89.18%	99.68%
Class 4	86.28%	93.78%	97.61%	76.39%	76.56%	91.31%
Class 5	91.66%	98.01%	98.95%	94.05%	91.43%	95.05%
Class 6	95.58%	90.27%	95%	79.16%	100%	50%
Class 7	100%	96.36%	98.21%	100%	100%	94.54%
Class 8	91.19%	98.63%	84.79%	98.63%	100%	63.94%
Class 9	100%	99.23%	98.81%	63.35%	99.18%	92.36%
Class 10	100%	100%	100%	88.37%	100%	97.67%
Class 11	95%	61.29%	87.5%	45.16%	0%	0%
Class 12	93.37%	89.24%	94.89%	82.27%	100%	64.55%
Class 13	96.03%	97.53%	93.36%	94.41%	100%	78.97%
Class 14	83.09%	56.19%	74.51%	36.19%	100%	30.47%
Class 15	89.72%	94.92%	66.32%	94.2%	81.33%	88.41%
Class 16	100%	99.02%	27.61%	100%	100%	58.25%

Table A.19: Precision and Recall figures for User 4 using the Bag-of-Keypoints approach for all classifiers with SIFT features and data split into 30% training data, 70% testing data

Setting	Precision (SVM)	Recall (SVM)	Precision (MLP)	Recall(MLP)	Precision (KNN)	Recall (KNN)
Class 1	96.31%	59.46%	99.34%	57.57%	100%	61.74%
Class 2	89.13%	82%	100%	66%	76.92%	20%
Class 3	96.16%	99%	97.53%	98.24%	96.98%	92.61%
Class 4	81.56%	90.68%	93.18%	76.39%	64.28%	44.72%
Class 5	90.82	88.11%	85.29%	86.13%	100%	63.36%
Class 6	91.66%	91.66%	98.07%	70.83%	87.5%	9.72%
Class 7	100%	98.18%	94.64%	96.36%	97.36%	67.27%
Class 8	96.59%	96.59%	85.54%	96.59%	98.83%	57.82%
Class 9	100%	96.18%	99.21%	95.41%	100%	96.18%
Class 10	95.45%	97.67%	95.29%	94.18%	100%	72.09%
Class 11	100%	77.41%	100%	70.96%	100%	48.38%
Class 12	77.22%	87.97%	63.36%	81.01%	61.42%	27.21%
Class 13	92.98%	91.94%	91.31%	89.26%	32.35%	98.65%
Class 14	96.15%	71.42%	61.53%	76.19%	100%	28.57%
Class 15	79.83%	68.84%	73.84%	69.56%	81.81%	45.65%
Class 16	91.07%	99.02%	48.71%	91.26%	57.14%	3.88%

Table A.20: Precision and Recall figures for User 4 using the Bag-of-Keypoints approach for all classifiers with U-SURF64 features and data split into 30% training data, 70% testing data

Setting	Precision (SVM)	Recall (SVM)	Precision (MLP)	Recall(MLP)	Precision (KNN)	Recall (KNN)
Class 1	96.57%	64.01%	99.34%	57.57%	99.31%	55.3%
Class 2	97.56%	80%	100%	56%	78.57%	22%
Class 3	97.04%	99.32%	97.07%	98.61%	98.01%	91.42%
Class 4	79.32%	88.19%	96.84%	57.14%	72.6%	32.91%
Class 5	92.77%	76.23%	97.46%	76.23%	100%	31.68%
Class 6	95.45%	87.5%	84.81%	93.05%	100%	4.16%
Class 7	98.18%	98.18%	96.29%	94.54%	100%	81.81%
Class 8	95.89%	95.23%	92.11%	95.23%	100%	42.85%
Class 9	99.19%	93.89%	99.15%	90.83%	100%	91.6%
Class 10	98.79%	95.34%	95.4%	96.51%	100%	74.41%
Class 11	93.1%	87.09%	61.22%	96.71%	100%	54.83%
Class 12	73.19%	89.87%	87.05%	76.58%	66.66%	15.19%
Class 13	92.32%	91.49%	94.1%	89.26%	26.87%	100%
Class 14	93.82%	72.38%	58.08%	75.23%	100%	20%
Class 15	80.95%	86.23%	62.82%	86.95%	70.37%	13.76%
Class 16	93.45%	97.08%	50.75%	98.05%	0%	0%

Table A.21: Precision and Recall figures for User 4 using the Bag-of-Keypoints approach for all classifiers with U-SURF128 features and data split into 30% training data, 70% testing data

APPENDIX B

Precision / Recall for Alternate Approach

Setting	Precision (SIFT)	Recall (SIFT)	Precision (U-SURF64)	Recall(U-SURF64)	Precision (U-SURF128)	Recall (U-SURF128)
Class 1	91.62%	53.35%	98.7%	57.92%	95.8%	48.78%
Class 2	12.94%	46.77%	6%	19.35%	6.52%	19.35%
Class 3	87.13%	69.76%	92.13%	81.72%	66.77%	68.11%
Class 4	0%	0%	91.42%	65.31%	93.75%	45.91%
Class 5	100	35.61%	90.41%	90.41%	81.71%	91.78%
Class 6	94.69%	61.51%	97.65%	80.41%	98.66%	81.79%
Class 7	92.31%	70.58%	80%	23.52%	0%	0%
Class 8	64.13%	55.35%	59.31%	81.54%	0%	0%
Class 9	11.08%	81.63%	2.19%	18.36%	2%	4.08%
Class 10	65.59%	36.11%	66.66%	19.19%	0%	0%
Class 11	48.96%	63.75%	86.51%	73.15%	0%	0%
Class 12	100%	64.51%	0%	0%	0%	0%
Class 13	3.95%	46.42%	0%	0%	0%	0%
Class 14	100%	60%	0%	0%	0%	0%
Class 15	0%	0%	0%	0%	0%	0%
Class 16	81.13%	53.75%	0%	0%	0%	0%
Class 17	100%	43.11%	0%	0%	99.02%	76.44%
Class 18	7.26%	39.47%	33.03%	48.68%	43.75%	82.89%
Class 19	86.66%	34.21%	60%	23.68%	18.18%	13.15%
Class 20	53.44%	15.27%	53.52%	63.54%	2.01%	3.44%
Class 21	0.91%	8.33%	0.75%	12.5%	0.68%	16.66%
Class 22	90.9%	41.66%	87.5%	58.33%	100%	91.66%
Class 23	14.1%	73.33%	12.38%	90%	0%	0%
Class 24	0%	0%	3.12%	4.54%	64%	72.72%

Table B.1: Precision and Recall figures for User 1 using the alternate approach for all descriptors

Setting	Precision (SIFT)	Recall (SIFT)	Precision (U-SURF64)	Recall(U-SURF64)	Precision (U-SURF128)	Recall (U-SURF128)
Class 1	61.07%	35.82%	81.17%	76.37%	87.55%	85.82%
Class 2	83.33%	34.48%	0%	0%	0%	0%
Class 3	100%	17.34%	99.24%	35.63%	100%	33.06%
Class 4	100%	25.92%	100%	40.74%	92.85%	48.14%
Class 5	100	22.58%	70%	45.16%	83.33%	64.51%
Class 6	100%	6.06%	61.11%	66.66%	0%	0%
Class 7	97.36%	51.38%	0%	0%	0%	0%
Class 8	30.14%	85.51%	5.66%	33.64%	4.11%	27.57%
Class 9	100%	63.34%	100%	15.9%	100%	47.71%
Class 10	3.44%	8.57%	2.31%	20%	1.59%	8.57%
Class 11	10.81%	30.76%	0%	0%	0%	0%
Class 12	0%	0%	0%	0%	100%	69.56%
Class 13	32%	18.18%	18.75%	54.54%	1.81%	2.27%
Class 14	83.33%	16.66%	0%	0%	76%	63.33%
Class 15	6.43%	15.27%	0%	0%	0%	0%
Class 16	9.81%	33.07%	0%	0%	61.66%	85.38%
Class 17	100%	30.76%	100%	58.97%	0%	0%
Class 18	61.62%	21.72%	66.07%	15.16%	93.54%	23.77%
Class 19	18.30%	88.88%	0.58%	1.58%	0.57%	1.58%
Class 20	0%	0%	0%	0%	94.59%	47.94%
Class 21	0%	0%	0%	0%	44.94%	67.79%
Class 22	0%	0%	1.81%	1.81%	0.76%	0.91%
Class 23	68.08%	23.35%	0.42%	0.36%	8.48%	8.75%
Class 24	66.66%	66.66%	0%	0%	0%	0%
Class 25	100%	26%	0%	0%	0%	0%
Class 26	93.13%	41.39%	49.51%	11.11%	63.46%	14.37%
Class 27	7.57%	21.27%	13.14%	70.21%	0.56%	2.12%
Class 28	17.39%	31.25%	0%	0%	0%	0%
Class 29	98.94%	39.61%	100%	57.62%	95.51%	63.13%
Class 30	60.46%	50.98%	38.57%	52.94%	0%	0%
Class 31	93.87%	61.33%	65.31%	42.66%	90.91%	53.33%
Class 32	0%	0%	15.78%	65%	11.82%	40%
Class 33	0%	0%	17.09%	64.38%	7.21%	34.24%
Class 34	0%	0%	90.62%	59.18%	96.29%	53.06%
Class 35	0.25%	3.84%	60.86%	53.84%	0%	0%
Class 36	96.66%	48.33%	75%	55%	20%	3.33%
Class 37	70.58%	32.87%	31.25%	41.09%	0%	0%
Class 38	100%	13.97%	94.23%	36.02%	0%	0%
Class 39	78.26%	37.89%	0%	0%	91.93%	60%
Class 40	46.77%	50.43%	57.25%	61.73%	6.81%	2.61%
Class 41	4.25%	21.73%	19.51%	17.39%	0%	0%
Class 42	2.5%	36.11%	0%	0%	0%	0%

Table B.2: Precision and Recall figures for User 2 using the alternate approach for all descriptors

Setting	Precision (SIFT)	Recall (SIFT)	Precision (U-SURF64)	Recall(U-SURF64)	Precision (U-SURF128)	Recall (U-SURF128)
Class 1	93.22%	58.35%	81.48%	72.94%	87.4%	74.53%
Class 2	63.46%	60.55%	11.11%	44.03%	15.9%	41.28%
Class 3	4.68%	56.81%	92.85%	29.54%	100%	13.63%
Class 4	20.9%	35.23%	13.23%	34.28%	23.78%	51.42%
Class 5	91.66	78.57%	4.67%	17.85%	7.29%	35.71%
Class 6	61.91%	46.42%	1.88%	3.57%	0%	0%
Class 7	68.75%	40.74%	30.3%	61.72%	22.58%	51.85%
Class 8	8.44%	50.79%	8.69%	57.14%	6.97%	80.95%
Class 9	8.52%	23.97%	0%	0%	0%	0%
Class 10	93.02%	40%	53.57%	30%	78.12%	50%
Class 11	96.69%	65.73%	62.16%	38.76%	82.75%	26.96%
Class 12	80.88%	29.1%	54.11%	59.25%	57.14%	27.51%
Class 13	100%	53.84%	41.37%	15.38%	100%	10.25%
Class 14	96%	33.33%	85.71%	50%	97.72%	59.72%
Class 15	91.66%	18.03%	30.9%	27.86%	35.95%	26.22%
Class 16	100%	30.64%	98.53%	60.39%	100%	61.11%
Class 17	100%	47.89%	94.65%	25.44%	96.24%	31.19%
Class 18	100%	65.62%	80%	37.5%	82.75%	37.5%
Class 19	90.9%	35.71%	0%	0%	0%	0%
Class 20	6.67%	73.46%	1.03%	8.16%	4.51%	42.85%

Table B.3: Precision and Recall figures for User 3 using the alternate approach for all descriptors

Setting	Precision (SIFT)	Recall (SIFT)	Precision (U-SURF64)	Recall(U-SURF64)	Precision (U-SURF128)	Recall (U-SURF128)
Class 1	70.23%	32.77%	83.14%	41.11%	51.51%	18.88%
Class 2	7.24%	13.88%	2.38%	13.88%	2.24%	22.22%
Class 3	99.12%	50.06%	99.86%	48.53%	99.86%	49.79%
Class 4	7.63%	28.31%	16.77%	23.01%	46.51%	35.39%
Class 5	4.59	25%	9.39%	51.38%	10.21%	38.88%
Class 6	9.03%	30.76%	6.52%	28.84%	7.97%	21.15%
Class 7	6.13%	25.64%	9.82%	56.41%	5.21%	41.02%
Class 8	25.92%	42.42%	31.89%	37.37%	48.42%	46.46%
Class 9	62.26%	35.48%	35%	7.52%	70.49%	46.23%
Class 10	6.36%	61.19%	5.01%	77.61%	5.68%	95.52%
Class 11	4%	22.72%	35.89%	63.63%	50%	50%
Class 12	73.33%	50%	0%	0%	66.66%	14.54%
Class 13	65.76%	53.27%	52.08%	38.94%	80.95%	37.07%
Class 14	100%	30.66%	85.71%	32%	94.87%	49.33%
Class 15	30.73%	61.21%	47.33%	68.96%	48.08%	75.86%
Class 16	72.34%	45.94%	56%	18.91%	19.04%	5.41%

Table B.4: Precision and Recall figures for User 4 using the alternate approach for all descriptors

Setting	Precision (SIFT)	Recall (SIFT)	Precision (U-SURF64)	Recall(U-SURF64)	Precision (U-SURF128)	Recall (U-SURF128)
Class 1	27.32%	50%	100%	50%	100%	40.42%
Class 2	26.26%	34.66%	27.93%	66.66%	23.52%	58.66%
Class 3	53.61%	59.09%	71.08%	33.52%	53.84%	23.86%
Class 4	15.94%	93.22%	69.01%	83.05%	75.67%	94.91%
Class 5	96.29	36.61%	80.48%	92.95%	57.14%	90.14%
Class 6	67.18%	22.51%	49.16%	46.07%	41.66%	41.88%
Class 7	96.7%	54.32%	32.14%	16.66%	16.12%	9.25%
Class 8	50%	18%	0%	0%	41.74%	57.33%
Class 9	80.68%	70.29%	35.69%	34.98%	41.1%	22.11%
Class 10	77.43%	67.41%	55.58%	95.53%	58.83%	95.08%

Table B.5: Precision and Recall figures for User 5 using the alternate approach for all descriptors

Setting	Precision (SIFT)	Recall (SIFT)	Precision (U-SURF64)	Recall(U-SURF64)	Precision (U-SURF128)	Recall (U-SURF128)
Class 1	58.06%	11.18%	81.25%	16.14%	57.81%	11.49%
Class 2	3.33%	3.17%	2.91%	11.11%	2.7%	20.63%
Class 3	91.52%	2.03%	92.94%	4.22%	94.41%	3.51%
Class 4	4.18%	12.31%	16.66%	9.85%	22.22%	6.89%
Class 5	3.51	14.06%	7.23%	27.34%	5.52%	15.62%
Class 6	7.78%	14.13%	1.63%	4.34%	7.83%	14.13%
Class 7	1.23%	21.73%	3.17%	11.59%	2.77%	14.49%
Class 8	21.52%	17.71%	24%	13.71%	0%	0%
Class 9	47.5%	11.31%	15.38%	1.19%	0%	0%
Class 10	1.32%	11.57%	1.18%	22.31%	0%	0%
Class 11	2.63%	7.5%	3.57%	2.5%	0%	0%
Class 12	36.66%	5.58%	68.75%	16.75%	0%	0%
Class 13	14.44%	2.24%	16.03%	3.62%	36.53%	3.28%
Class 14	83.33%	7.41%	94.44%	12.59%	0%	0%
Class 15	8.94%	8.25%	21.05%	13.59%	3.42%	2.42%
Class 16	0%	0%	0%	0%	70%	10.61%

Table B.6: Precision and Recall figures for User 4 using the alternate approach for all descriptors and data split into 10% training data, 90% testing data

Setting	Precision (SIFT)	Recall (SIFT)	Precision (U-SURF64)	Recall(U-SURF64)	Precision (U-SURF128)	Recall (U-SURF128)
Class 1	58.66%	16.66%	80.55%	21.96%	41.66%	11.36%
Class 2	7.14%	10%	2.28%	10%	2.2%	16%
Class 3	98.59%	30.84%	99.6%	30.61%	99.74%	28.24%
Class 4	7.58%	19.87%	13.15%	12.42%	45.88%	24.22%
Class 5	3.84	14.85%	8.12%	32.67%	9.18%	27.72%
Class 6	7.51%	22.22%	5.35%	20.83%	5.94%	15.27%
Class 7	5.58%	18.18%	6.31%	25.45%	3.14%	21.81%
Class 8	20.12%	21.08%	24.34%	19.04%	36.51%	31.29%
Class 9	54.38%	23.66%	35%	5.34%	67.18%	32.82%
Class 10	4.81%	44.18%	4.41%	55.81%	4.62%	66.27%
Class 11	4.03%	16.12%	35.89%	45.16%	50%	32.25%
Class 12	70%	26.58%	0%	0%	66.66%	10.12%
Class 13	57.54%	36.68%	39.11%	25.72%	70.21%	22.14%
Class 14	100%	19.04%	84.61%	20.95%	90%	17.14%
Class 15	29.95%	51.44%	43.01%	57.97%	46.07%	63.76%
Class 16	68%	33.01%	53.84%	13.59%	10.81%	3.88%

Table B.7: Precision and Recall figures for User 4 using the alternate approach for all descriptors and data split into 30% training data, 70% testing data

APPENDIX C

User Questionnaire

The My Places Image Browser

Thank you for agreeing to test ”My Places”. Please read this guide carefully as it will provide you with some background information about the system, as well as explaining the tasks you are expected to complete.

1. Background

   The system is a web-based application designed to assist a user in browsing a collection of lifelog images. In this case, the images have all been captured using the Microsoft SenseCam.

   On opening the application, a single screen is presented with a weeks images displayed. This is your main photo-collection page and is the only page in the system. There are two main sections on this page. The calendar, in the top right, allows you to select which days images you wish to view. Images from the selected date are displayed on the left hand side of the screen, below the calendar, with subsequent days images being presented in the additional columns, from left to right. Days for which there are no images currently available are greyed out.

   The main focus of the system is on the displayed images. A weeks images are presented in column format, from left to right, with the leftmost column of images displaying the currently selected date. Images displayed here are keyframes selected from events which

have been detected using an offline process involving an analysis of MPEG-7 features and SenseCam metadata.

Images which are "clear" when the system is first loaded (and remain clear during normal use) represent those for which additional links are available. By clicking one of these images, a red border is displayed around the image, and other linked images are also highlighted by a red border. These are images for which a strong match occurs. Other, weak links, may be highlighted with a yellow box around the image.

The links are based on an analysis of particular locations, or settings, detected in the SenseCam images. Therefore, images taken at similar locations, should be linked together. The idea is that by providing these links, it allows a user to very quickly determine when and where they spent time in certain locations throughout a large collection of images.

Finally, certain images are displayed in different sizes. The size is related to the perceived "importance" of the event in question. The smallest images relate to events deemed to be of least importance and larger images to those deemed to of most importance.

2. Description of Tasks

   You will be presented with five short tasks to complete. Each task involves searching for particular images in your collection. It is anticipated that the total time necessary will be 10 minutes. Using the interface, all relevant images found should be marked by selecting the grey dot in the top right corner of each image. Once selected, the dot will turn orange. Images can be selected/unselected by clicking on the dot at any stage. In finding relevant images, make full use of the "find similar" features - **simply mouse-over or click on an image and the system will automatically highlight all other images that are similar to it, helping you to find the relevant images more quickly**.

   In addition, comments and suggested improvements to the user interface would be welcomed. Finally, we would like you to complete a short questionnaire after the tasks have been completed. This should take no more than 5 minutes of your time.

3. Tasks

   For each of the five tasks below, you should browse through your collection of images to locate the relevant images. Once you have located an image, please click once on it (to

record finding it). Please also list all relevant events and provide any additional information requested in the space provided below each question.

Task1. Mark the events where you were chatting with your colleague(s). For each finding, (i) write down the place where it took place (e.g. 'corridor in Computing building', 'at my desk') and (ii) rate how important that event was (1-5; 1 least important, 5 most important)

——-(Re-load the web browser before starting the next task (F5))——-

Task 2. Mark the events where you were in a vehicle (bus, car, train, plane, etc.). For each event, write down the place you were travelling to.

——-(Re-load the web browser before starting the next task (F5))——-

Task 3. Find and mark all scenes where you were eating.

——-(Re-load the web browser before starting the next task (F5))——-

Task 4. Find and mark any interesting or important events that happened in the evening after work. For each finding, write down (i) what occasion it was, and (ii) how interesting/important it was (1-5; 1 least important, 5 most important).

——-(Re-load the web browser before starting the next task (F5))——-

Task 5. Find and mark all images that are of poor quality. Write down if any of them findings depicted anything important or valuable to you.

All tasks are now complete. Fill in the questionnaire starting from the next page - please provide as much detail as possible.

Questions after the session:

How useful is the system overall?

NOT AT ALL ” ” ” ” ” ” ” VERY MUCH

Regarding the image linking

u How useful is it?

NOT AT ALL ” ” ” ” ” ” ” VERY MUCH

u Did it help you to find the information requested? ” Yes ” No

Why?

u How well is the information presented?

NOT AT ALL ” ” ” ” ” ” ” VERY MUCH

u What do you suggest be improved?

Regarding the sizing of the images (relating to event importance)

u How useful is it?

NOT AT ALL ” ” ” ” ” ” ” VERY MUCH

u Does it assist you in finding the information requested? ” Yes ” No

Why?

u What do you suggest be improved regarding this feature?

Regarding the user interface

u Overall, how easy is it to use the system?

NOT AT ALL ” ” ” ” ” ” ” VERY MUCH

u Was it easy to learn how to use the system?

NOT AT ALL ” ” ” ” ” ” ” VERY MUCH

u Is it easy to find the information you needed?

NOT AT ALL ” ” ” ” ” ” ” VERY MUCH

u Was the information provided about the system easy to understand?

NOT AT ALL ” ” ” ” ” ” ” VERY MUCH

u Was the information effective in helping you complete the tasks?

NOT AT ALL ” ” ” ” ” ” ” VERY MUCH

u Is the organisation of the information on the screen clear?

NOT AT ALL ” ” ” ” ” ” ” VERY MUCH

u Is the interface of the system pleasant?

NOT AT ALL ” ” ” ” ” ” ” VERY MUCH

u Did you enjoy using this interface?

NOT AT ALL ” ” ” ” ” ” ” VERY MUCH

u What do you suggest be improved regarding the interface?

u What are the things that you like most about this system?

1.

2.

3.

u What are the things that you like least about this system?

1.

2.

3.

u Are there any new features you would like to see in this system?

u Any other comments on the system?

Thank you very much !

Why?

u What do you suggest be improved regarding this feature?

Regarding the user interface

u Overall, how easy is it to use the system?

NOT AT ALL ” ” ” ” ” ” ” VERY MUCH

u Was it easy to learn how to use the system?

NOT AT ALL ” ” ” ” ” ” ” VERY MUCH

u Is it easy to find the information you needed?

NOT AT ALL ” ” ” ” ” ” ” VERY MUCH

u Was the information provided about the system easy to understand?

NOT AT ALL ” ” ” ” ” ” ” VERY MUCH

u Was the information effective in helping you complete the tasks?

NOT AT ALL ” ” ” ” ” ” ” VERY MUCH

u Is the organisation of the information on the screen clear?

NOT AT ALL ” ” ” ” ” ” ” VERY MUCH

u Is the interface of the system pleasant?

NOT AT ALL ” ” ” ” ” ” ” VERY MUCH

u Did you enjoy using this interface?

NOT AT ALL ” ” ” ” ” ” VERY MUCH

u What do you suggest be improved regarding the interface?

u What are the things that you like most about this system?

1.

2.

3.

u What are the things that you like least about this system?

1.

2.

3.

u Are there any new features you would like to see in this system?

u Any other comments on the system?

Thank you very much !

Why?

APPENDIX D

Applications of Interest Point Detectors

In this section, we present some related work which also utilises interest point detectors. The applications presented do not incorporate the setting detection algorithms described in this thesis. However, they do illustrate the potential usefulness of the SIFT descriptors. For this reason, we describe them briefly in the sections that follow.

D.0.3 Mo Mhúsaem Fíorúil (My Virtual Museum)

The traditional museum visitor experience has been characterised by having to choose between a limited number of predefined guided tours and the challenge of visiting on one's own. Despite the stimulating environment created in museums, they often fall short of supporting their visitors, either before, during, or after the visit, in terms of analysing and learning about what's been seen and found to be of interest. One possibility for making exhibitions more attractive to the visitor is to improve the interaction between the visitor and the objects of interest by means of supplementary information either during or after the visit. For example, a visitor, wearing a passive image capture device, generates images of various artifacts whilst wandering around the museum. The device could be supplied by the museum and retrieved at the end of each visit, thereby ensuring control over the image collections generated. The visitor can subsequently access their personalised museum tour via the museum's web site, using a unique username supplied by the museum. Once they get home, they can log on to the museum website and relive their museum experience by browsing their photos and automatically recommended supplementary material, chosen based on their known interactions. Given that the system can determine which particular artifacts the user visited, additional information (e.g. sketches, 3D models, explanatory text, professional photos, etc) about a particular object could be provided to the user, as well as images other visitors

have captured of the same artifact. This has the ancillary benefit of increasing usage of museum web-resources and providing web access to museum catalogues, but not at the expense of deterring visitors – a key concern for museums when considering web-based services.

Mo Mhúsaem Fíorúil (My Virtual Museum in the Irish language) is a web-based museum artifact search service where the users of the service, after visiting a museum and taking a number of photos of artifacts, can upload their photos to a website and find information about the artifacts those photos had captured [223]. On its web interface (see Figure D.1), a user's uploaded photos are displayed with the groupings of photos automatically formed based on the unique artifacts among the photos, and the user can drag and drop the photos into different groupings if wished. Once a particular grouping that features a unique museum artifact is selected, the system presents a list of museum artifacts that matches the user's photos, and selecting one of these will present full information about the artifact. Another way to view the interaction paradigm of this service is that the museum visitor can use their photos as query images to the service, and the retrieval result shows full information about the artifacts those photos contain.

Two passive capture devices were used to acquire the images used in this system: the Microsoft SenseCam and a Nokia N95 running the Campaignr software [8]. Campaignr is a software framework for mobile phones which enables owners of smartphones (specifically Symbian Series 60 3rd edition phones) to participate in data gathering campaigns. Should users wish to manually capture an image, they can do so using the SenseCam, by simply pressing a button on the side of the camera, or by using the N95 in the traditional manner in which camera phones operate. In this initial prototype, artificial artifacts have been used with images captured in a lab environment. The artifacts are limited in size to $30 \times 20 \times 30$ cm, due to the constraints imposed by our object model capture system. The descriptions of the recognised artifacts are fictional and are intended to simulate the workings of a real system. Once the user has selected an artifact of interest, the system will also show the pre-captured model of the artifact, that the user can rotate 360°. Images that other users have taken of the same object and which may also be of interest are also displayed. This system is freely accessible online for demonstration purposes (http://www.eeng.dcu.ie/˜vmpg/ksDemo/ks.html).

In order to demonstrate the artifact matching capabilities of our system, we created a database with artifical museum objects. The database contains images of 10 different objects, taken from multiple viewpoints with lighting, rotation, and scale changes. A sample image of each of the 10 chosen objects is shown in Figure D.2.

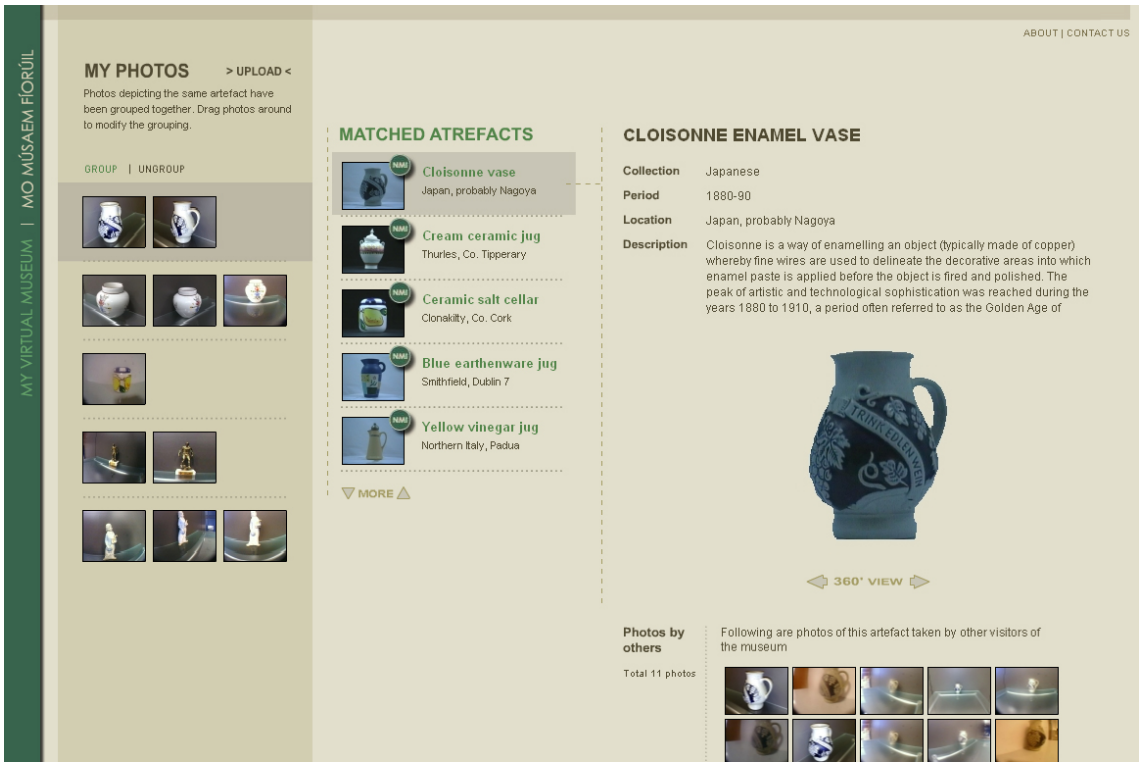


Figure D.1: Museum Information System

D.0.3.1 Object Matching System

Model images are generated using a static camera rig and an automated turntable. The turntable is situated in a light tent with diffuse ambient lighting and a controlled known-colour background. Each artifact is placed on the table and captured as it is rotated. The object is then segmented from the background using a straightforward chroma-keying process.

In order to perform matching, we utilise an approach similar to that outlined by [83]. This approach uses the SIFT features, discussed in Section 4.2.2. In order to perform object matching, the SIFT features are computed from the input image. Each keypoint is then independently matched to the database of keypoints extracted from the training images. This feature matching is done through a Euclidean distance-based nearest neighbour approach. Many of these initial matches will be incorrect due to ambiguous features or features that arise from background clutter. To increase robustness, matches are rejected for those keypoints for which the ratio of the nearest neighbour distance to the second nearest neighbour distance is greater than 0.8. This discards many of the false matches arising from background clutter. Finally, to avoid the expensive search required for finding the Euclidean distance-based nearest neighbour, an approximate algorithm, called the Best-Bin-First (BBF) algorithm is used [235]. This is a fast method for returning

Figure D.2: Sample images of the 10 artificial artifacts

the nearest neighbour with high probability. For a database of 100,000 keypoints, this provides a speedup over exact nearest neighbour search by about 2 orders of magnitude, yet results in less than a 5% loss in the number of correct matches.

Although the distance ratio test described above discards many of the false matches arising from background clutter, we can still have matches that belong to different objects. Therefore, to increase robustness to object identification, we want to cluster those features that belong to the same object and reject the matches that are left out in the clustering process. This is done using the Hough Transform [236]. Each keypoint specifies 4 parameters: 2D location, scale, and orientation. Using these parameters we use the Hough Transform to identify clusters of features that vote for the same object pose. The probability of the interpretation being correct is much higher than for any single feature. Each keypoint votes for the set of object poses that are consistent with the keypoint's location, scale, and orientation. Bins that accumulate at least 3 votes are identified as candidate object/pose matches [83]. Therefore, clusters of at least 3 features are first identified that agree on an object and its pose, as these clusters have a much higher probability of being correct than individual feature matches. Then, each cluster is checked by performing a detailed geometric fit to the model, and the result is used to accept or reject the interpretation.

For each candidate cluster, a least-squares solution for the best estimated affine projection parameters, relating the training image to the input image, is obtained. If the projection of a keypoint through these parameters lies within half the error range that was used for the parameters in the Hough transform bins, the keypoint match is kept. If fewer than 3 points remain after discarding outliers for a bin, then the object match is rejected. The least-squares fitting is repeated until no more rejections take place.

D.0.3.2 Initial Results

A number of experiments were carried out on different combinations of test and model images. We created 3 sets of model images. The reasons for the choice of three different model sets were the use of two different cameras and in order to determine if the effort required to segment the artifacts from the background using the static camera rig was justified. The first set of model images, labeled *m1*, were captured using the static camera rig. This created images of size 320×240, taken from twelve different viewing angles, for each of the 10 artifacts in our database. This allows for a greater degree of view-point independence. Due to the fact that our training images were all taken from different viewing angles in front of the artifact, we only use 5 of these images in this model set (although the 12 images are used to rotate the artifact on the user interface) (see Figure D.3). This gave a total of 50 model images.

Figure D.3: Example of the 5 model images for one of the 10 artifacts

The second set of model images, labeled *m2*, contained 3 SenseCam images for each artifact in the database, taken from 3 different viewing angles in front of the artifact in question. This gave a total of 30 model images. The final model collection, *m3*, consisted of 10 images (1 for each artifact) taken with the higher resolution Nokia N95 camera. Sample images from *m2* and *m3* can be see in Figure D.4.

Figure D.4: Example of SenseCam (1^{st} row) & N95 (2^{nd} row) model images

We used two different test sets, one for each of the cameras used. 100 images of size 640×480 were taken with the Microsoft SenseCam and 100 images of size 2592×1944 with the Nokia N95. Each set contains multiple images of all objects with differing scale, rotation, viewpoint, and lighting conditions. Images were captured by simulating a museum visitors inspection of the

artifacts. The objects used are made of different materials, have different shapes, and include ceramic vases, statues and jugs, metal and stone items, and a teddy bear. Some of the objects were placed on a glass table which produced interfering reflections. Each test image set was evaluated on each model set, giving a total of 6 different sets of experimental results. We used the confusion matrix in order to evaluate our results (shown in Tables D.1-D.6).

The results varied considerably across each combination of test and model sets of images. The *Footballer* proved challenging across all experiments. The highest recognition rate achieved for this artifact was only 40% using SenseCam test images and the *m2* set of model images. Other objects, such as the *Statue*, could not be detected at all using SenseCam and the *m1* set of model images, but achieved recognition rates of 80% using SenseCam test images and model images *m2*. Recognition rates of 100% were obtained for the *Striped Vase* and *Vinegar* using N95 test images and the *m3* set of model images. In general terms, the worst performing results were those obtained using the set of images captured using the static camera rig (*m1*) for both cameras. The best sets of results were obtained when both the test and model images were taken with the same cameras. However, impressive results can also be seen using test and model images from different cameras.

The poor results obtained using the segmented model images, *m1*, was surprising, as this is an approach often taken in the object recognition literature. However, in many of the test images, the artifacts were extremely small in size meaning that the image contained a lot of background information (see Figure D.5). In many of these cases, the algorithm found more matches on the background objects, leading to a matching failure (see Figure D.6). Initial results would therefore suggest that the effort required to remove the background from the images, using the static camera rig, is not justified.

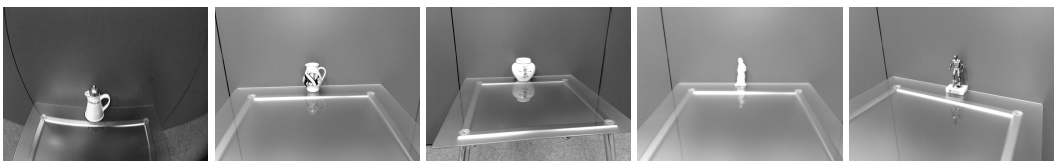


Figure D.5: Sample test images showing how many of the images used in the experiments contained a very significant amount of background information

The importance of including the background as part of the model image can be seen in the improvement in results using the remaining sets of model images. Certain artifacts were successfully matched despite variations in lighting, scale, rotation, and viewpoint. However, the recognition

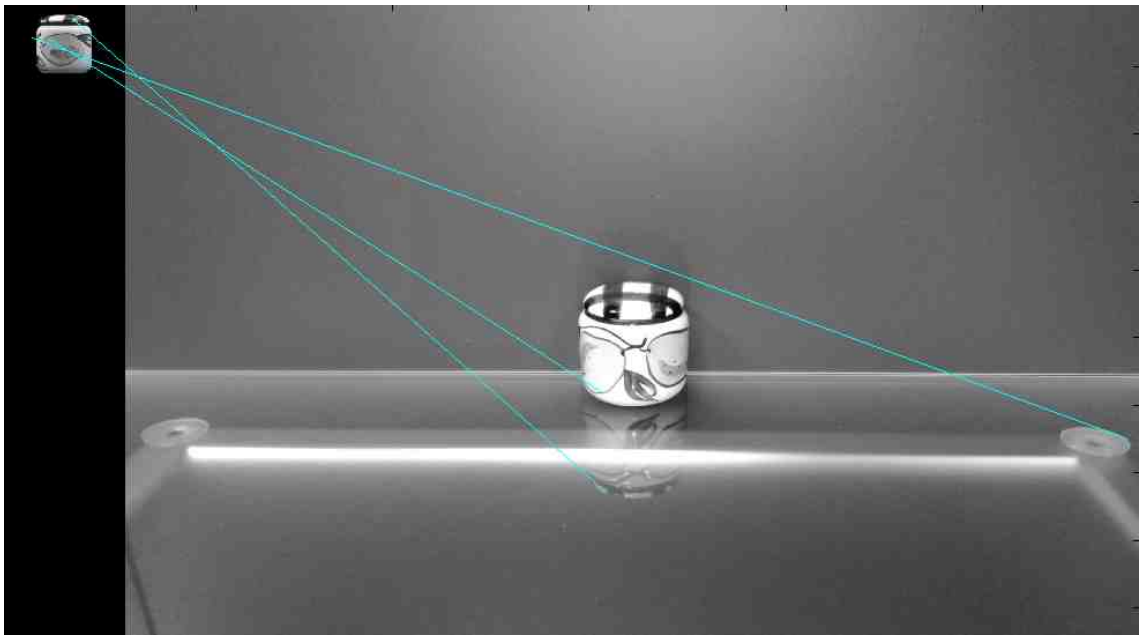


Figure D.6: Example of background matching problems. Due to the small size of the artifact in the test image, the algorithm found more matches on the background than on the artifact itself. This created problems for the experiments ran using model images where the background had been removed.

performance for others was quite low. This was again due to the background, however, it was caused by deficiencies in our experimental setup. Certain artifacts were taken in exactly the same location (i.e. we placed one object on the surface, captured images of it, and then replaced it with the next artifact). This meant that the background information in certain groups of artifacts was the same. In situations where the artifact did not provide enough robust or discriminant features, the background information was used to match the image. In many cases, the background was matched to the same background object but the image contained a different artifact captured in the same location. In a realistic museum setting, this problem should not occur, and future work will revolve around attempting to resolve these issues. In addition, as we extend to more museum artifacts, the matching accuracy and speed of the system will decrease as many more similar artifacts are added. More background clutter could also lead to more false detections. We plan to explore the use of location based methods in order to assist us in reducing the search space necessary to match in a database of many more musuem artifacts (see Figure D.7 for an initial version of this).

D.0.4 Tourist Information System

A similar system is also being developed to provide information to tourists. The tourist information system is a similar web-based search service where the users of the service, after vis-

True classes	Teddy	Cellar	Floral vase	Blue jug	Footballer	Navy Jug	Plaque	White Statue	Striped vase	Vinegar
Teddy	**40**	10	0	10	0	20	10	0	10	0
Cellar	0	**20**	0	0	20	10	20	0	0	20
Floral vase	10	10	**60**	0	0	10	0	10	0	0
Blue jug	0	0	10	**20**	10	30	10	10	0	0
Footballer	20	10	0	0	**20**	0	10	10	0	10
Navy jug	0	0	0	0	0	**80**	10	0	10	0
Plaque	20	0	10	0	20	0	**30**	0	10	10
White statue	10	0	20	0	10	60	0	**0**	0	0
Striped vase	0	10	10	10	10	10	0	0	**50**	0
Vinegar	0	0	0	0	0	70	0	0	20	**95**

Table D.1: Confusion matrix for SenseCam test images and 3D model images

True classes	Teddy	Cellar	Floral vase	Blue jug	Footballer	Navy Jug	Plaque	White Statue	Striped vase	Vinegar
Teddy	**40**	40	0	10	10	0	0	0	0	0
Cellar	0	**60**	0	10	0	20	10	0	0	0
Floral vase	0	10	**70**	10	10	0	0	0	0	0
Blue jug	0	10	10	**60**	0	0	10	10	0	0
Footballer	0	0	30	0	**10**	20	10	0	0	10
Navy jug	0	0	0	11	0	**78**	0	0	11	0
Plaque	20	0	30	0	0	10	**30**	0	0	10
White statue	0	10	20	0	0	0	0	**30**	0	10
Striped vase	0	0	10	0	0	0	0	0	**90**	0
Vinegar	0	8	0	5	0	0	0	0	0	**92**

Table D.2: Confusion matrix for N95 test images and 3D model images

True classes	Teddy	Cellar	Floral vase	Blue jug	Footballer	Navy Jug	Plaque	White Statue	Striped vase	Vinegar
Teddy	**80**	0	0	0	0	0	20	0	0	0
Cellar	20	**60**	0	0	0	0	20	0	0	0
Floral vase	30	0	**40**	10	0	0	10	0	0	10
Blue jug	30	0	0	**50**	0	0	10	10	0	0
Footballer	60	0	0	0	**40**	0	0	0	0	0
Navy jug	10	0	0	20	0	**50**	20	0	0	0
Plaque	0	0	0	0	0	0	**90**	0	10	0
White statue	0	0	0	10	0	0	10	**80**	0	0
Striped vase	20	0	0	0	10	0	10	0	**60**	0
Vinegar	40	0	0	0	0	0	0	0	40	**92**

Table D.3: Confusion matrix for SenseCam test and model images

True classes	Teddy	Cellar	Floral vase	Blue jug	Footballer	Navy Jug	Plaque	White Statue	Striped vase	Vinegar
Teddy	**80**	0	10	0	0	0	10	0	0	0
Cellar	20	**50**	10	0	0	0	0	10	20	0
Floral vase	40	0	**10**	0	0	0	0	10	30	10
Blue jug	20	0	0	**30**	0	0	0	10	20	20
Footballer	0	20	10	0	**10**	0	20	10	30	0
Navy jug	11	11	0	11	0	**23**	0	11	33	0
Plaque	10	0	0	0	0	0	**90**	0	0	0
White statue	20	20	0	0	0	0	0	**40**	20	0
Striped vase	10	0	10	0	0	10	0	0	**70**	0
Vinegar	0	8	8	0	0	0	0	0	40	**84**

Table D.4: Confusion Matrix for N95 test and SenseCam model images

True classes	Teddy	Cellar	Floral vase	Blue jug	Footballer	Navy Jug	Plaque	White Statue	Striped vase	Vinegar
Teddy	**80**	0	0	0	0	0	20	0	0	0
Cellar	0	**70**	0	0	0	0	20	0	0	10
Floral vase	20	10	**50**	0	0	0	20	0	0	0
Blue jug	10	10	10	**40**	10	0	10	0	0	10
Footballer	20	10	10	0	**30**	0	20	0	0	10
Navy jug	11	0	0	0	11	**78**	0	0	0	0
Plaque	10	0	0	0	10	0	**80**	0	0	0
White statue	60	10	10	0	0	0	10	**10**	0	0
Striped vase	0	0	0	0	0	0	0	0	**100**	0
Vinegar	0	0	0	5	0	0	0	0	0	**100**

Table D.5: Confusion matrix for N95 test and model images

Figure D.7: Museum Information System Version 2. This systems integrates location based information in order to attempt to narrow the search space in a very large collection of museum artifacts. Artifacts are displayed in the order the visitor viewed them and their path through the museum space is highlighted on a map.

iting a city on a weekend break (or any other similar activity), and taking a number of photos during their trip, can upload their photos to a website and find information about the buildings, monuments, etc. that those photos captured. On its web interface (see Figure D.8), a user's uploaded photos are displayed with the groupings of photos automatically formed based on the unique buildings or monuments among the photos, and the user can drag and drop the photos into different groupings if wished. Once a particular grouping that features a unique object is selected, the system displays it's location on the map, along with additional information gathered from other sources about the buildings or monument in question. The technology used to match different buildings and monuments in the images is the same as that used in the museum system described in Section D.0.3.1. A demo version of the system is accessible online (http://elm.eeng.dcu.ie/vmpg/mapDemo/mapTest.html).

Figure D.8: Tourist Information System

True classes	Teddy	Cellar	Floral vase	Blue jug	Footballer	Navy Jug	Plaque	White Statue	Striped vase	Vinegar
Teddy	**60**	0	0	10	10	0	10	10	0	0
Cellar	30	**60**	0	0	0	0	20	0	10	0
Floral vase	20	10	**0**	20	10	10	20	0	10	0
Blue jug	10	0	20	**30**	0	0	20	0	10	0
Footballer	10	20	0	0	**30**	20	0	0	0	10
Navy jug	0	10	0	0	0	**90**	0	0	0	0
Plaque	10	10	0	0	0	0	**70**	0	10	0
White statue	30	10	0	0	0	0	20	**30**	0	10
Striped vase	0	0	10	0	0	0	0	0	**90**	0
Vinegar	10	10	20	0	0	0	10	0	20	**30**

Table D.6: Confusion matrix for SenseCam test and N95 model images

APPENDIX E

Detailed Description of SIFT and SURF

E.1 Scale Invariant Feature Transform

The following subsections provide more detail on the steps involved in the generation of SIFT keypoints.

E.1.1 Scale-space Extrema Detection

Interesting image features or key points are detected using a cascade filtering approach that identifies image candidate locations that will be further evaluated later. The first step is to determine image location coordinates and scales that can be repeatably assigned under pose variation of the object of interest. Finding locations that are invariant to scale is performed by a scale function that searches for stable features across different scales. The scale-space convolution kernel of choice is the Gaussian function used to define the scale-space function of an input image according to:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{E.1}$$

where $I(x, y)$ is the grey scale image, and $*$ is the convolution operation in x and y with Gaussian:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{(-x^2+y^2)/2\sigma^2} \tag{E.2}$$

To detect stable keypoint locations in scale space, the Difference-of-Gaussian (DoG) function convolved with the image $D(x, y, \sigma)$ is computed from the difference of two nearby scales

separated by a constant multiplicative factor k as in:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \tag{E.3}$$

The DoG function is a close approximation to the scale-normalised Laplacian of Gaussian $\sigma^2\nabla^2 G$. It is known that the maxima and minima of $\sigma^2\nabla^2 G$ produces the most stable image features compared to a range of other possible image functions, such as the Harris corner function.

An approach to the construction of $D(x, y, \sigma)$ is shown in Figure E.1. The input image is incrementally convolved with Gaussians using $\sigma = \sqrt{2}$ to produce images shown stacked in the left column. That is, the bottom image is first convolved with Gaussian using $\sigma = \sqrt{2}$, and then repeated with a further incremental smoothing of $\sigma = \sqrt{2}$ to give the second image from the bottom, which now has an effective smoothing of $\sigma = 2$. The bottom DoG function is obtained by subtracting the second image from the bottom image, resulting in a ratio of $2/\sqrt{2} = \sqrt{2}$ between the two Gaussians. We repeat these procedures until we generate $s + 3$ images in the stack of Gaussian images and thus $s + 2$ images in the stack of DoG images on each pyramid level or octave, where s is the number of intervals used. Figure E.2 shows one interval of local extrema computation which uses three levels of the DoG function.

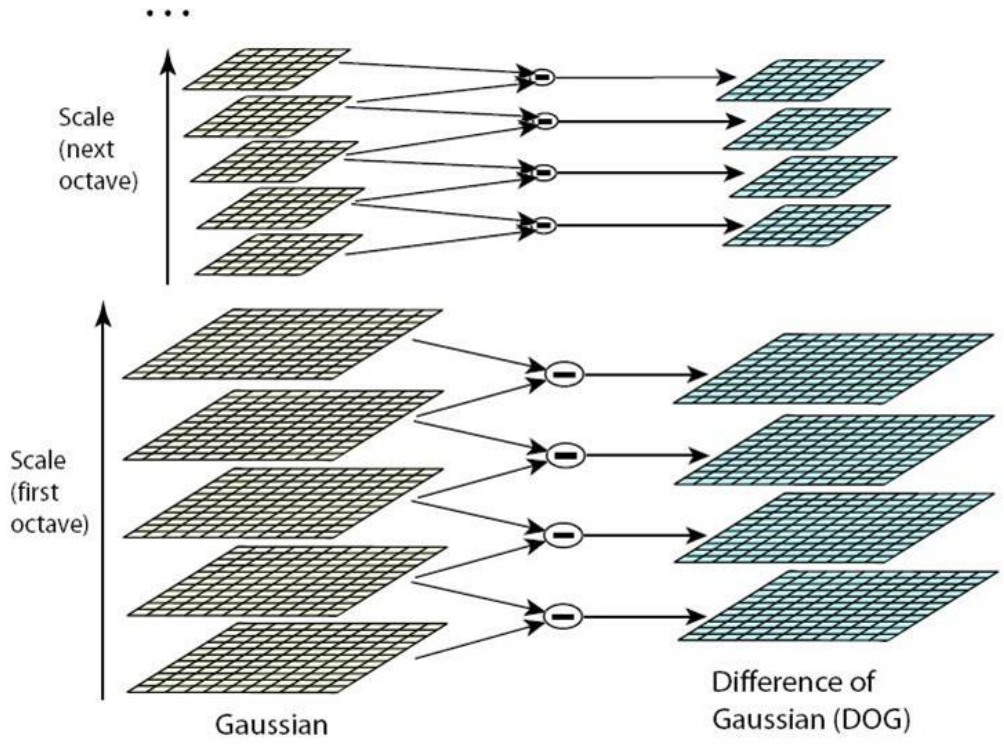


Figure E.1: Gaussian and DoG Pyramids [83]

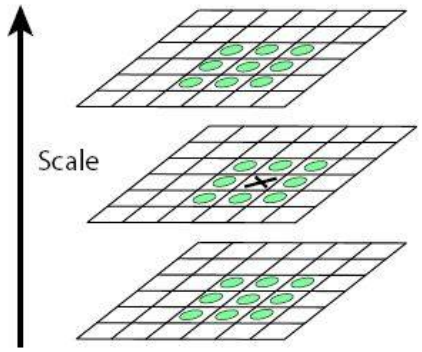


Figure E.2: One interval of local extrema detection [83]

To generate the next pyramid level, we downsample the bottom image of the current level by taking every second pixel in each row and column, which now has twice the initial value of σ (i.e. has an effective smoothing of $\sigma = 2\sqrt{2}$). Then, the same computations are repeated until we obtain the specified number of pyramid levels.

E.1.2 Keypoint Localisation

Local maxima and minima of the DoG function are detected as keypoint candidates. To detect local maxima and minima of the DoG function, $D(x, y, \sigma)$, each sample point is compared to its eight nearest neighbours in the current image and nine neighbours in the scale above and below, as in Figure E.2. Selection takes place only when a sample point is larger or smaller than all neighbours under comparison.

Once a keypoint candidate has been detected, the next step is to perform a detailed fit to local image data for location, scale, and ratio of principal curvatures. With this information, points are rejected that have low contrast because they are sensitive to noise or are poorly localised along an edge. A simple approach is to locate keypoints at the location and scale of the central sample point.

A more advanced approach is to fit a three dimensional quadratic function to the local sample points to determine the location of the maximum [148]. This approach provides improvements to matching and stability and uses a Taylor expansion (up to the quadratic terms) of the scale-space function $D(x, y, \sigma)$, shifted so the origin is located at the sample point:

$$D(x) = D + \frac{\alpha D^T}{\alpha x} x + \frac{1}{2} x^T \frac{\alpha^2 D}{\alpha x^2} x \tag{E.4}$$

where D and its derivatives are evaluated at the sample point and $x = (x, y, \sigma)^T$ is the offset from the particular point. The location of the extremum $\hat{x}$ is determined by taking the derivative of D with respect to x and setting it to zero such that:

$$\hat{x} = \frac{\alpha^2 D^{-1}}{\alpha x^2} \frac{\alpha D}{\alpha x} \tag{E.5}$$

In practice, the Hessian and derivative of D are approximated by using differences of neighbouring sample points, resulting in the solution of a 3×3 linear system. When the offset, $\hat{x}$, is larger than 0.5 in any dimension, then the implication is that the extremum lies closer to a different sample point. In this case, it is necessary to change sample points, and interpolation is then performed about the point instead. The final offset $\hat{x}$ is added to the location of its sample point to get the interpolated estimate for the location of the extremum. By taking the function value at the extremum, $D(\hat{x})$, it is possible to reject unstable extrema with low contrast. Weak keypoints are removed if $|D(\hat{x})| < 0.03$ [148]

However, using low contrast rejecting criteria alone is not sufficient because the DoG function will have a strong response at edges even when the location along the edge is poorly determined and therefore sensitive to small amounts of noise. These can be eliminated by computing the 2×2 Hessian matrix. A poorly defined peak in the Difference-of-Gaussian function will have a large principal curvature across the edge but a small curvature in the perpendicular direction. Principal curvatures are computed from the Hessian matrix:

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{E.6}$$

where the derivatives are found by taking differences of neighbouring points. The eigenvalues of H are proportional to the principal curvatures of D. It is not necessary to compute the eigenvalues explicitly as only the ratio is interesting. Letting α be the eigenvalue with largest magnitude and β the smaller, then the sum of eigenvalues is the trace of H:

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta \tag{E.7}$$

and the product is the determinant:

$$Det(H) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta \tag{E.8}$$

When the determinant is negative, then the curvatures have different signs so the point is discarded as a candidate extremum. Letting r be the ratio between the largest magnitude eigenvalue and the smaller one such that $\alpha = r\beta$, then:

$$\frac{Tr(H)^2}{Det(H)} = \frac{(r+1)^2}{r} \tag{E.9}$$

depending only on the ratio of the eigenvalues instead of their individual values. When the two eigenvalues are equal, the quantity $(r+1)^2/r$ is at a minimum and will increase with r. As a result, to make sure the ratio of principal curvatures is below some threshold, r, it is necessary to check:

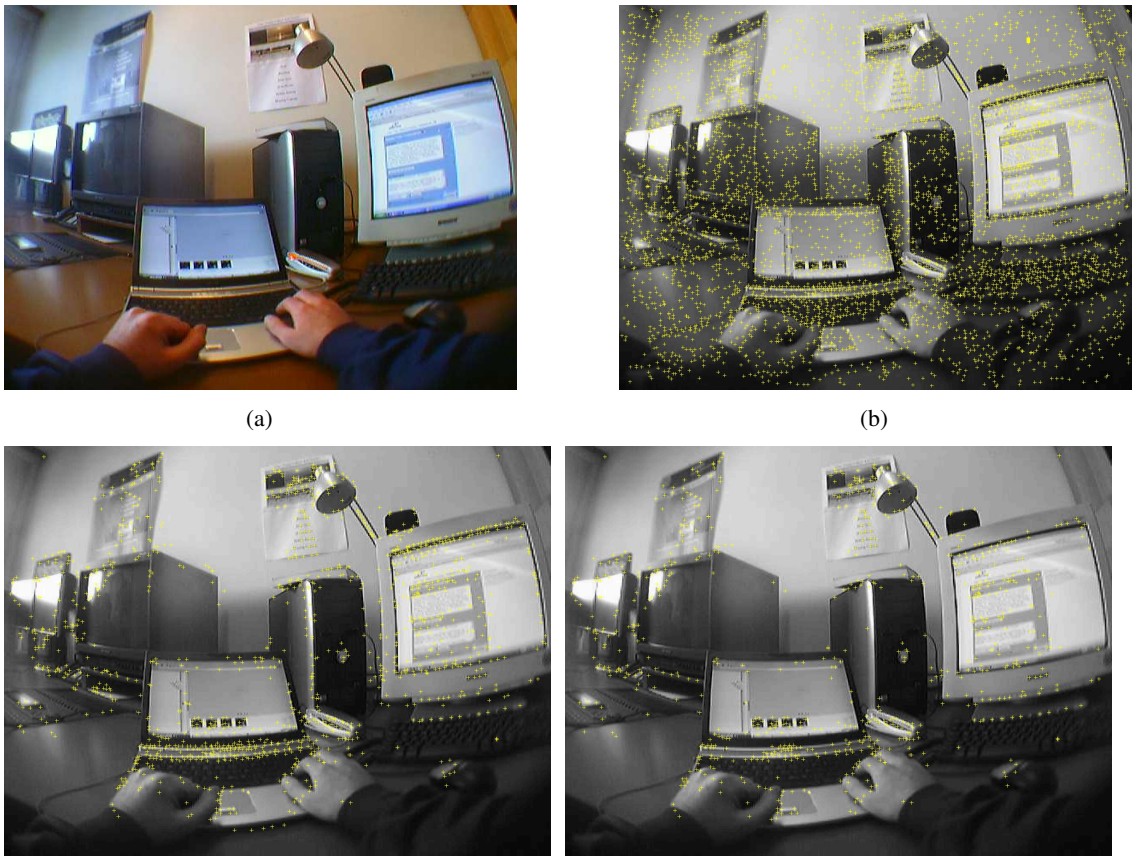
$$\frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r}, \tag{E.10}$$

For $r = 3$ [83], the keypoint is also pruned. Figure E.3 shows the stages of keypoint selection. Figure E.3(a) shows the original SenseCam image. Figure E.3(b) shows the 896 keypoints detected at maxima and minima of the Difference-of-Gaussian function. Figure E.3(c) shows the remaining keypoints following the removal of those with a value of $|D(\hat{x})| < 0.03$. Figure E.3(d) shows the final keypoints remaining after eliminating edge responses.

E.1.3 Orientation Assignment

Consistent orientations based on local image properties are assigned to each keypoint. Representing a keypoint descriptor relative to the orientation assignment is motivated by the desire to achieve invariance to image rotation.

The scale of the keypoint is used to select the Gaussian smoothed image, L, with the closest scale, so that all computations are performed in a scale invariant manner. For each image sample, $L(x, y)$, at a particular scale, the gradient magnitude, $m(x, y)$, and orientation, $\theta(x, y)$, is precomputed:



(a) (b)

(c) (d)

Figure E.3: This figure shows the stages of keypoint selection: (a) The original SenseCam image; (b) The initial 896 keypoints at maxima and minima of the Difference-of-Gaussian function; (c) The 729 remaining keypoints after applying a threshold on minimum contrast; (d) The final 526 keypoints remaining after an additional threshold on the ratio of principal curvatures.

$$m(x,y) = \sqrt{L_x^2 + L_y^2} \tag{E.11}$$

$$\theta(x,y) = tan^{-1}(L_y/L_x) \tag{E.12}$$

where $L_x = L(x+1,y) - L(x-1,y)$ and $L_y = L(x,y+1) - L(x,y-1)$ are pixel differences.

A histogram is built from the gradient orientations of the neighbours of a keypoint. The histogram has 36 bins representing the 360 degree range of orientation. Each point sampled from around the keypoint is weighted by its gradient magnitude and with a Gaussian-weighted circular window with σ equal to 1.5 times the scale of the keypoint [83]. The peaks in the orientation histogram correspond to the dominant directions of local gradients. The highest peak in the histogram, and all other peaks within 80% of the highest peak, are set as the orientation of the key-

point. Therefore, for multiple peaks of similar magnitude, there will be multiple keypoints created at the same location and scale, but different orientations. To improve accuracy, a parabola is fitted to the three histogram values that are closest to each peak.

E.1.4 Keypoint Descriptor

A keypoint descriptor is created by computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location, as shown on the left of Figure E.4. These are weighted by a Gaussian window, indicated by the overlaid circle. These samples are accumulated into orientation histograms over 4×4 subregions, as shown on the right of Figure E.4. The length of each arrow corresponds to the sum of the gradient magnitudes near that direction within the region [83]. Figure E.4 is a 2×2 descriptor array computed from an 8×8 set of samples, whereas the procedure herein uses a 4×4 descriptor computed from a 16×16 sample array. Using this method, a descriptor of $4 \times 4 \times 8 = 128$ elements is obtained, 4×4 descriptors and 8 bins. Descriptors are generated using a 16×16 image patch from $I(x, y)$ and Equation E.2, or compactly written as $I * G_\alpha$, where α_i is the scale of the keypoint centered at (x_i, y_i). Next, the gradient orientation relative to the keypoint's orientation is computed followed by the orientation histogram of each 4×4 pixel block. Due to the Gaussian weighted window, pixels closer to the centre of the 16×16 patch contribute more to the orientation histograms.

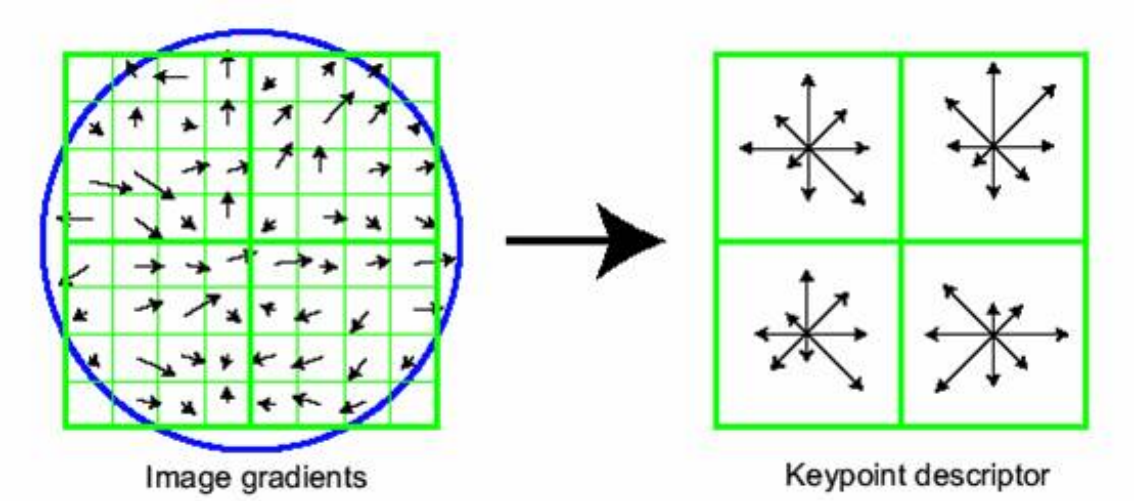


Figure E.4: Keypoint descriptor generation [83]

Some post processing is necessary in order to manage the effects of nonlinear illumination changes that affect 3D surfaces by different orientations and magnitudes. Such illumination effects cause large change in relative magnitudes but not orientations. Consequently, large gradient

magnitudes are thresholded by a factor of 0.2 and the entire feature vector is renormalised.

E.2 Speeded Up Robust Features

The following subsections provide more detail on the steps involved in the generation of SURF keypoints.

E.2.1 Interest Point Localisation

The SURF detector is based on the Hessian matrix. Given a point $x = [x, y]$ in an image I, the Hessian matrix $H(x, \sigma)$ in x at scale σ is defined as follows:

$$H = \begin{bmatrix} L_{xx}(x,\sigma) & L_{xy}(x,\sigma) \\ L_{xy}(x,\sigma) & L_{yy}(x,\sigma) \end{bmatrix} \tag{E.13}$$

where $L_{xx}(x, \sigma)$ is the convolution of the Gaussian second order derivative, $\frac{\partial^2}{\partial x^2} g(\sigma)$, with the image I in point x, and similarly for $L_{xy}(x, \sigma)$ and $L_{yy}(x, \sigma)$. In contrast to SIFT, which approximates Laplacian of Gaussian (LoG) with Difference of Gaussians (DoG), SURF approximates second order Gaussian derivatives with box filters. An example of one of these filters for the lowest scale analysed is shown in Figure E.5. Image convolutions with these box filters can be computed rapidly by using integral images [144].

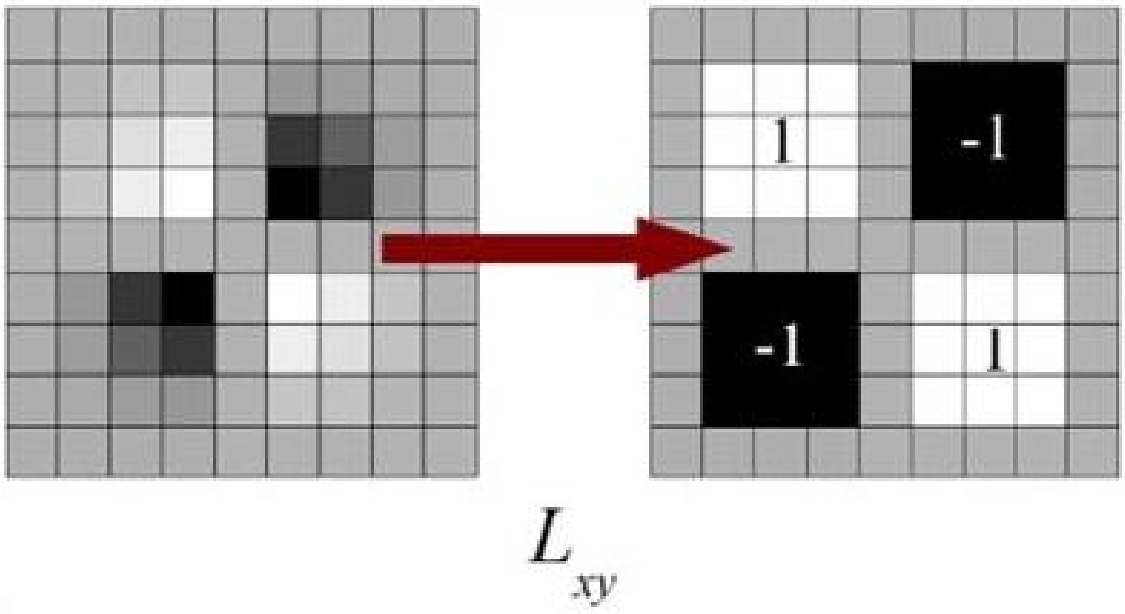


Figure E.5: Left: gaussian second order derivative in xy-direction. Right: corresponding box filter approximation [130].

The location and scale of interest points are selected by relying on the determinant of the Hessian. For each point $x = (x, y)$ of the 'image', its Hessian determinant at scale σ is approximated as follows:

$$det|H_{approx}(x, \sigma)| = D_{xx,\sigma}D_{yy,\sigma} - (c_\sigma \cdot D_{xy,\sigma}) \tag{E.14}$$

where $D_{xx,\sigma}$, $D_{yy,\sigma}$, and $D_{xy,\sigma}$ are box filter approximations for Gaussian second-order derivatives at scale σ, and c_σ is a correction constant, depending on the current scale and the size of the box filters.

The computation of the Hessian determinant is stored on a different layer for each scale. The combination of these layers is a three-dimensional image, on which is applied a non-maxima suppression in a $3 \times 3 \times 3$ neighbourhood. The maxima are then interpolated in scale and image space, and interest points are extracted from this new three-dimensional 'image' [130].

E.2.2 Interest Point Descriptor

The first step is to construct a circular region around the detected interest points in order to assign a unique orientation to the former and thus gain invariance to image rotations. The orientation is computed using Haar wavelet responses in both x and y directions. The Haar wavelets can be quickly computed via integral images, similar to the Gaussian second order approximated box filters. The dominant orientation is estimated and included in the interest point information.

Following this, SURF descriptors are constructed by extracting square regions around the interest points. These are oriented in the directions assigned in the previous step. The windows are split up into 4×4 sub-regions in order to retain some spatial information. In each sub-region, Haar wavelets are extracted at regularly spaced sample points. The wavelet responses in horizontal and vertical directions (dx and dy) are summed over each sub-region. Furthermore, the absolute values, $|dx|$ and $|dy|$, are summed in order to obtain information about the polarity of the image intensity changes. Hence, the underlying intensity pattern of each sub-region is described by a vector $V = [\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|]$. The resulting descriptor vector for all 4×4 sub-regions is of length 64, giving the standard SURF descriptor, SURF-64. The Haar wavelets are invariant to illumination bias and additional invariance to contrast is achieved by normalising the descriptor vector to unit length.

An important characteristic of SURF is the fast extraction process, which takes advantage of

integral images and a fast non-maximum suppression algorithm. SURF also facilitates fast image matching, mainly achieved by a single step added to the indexing based on the sign of the Laplacian (trace of the Hessian matrix) of the interest point. The sign of the Laplacian distinguishes bright blobs on a dark background from the inverse situation. Bright interest points are only matched against other bright interest points and similarly for the dark ones. This information facilitates a significant increase in matching speed and it comes at no computational cost, as it has already been computed in the interest point detection step.

The SURF descriptor associated with a SURF keypoint is made up of 6D localisation and 64D description components [130]. The structure of this 70D descriptor is $[x, y, a, b, a, l, desc]$, where (x, y) are the x and y coordinates (subpixel) of the position of the keypoint; a represents the scale at which the keypoint is detected; b represents the corner strength of the keypoint which is detected by a Hessian matrix; l is the sign of the Laplacian $[+1, -1]$ that allows for rapid matching. The first six elements form the localisation component, while the 64D $desc$ vector forms the actual description component that is used for determining matches.

Bibliography

[1] J. Gemmell, L. Williams, K. Wood, R. Lueder, and G. Bell. Passive capture and ensuing issues for a personal lifetime store. October 2004.

[2] M. Lamming and M. Flynn. Forget-me-not: Intimate computing in support of human memory. In *FRIEND21, Int. Symp. Next Generation Human Interface*, pages 125–128, February 1994.

[3] J. Gemmell, R. Lueder, and G. Bell. Living with a lifetime store. In *ATR Workshop on Ubiquitous Experience Media*, September 2003.

[4] N. O'Hare, C. Gurrin, G. Jones, and A. Smeaton. Combination of content analysis and context features for digital photograph retrieval. *2nd IEE European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies*, November 2005.

[5] J. Gemmell, G. Bell, R. Lueder, S. Drucker, and C. Wong. Mylifebits: Fulfilling the memex vision. In *ACM Multimedia*, December 2002.

[6] S. Firth. Photographic memories: Always-on camera captures life's fleeting moments. Technical report, Hewlett Packard Research Labs, http://www.hpl.hp.com/news/2004/jan-mar/casualcapture.html, March 2004.

[7] J. Healey and R.W. Picard. Startlecam: A cyberbetic wearable camera. October 1998.

[8] A. Joki, J. Burke, and D. Estrin. Campaignr: A framework for participatory data collection on mobile phones. Technical Report 770, Centre for Embedded Network Sensing, University of California, Los Angeles, October 2007.

[9] M. Bukhin and M. DelGaudio. Waymarkr - acquiring perspective through continuous documentation. In *5th international conference on Mobile and ubiquitous multimedia*, volume 193, Stanford, California, December 2006.

[10] Vannevar Bush. *As we may think*. The Atlantic Monthly, July 1945.

[11] T. Hori and K. Aizawa. Context-based video retrieval system for the life-log applications. In *5th ACM SIGMM international workshop on Multimedia Information Retrieval*, Berkeley, California, USA, 2003.

[12] B. Clarkson, K. Mase, and A. Pentland. Recognizing user context via wearable sensors. In *Fourth International Symposium on Wearable Computers*, pages 69–76, 2000.

[13] E. Harrison. The eat22 project. http://www.ellieharrison.com.

[14] Stephanie. The all-consuming project. http://www.all-consuming.com.

[15] A. Warhol. The time capsules project. http://www.warhol.org/collections/archives.html.

[16] D. Engelbart. Authorship provisions in augment. In *IEEE Computer Conference (COMPCON)*, pages 465–472, San Francisco, USA., March 1984.

[17] T. Nelson. Xanalogical structure, needed now more than ever: Parallel documents, deep links to content, deep versioning, and deep re-use. In *ACM Computing Surveys 31*, December 1999.

[18] S. Mann and H. Niedzviecki. *Cyborg: Digital Destiny and Human Possibility in the Age of the Wearable Computer*. Random House, 2001.

[19] S. Mann. Sousveillance: Inverse surveillance in multimedia imaging. In *12th annual ACM international conference on Multimedia*, pages 620–627, October 2004.

[20] J. Gemmell, A. Aris, and R. Lueder. Telling stories with mylifebits. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 1536–1539, July 2005.

[21] D. Ellis and K. Lee. Minimal-impact audio-based personal archives. In *1st ACM workshop on Continuous Archiving and Recording of Personal Experiences (CARPE)*, New York, 2004.

[22] Nokia Corporation. Nokia lifeblog. http://www.nokia.com/lifeblog.

[23] W. Cheng, L. Golubchik, and D. Kay. Total recall: Are privacy changes inevitable? In *1st ACM workshop on Continuous Archiving and Recording of Personal Experiences (CARPE)*, New York, 2004.

[24] T. Harada and Y. Kuniyoshi. Cyber goggles: A high-tech memory aid. http://www.pinktentacle.com/2008/03/cyber-goggles-high-tech-memory-aid/.

[25] A. Fitzgibbon and E. Reiter. Memories for life: Managing information over a human lifetime. Technical report, UK Computing Research Committee Grand Challenge proposal, 2003. http://www.ukcrc.org.uk/gcresearch.pdf.

[26] S. Reich, L. Goldberg, and S. Hudek. Deja view camwear model 100. In *1st ACM workshop on Continuous Archiving and Recording of Personal Experiences (CARPE)*, New York, 2004.

[27] Bodymedia. Bodymedia devices. http://www.bodymedia.com/main.jsp.

[28] G. Steketee and R. Frost. Compulsive hoarding - current status of the research. *Clinical Psychology Review*, 23(7):905–927, December 2003.

[29] A. Frigo. Storing, indexing and retrieving my autobiography. In *Pervasive Workshop on Memory and Sharing of Experiences*, pages 52–56, Vienna, Austria, 2004.

[30] L. Kimbell. I measure therefore i am. http://www.lucykimbell.com.

[31] J. Kelleher. The daily photo project. http://www.c71123.com.

[32] S. Hodges, L. Williams, E. Berry, S. Izadi, J. Srinivasan, A. Butler, G. Smyth, N. Kapur, and K. Wood. Sensecam: A retrospective memory aid. In *Eighth International Conference on Ubiquitous Computing*, September 2006.

[33] Defence Advanced Research Projects Agency (DARPA). Lifelog: Proposer information pamphlet. Technical report, 2003.

[34] C. Schlenoff, B. Weiss, M. Steves, A. Virts, M. Shneier, and M. Linegang. Overview of the first advanced technology evaluations for ASSIST. In *Performance Metrics for Intelligent Systems Workshop (PerMIS), IEEE Safety, Security, and Rescue Robotics Conference*, August 2006.

[35] UK House of Lords. Judgments: Campbell (appellant) v. MGN limited (respondents). http://www.publications.parliament.uk/pa/ld200304/ldjudgmt/jd040506/campbe-1.htm, May 2004.

[36] D.L. Schacter. *The Seven Sins of Memory: How the Mind Forgets and Remembers*. Houghton Mifflin, 2001.

[37] H.B. Coslett. Consciousness and attention. *Seminars in Neurology*, 17(2):137–144, June 1997.

[38] E.S. Parker, L. Cahill, and J.L. McGaugh. A case of unusual autobiographical remembering. *Neurocase - case studies in neuropsychology, neuropsychiatry, and behavioural neurology*, 12(1):35–49, February 2006.

[39] L.J. Bannon. Forgetting as a feature, not a bug: the duality of memory and implications for ubiquitous computing. *CoDesign*, 2(1):3–15, 2006.

[40] M. Dodge and R. Kitchin. Outlines of a world coming into existence - pervasive computing and the ethics of forgetting. *Environment and Planning B - Planning and Design*, 34:431–445, March 2007.

[41] P. Danielson. Video surveillance for the rest of us: Proliferation, privacy, and ethics education. In *IEEE International Symposium on Technology and Society (ISTAS'02)*, pages 162–167, 2002.

[42] M. McCahill and C. Norris. CCTV in london. Technical Report 6, Centre for Criminology and Criminal Justice, University of Hull, June 2002. http://www.urbaneye.net.

[43] D. Lyon. *Surveillance as Social Sorting: Privacy, Risk and Digital Discrimination*. Routledge, London, 2003.

[44] National Roads Authority of Ireland. Intelligent transport systems (ITS) - information brochure. Technical report, National Roads Authority, 2007. http://www.nra.ie/Publications/DownloadableDocumentation/GeneralPublications/.

[45] B. Webster. Parking fines via CCTV to force drivers to obey. http://business.timesonline.co.uk/tol/business/, February 2008.

[46] D. Wood and K. Ball. A report on the surveillance society for the information commissioner, by the surveillance studies network. public discussion document. Technical report, Data Protection Commissioner, September 2006. http://www.dataprotection.ie/viewprint.asp?DocID=386&StartDate=1+January+2008.

[47] B. Welsh and D. Farrington. Crime prevention effects of closed circuit television - a systematic review. Technical Report 252, Home Office Research, Development and Statistics Directorate, August 2002.

[48] F. Nack. You must remember this. *IEEE Multimedia*, 12(1):4–7, March 2005.

[49] N. Beagrie and M. Jones. Preservation management of digital materials - a handbook. British Library, London.

[50] G. Bell and J. Gemmell. A digital life. http://www.sciam.com/article.cfm?id=a-digital-life&colID=1, March 2007.

[51] D.M. Smith. The cost of lost data. *Graziadio Business Report - Journal of Contemporary Business Practice*, 6, 2003. http://gbr.pepperdine.edu/033/dataloss.html.

[52] J. Gemmell, G. Bell, and R. Lueder. Mylifebits: A personal database for everything. *Communications of the ACM*, 49(1):88–95, 2006.

[53] M. Naaman, S. Harada, Q. Wang, H. Garcia-Molina, and A. Paepcke. Context data in geo-referenced digital photo collections. *MM'04*, pages 10–16, October 2004.

[54] Google. Google mail. http://mail.google.com, 2008.

[55] E. Freeman and D. Gelernter. Lifestreams: A storage model for personal data. *ACM SIGMOD Bulletin*, 25(1):80–86, 1996.

[56] C. Thompson. A head for detail. http://www.fastcompany.com/magazine/110/head-for-detail.html, November 2006.

[57] S. Santini. *Exploratory Image Databases: Content-Based Retrieval*. Academic Press, 2001.

[58] E. Rasmussen. Indexing images. *Annual Review of Information Science and Technology*, 32:169–196, 1997.

[59] A. Cawkell. Indexing collections of electronic images: A review. *British Library Research Review*, 15, 1993.

[60] C. Gordon. An introduction to ICONCLASS. In *International Conference in Terminology for Museums*, pages 233–244, Cambridge, 1990. Museum Documentation Association.

[61] P. Enser and C.G. McGregor. Analysis of visual information retrieval queries. Technical Report 6104, British Library Research and Development Report, 1992.

[62] J. Sunderland. Image collections. *Art Libraries*, 7(2), 1982.

[63] N. Murphy C. Gurrin and G. Jones. Mediassist: Managing personal digital photo archives. ERCIM News, July 2005. No. 62.

[64] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE trans. on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

[65] R. Datta, D. Joshi, J. Li, and J.Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Transactions on Computing Surveys*, 2008.

[66] J. Chen, C. Bouman, and J. Dalton. Similarity pyramids for browsing and organization of large image databases. In *SPIE/IS&T Conf. on Human Vision and Electronic Imaging III, San Jose, CA.*, volume 3299, pages 563–575, Jan. 26-29 1998.

[67] K. Rodden, W. Basalaj, D. Sinclair, and K. Wood. Does organisation by similarity assist image browsing? In *ACM CHI*, pages 190–197, 2001.

[68] J. Platt, M. Czerwinski, and B. Field. Phototoc: Automatic clustering for browsing personal photographs. Technical Report MSR-TR-2002-17, Microsoft Research, 2002.

[69] A. Loui and A. Savakis. Automatic image event segmentation and quality screening for albuming applications. In *IEEE International Conference on Multimedia and Expo*, New York, NY, July 2000.

[70] A. Jaimes, A. B. Benitez, S.-F. Chang, and A. C. Loui. Discovering recurrent visual semantics in consumer photographs. In *IEEE Intl. Conf. on Image Processing*, 2000.

[71] M. Boutell and J. Luo. Bayesian fusion of camera metadata cues in semantic scene classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 623–630, June 2004.

[72] M. Cooper, J. Foote, A. Girgensohn, and L. Wilcox. Temporal event clustering for digital photo collections. In *MM'03, Berkeley, California*, November 2003.

[73] J.C. Platt. Autoalbum: Clustering digital photographs using probabilistic model merging. In *IEEE Workshop on Content-Based Access of Image and Video Libraries*, 2000.

[74] B.S. Manjunath and W.Y. Ma. Texture features for browsing and retrieval of image data. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 18(8), pages 837–842, August 1996.

[75] S. Belongie, C. Carson, H. Greenspan, and J. Malik. Color and texture-based image segmentation using EM and its application to content-based image retrieval. In *IEEE International Conference on Computer Vision*, 1998.

[76] A. Pentland, R. Picard, and S. Sclaroff. Photobook: Content-based manipulation of image databases. In *SPIE Storage and Retrieval of Image and Video Databases*, volume II, pages 34–37, February 1994.

[77] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. The QBIC project: Querying images by content using color, texture and shape. In *SPIE Storage and Retrieval of Image and Video Databases*, pages 171–181, 1993.

[78] C.S. Li and V. Castelli. Deriving texture set for content based retrieval of satellite image databases. In *IEEE International Conference on Image Processing*, pages 576–579, 1997.

[79] B.S. Manjunath, J.-R Ohm, V.V Vasudevan, and A. Yamada. Color and texture descriptors. *IEEE trans. circuits and systems for video technology*, 11(6):703–715, 2001.

[80] MPEG. Mpeg-7 overview. http://www.chiariglione.org/MPEG/standards/mpeg-7/mpeg-7.htm.

[81] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–535, 1997.

[82] A. Baumberg. Reliable feature matching across widely separated views. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 774–781, 2000.

[83] D. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, volume 60(2), pages 91–110, 2004.

[84] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000.

[85] R. Nair, N. Reid, and M. Davis. Photo loi: Browsing multi-user photo collections. In Singapore, editor, *MM'05*, November 2005.

[86] S. Ahern, S. King, and M. Davis. Mmm2: Mobile media metadata for photo sharing. In Singapore, editor, *MM'05*, November 2005.

[87] A. Graham, H. Garcia-Molina, A. Paepcke, and T. Winograd. Time as essence for photo browsing through personal digital libraries. In *2nd ACM/IEEE Joint Conf. on Digital Libraries, Portland, Oregon*, pages 326–335, 2002.

[88] K. Aizawa, D. Tancharoen, S. Kawasaki, and T. Yamasaki. Efficient retrieval of life log based on context and content. *CARPE*, October 2004.

[89] J.R. Smith. *Integrated Spatial and Feature Image Systems: Retrieval, Analysis and Compression*. PhD thesis, Columbia University, 1997.

[90] Y. Rui, T.S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: A power tool for interactive content-based image retrieval. In *IEEE Transactions on Circuits and Systems for Video Technology*, volume 8(5) of *Special Issue on Segmentation, Description and Retrieval of Video Content*, pages 644–655, September 1998.

[91] S. Sclaroff, L. Taycher, and M.L. Cascia. Imagerover: A content-based image browser for the world wide web. In *IEEE Workshop on Content-Based Access of Image and Video Libraries*, 1997.

[92] A.P. Berman and L.G. Shapiro. Efficient image retrieval with multiple distance measures. In *SPIE Storage and Retrieval of Image and Video Databases*, pages 12–21, February 1997.

[93] C. Nastar, M. Mitschke, and C. Meilhac. Efficient query refinement for image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition, Santa Barbara, CA*, pages 547–552, 1998.

[94] R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. Wiley, 1973.

[95] C. Venters and R. Hartley. Query by visual example: Assessing the usability of content-based image retrieval system user interfaces. In *Second IEEE Pacific-Rim Conference on Multimedia*, Beijing, China, 2001. Springer-Verlag.

[96] M. Blighe, H. Le Borgne, and N. O'Connor. Exploiting context information to aid landmark detection in sensecam images. In *2nd International Workshop on Exploiting Context Histories in Smart Environments - Infrastructures and Design (ECHISE)*, September 2006.

[97] R. Veltkamp. Content-based image retrieval systems: A survey. Technical Report UU-CS-2000-34, Department of Computing Science, Utrecht University, October 2000.

[98] F. Juster and F. Stafford. Time, goods, and well-being. *Institute for Social Research*, pages 397–414, 1985.

[99] L. Flood. Household, market, and nonmarket activities: Procedures and codes for the 1993 time-use survey. Technical Report vol. VI., Uppsala Univ. Dept. Economics, Uppsala, Sweden, 1997.

[100] A. Campbell. *The Sense of Well-Being in America*. McGraw-Hill, New York, 1981.

[101] F. M. Andrews and S. B. Whithey. *Social Indicators of Well-Being: Americans Perceptions of Life Quality*. Plenum, New York, 1976.

[102] D. Kahneman, A. B. Krueger, D. A. Schkade, N. Schwarz, and A. Stone. A survey method for characterizing daily life experience: The day reconstruction method. *Science*, 306:1776–1780, December 2004.

[103] M. Csikszentmihalyi and R. E. Larsen. The experience sampling method. *New Directions for Methodology of Social and Behavioral Science*, 15:41–56, 1983.

[104] M.F. Beal, A.E. Lang, and A.C. Ludolph. *Neurodegenerative Diseases: Neurobiology, Pathogenesis and Therapeutics*. Cambridge Press, July 2005.

[105] The Alzheimer Society of Ireland. The Alzheimer Society of Ireland Official Website. http://www.alzheimer.ie/.

[106] World Health Organisation. Mental and neurological disorders. http://www.who.int/mediacentre/factsheets/fs265/en/.

[107] The Alzheimer Society of Ireland. Economic cost. www.alzheimer.ie/eng/content/download/485/3484/file/oasis%5Fautumn06.pdf (last visited August 2008).

[108] Alzheimer's Association. Worldwide cost. http://www.alz.org/icad/media.asp.

[109] K. Utterback. Supporting a new model of care with telehealth technology. *Telehealth Practice Report*, 9(6):3–11, 2005.

[110] B.K. Smith, J. Frost, and M. Albayrak R. Sudhakar. Improving diabetes self-management with glucometers and digital photography. *Personal and Ubiquitous Computing*, 2006.

[111] California Health Care Foundation. *Patient Self Management Tools - An Overview*. Critical Mass Consulting, June 2005. http://www.chcf.org/.

[112] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man and Cybernetics*, 34(3):334–350, August 2004.

[113] G. E. Gresham, P. W. Duncan, and W. STASON. *Post-Stroke Rehabilitation/Clinical Practice Guideline*. Aspen Publishers, Inc., 1996.

[114] D. White, K. Burdick, G. Fulk, J. Searleman, and J. Carroll. A virtual reality application for stroke patient rehabilitation. In *IEEE International Conference on Mechatronics and Automation*, pages 1081–1086, July 2005.

[115] R. Colombo, F. Pisano, S. Micera, A. Mazzone, C. Delconte, M. Chiara Carrozza, P. Dario, and Giuseppe Minuco. Upper limb rehabilitation and evaluation of stroke patients using robot-aided techniques. *IEEE 9th International Conference on Rehabilitation Robotics*, pages 515–518, July 2005.

[116] S.S. Intille, K. Larson, J. Beaudin, E. Munguia Tapia, P. Kaushik, J. Nawyn, and T.J. McLeish. The placelab: a live-in laboratory for pervasive computing research. In *Pervasive 2005 Video Program*, May 2005.

[117] L. T. Kohn, J. M. Corrigan, and M. S. Donaldson. *To Err Is Human - a Safer Health System*. National Academy Press, 2000.

[118] N. Kuwahara, F. Naya, H. Itoh Ozaku, and K. Kogure. Context-awareness in a real working environment - model for understanding nursing activities. In *Exploiting Context Histories in Smart Environments (ECHISE)*, September 2006.

[119] M. Cooper, T. Liu, and E. Rief. Video segmentation via temporal pattern classification. *IEEE Transactions on Multimedia*, 9(3):610–618, 2007.

[120] F. Schaffalitzky and A. Zisserman. Automated location matching in movies. *Computer Vision and Image Understanding*, 92:236–264, 2003.

[121] K. Toyama, R. Logan, and A. Roseway. Geographic location tags on digital images. In *11th ACM international conference on Multimedia*, pages 156–166, 2003.

[122] A. Varshavsky, M. Chen, E. de Lara, J. Froehlich, D. Haehnel, J. Hightower, A. LaMarca, F. Potter, T. Sohn, K. Tang, and I. Smith. Are gsm phones the solution for localization? In *7th IEEE Workshop on Mobile Computing Systems and Applications (HotMobile)*, 2006.

[123] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit. Place lab: Device positioning using radio beacons in the wild. In *Pervasive*, volume 3468, pages 116–133, 2005.

[124] M. Chen, T. Sohn, D. Chmelev, D. Haehnel, J. Hightower, J. Hughes, A. LaMarca, F. Potter, I. Smith, and A. Varshavsky. Practical metropolitan-scale positioning for gsm phones. In *8th International Conference on Ubiquitous Computing (UbiComp)*, pages 225–242, 2006.

[125] S. Patel, J. Kientz, G. Hayes, S. Bhat, and G. Abowd. Farther than you may think: An empirical investigation of the proximity of users to their mobile phones. In *8th International Conference on Ubiquitous Computing (UbiComp)*, pages 123–140, 2006.

[126] A. Doherty, A. Smeaton, K. Lee, and D. Ellis. Multimodal segmentation of lifelog data. In *RIAO 2007 - Large-Scale Semantic Access to Content (Text, Image, Video and Sound)*, Pittsburgh, PA, USA, 30 May - 1 June 2007 2007.

[127] A. Doherty, D. Byrne, A.F. Smeaton, G. Jones, and M. Hughes. Investigating keyframe selection methods in the novel domain of passively captured visual lifelogs. In *ACM International Conference on Image and Video Retrieval (CIVR)*, July 2008.

[128] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, February 2005.

[129] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *9th IEEE Int'l Conf. on Computer Vision*, volume 2, 2003.

[130] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *9th European Conference on Computer Vision*, May 2006.

[131] M. Boutell and C. Brown. Review of the state of the art in semantic scene classification. Technical report, Department of Computer Science, University of Rochester, Rochester, NY, 2002.

[132] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer. Weak hypotheses and boosting for generic object detection and recognition. In *European Conference on Computer Vision*, volume 2, pages 71–84, 2004.

[133] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 264–271, 2003.

[134] C. Pellegrini, H. Bosch, A. Labbi, and W. Gerstner. Viewpoint: Invariant object recognition using independent component analysis. *NOLTA 99, Hawaii, USA*, 1999.

[135] Y. Amit and M. Mascaro. An integrated network for invariant visual detection and recognition. *Vision Research*, 43(19):2073–2088, 2003.

[136] B.W. Mel. Seemore: Combining color, shape and texture histogramming in a neurally-inspired approach to visual object recognition. *Neural Computation*, 9(4):777804, 1997.

[137] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2(11):1019–1025, 1999.

[138] Gy. Dorko and C. Schmid. Selection of scale-invariant parts for object class recognition. In *International Conference on Computer Vision*, 2003.

[139] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. In *Pattern Analysis and Machine Intelligence (PAMI)*, 2002.

[140] A. Torralba, K.P. Murphy, and W.T. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[141] H. Bay, B. Fasel, and L. Van Gool. Interactive museum guide: Fast and robust recognition of museum objects. In *First International Workshop on Mobile Vision*, 2006.

[142] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *European Conference on Computer Vision*, 2000.

[143] P. Schugerl, R. Sorschag, W. Bailer, and G. Thallinger. Object re-detection using sift and mpeg-7 color descriptors. In *International Workshop on Multimedia Content Analysis and Mining*, pages 305–314. Springer, July 2007.

[144] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition (CVPR)*, 2001.

[145] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *European Conference on Computer Vision*, pages 113–130, 2002.

[146] M. Shneier. Road sign detection and recognition. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2005.

[147] D.G. Lowe. Local feature view clustering for 3d object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 682–688, 2001.

[148] D.G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision (ICCV)*, pages 1150–1157, 1999.

[149] C. Harris and M. Stephens. A combined corner and edge detector. pages 189–192, 1988.

[150] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *European Conference on Computer Vision*, May 2003.

[151] P. Fockler, T. Zeidler, B. Brombach, E. Bruns, and O. Bimber. Phoneguide: Museum guidance supported by on-device object recognition on mobile phones. Technical Report 54:74, Bauhaus-University Weimar, Weimar, Germany, 2005.

[152] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *Computer Vision and Pattern Recognition*, volume 2, pages 506–513, 2004.

[153] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classifcation of texture and object categories: An in-depth study. Technical Report RR-5737, INRIA Rhone-Alpes, November 2005.

[154] R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[155] G. Bouchard and B. Triggs. Hierarchical part-based visual object categorization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[156] G. Carneiro and D. Lowe. Sparse flexible models of local features. In *European Conference on Computer Vision*, 2006.

[157] J. Amores, N. Sebe, P. Radeva, T. Gevers, and A. Smeulders. Boosting contextual information in content-based image retrieval. In *MIR Workshop, ACM Multimedia*, 2004.

[158] J. Amores, N. Sebe, and P. Radeva. Fast spatial pattern discovery integrating boosting with constellations of contextual descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[159] H. Zhang, R. Rahmani, S.R. Cholleti, and S.A. Goldman. Local image representations using pruned salient points with applications to CBIR. In *ACM Multimedia*, 2006.

[160] A. Torralba. Contextual priming for object detection. *International Journal of Computer Vision*, 53(2):169–191, 2003.

[161] W.H. Adams, G. Iyengar, C. Lin, M.R. Naphade, C. Neti, H.J. Nock, and J.R. Smith. Semantic indexing of multimedia content using visual, audio, and text cues. In *Journal on Applied Signal Processing*, volume 2, page 116, 2003.

[162] F. Jing, M. Li, L. Zhang, H. Zhang, and B. Zhang. Learning in region based image retrieval. In *International Conference on Image and Video Retrieval*, pages 206–215, 2003.

[163] J. Wang, J. Li, and G. Wiederhold. Simplicity: semantics-sensitive integrated matching for picture libraries. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 23 of *9*, pages 947–963, 2001.

[164] M. Boutell and J. Luo. Beyond pixels: Exploiting camera metadata for photo classification. *IEEE Conference on Computer Vision and Pattern Recognition*, June 2004.

[165] A. Vailaya, M. Figueiredo, A. Jain, and H. Zhang. Image classification for content-based indexing. In *IEEE Trans. on Image Processing*, volume 10 of *1*, pages 117–130, 2001.

[166] N. OConnor, E. Cooke, H. Le Borgne, M. Blighe, and T. Adamek. The acetoolbox: Low-level audiovisual feature extraction for retrieval and classification. *2nd IEE European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies*, November 2005.

[167] N. Serrano, A. Savakis, and J. Luo. Improved scene classification using efficient low-level features and semantic cues. *Pattern Recognition*, 37:1773–1784, 2004.

[168] S. Thorpe, D. Fize, and C. Marlot. Speed of processing in the human visual system. *Nature*, 381:520–522, 1996.

[169] L. Fei-Fei, R. VanRullen, C. Koch, and P. Perona. Rapid natural scene categorisation in the near absence of attention. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 99 of *14*, pages 9596–9601, 2002.

[170] A. Treisman and G. Gelade. A feature-integration theory of atention. *Cognitive Psychology*, 12:97–136, 1980.

[171] M. Varma and A. Zisserman. Texture classification: Are filter banks necessary? In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 691–698, 2003.

[172] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using affine-invariant regions. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 319–324, 2003.

[173] J. Willamowski, D. Arregui, G. Csurka, C.R. Dance, and L. Fan. Categorizing nine visual classes using local appearance descriptors. In *ICPR Workshop on Learning for Adaptable Visual Systems.*, 2004.

[174] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *The British Machine Vision Conference*, pages 384–393, 2002.

[175] P. Quelhas, F. Monay, J.M. Odobez, D. Gatica-Perez, T. Tuytelaars, and L. VanGool. Modeling scenes with local descriptors and latent aspects. In *IEEE International Conference on Computer Vision*, 2005.

[176] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *IEEE International Conference on Computer Vision And Pattern Recognition*, 2005.

[177] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or "how do i organize my holiday snaps?". In *7th European Conference on Computer Vision*, volume 1, pages 414–431, 2002.

[178] P. Pritchett and A. Zisserman. Matching and reconstruction from widely separated views. *3D Structure from Multiple Images of Large-Scale Environments*, 1506:78–92, June 1998.

[179] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman. Discovering object categories in image collections. Technical Report MIT-CSAIL-TR-2005-012, Computer Science and Artificial Intelligence Laboratory, MIT, February 2005.

[180] J. Hays and A.A. Efros. Scene completion using millions of photographs. In *ACM Transactions on Graphics (SIGGRAPH)*, volume 26 of *3*, 2007.

[181] A. Torralba, R. Fergus, and W.T. Freeman. Tiny images. Technical Report MIT-CSAIL-TR-2007-024, MIT, 2007.

[182] M. Brown and D. G. Lowe. Recognising panoramas. In *9th International Conference on Computer Vision (ICCV)*, pages 1218–1225, 2003.

[183] H.J. Zhang, C.Y. Low, and S.W. Smoliar. Video parsing and browsing using compressed data. *Multimedia Tools Appl.*, 1:89–111, 1995.

[184] D. Zhang, W. Qi, and H. J. Zhang. A new shot boundary detection algorithm. *Lecture Notes in Computer Science*, 2195(63), 2001.

[185] U. Gargi, R. Kasturi, and S.H. Strayer. Performance characterization of video shot change detection methods. In *IEEE Transactions on Circuits and Systems for Video Technology*, volume 10 of *1*, 2000.

[186] B. Yeo and B. Liu. Rapid scene change analysis on compressed video. *IEEE Transactions on Circuits and Systems for Video Technology*, 5(6):533–544, 1995.

[187] I. Koprinska and S. Carrato. Temporal video segmentation: A survey. *Signal Processing: Image Communication*, 16:477–500, 2001.

[188] H.J. Zhang, A. Kankanhalli, and S.W. Smoliar. Automatic partitioning of full-motion video. *Multimedia Systems*, 1(1):10–28, 1993.

[189] B. Shahraray. Scene change detection and content-based sampling of video sequences. In *Digital Video Compression: Algorithms and Technologies*, volume 2419, pages 2–13, 1995.

[190] A. Hampapur, R. Jain, and T. Weymouth. Digital video segmentation. In *ACM Multimedia*, pages 357–364, 1994.

[191] R. Kasturi and R. Jain. *Computer Vision: Principles*. IEEE Computer Society Press, 1991.

[192] D. Swanberg, C.F. Shu, and R. Jain. Knowledge guided parsing and retrieval in video databases. In *Storage and Retrieval for Image and Video Databases*, pages 173–187, 1993.

[193] R. Zabih, J. Miller, and K. Mai. A feature-based algorithm for detecting and classifying scene breaks. In *ACM Multimedia*, pages 189–200, 1993.

[194] R. Lienhart. Reliable transition detection in videos: A survey and practitioners guide. *International Journal of Image and Graphics*, 1:469–486, 2001.

[195] Y. Ke, R. Sukthankar, and L. Huston. Efficient near-duplicate detection and sub-image retrieval. In *ACM Multimedia*, 2004.

[196] R.M. Haralick and L.G. Shapiro. *Computer and Robot Vision*. Addison-Wesley, Reading, Massachusetts, USA, 1992.

[197] T. Lindeberg. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention. *International Journal of Computer Vision*, 11(3):283–318, 1993.

[198] H. Moravec. Obstacle avoidance and navigation in the real world by a seeing robot rover. Technical Report CMU-RI-TR-3, Carnegie-Mellon University, Robotics Institute., 1980.

[199] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *7th European Conference on Computer Vision (ECCV 2002)*, pages 128–142, 2002.

[200] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *International Journal on Computer Vision*, 60(1):63–86, 2004.

[201] V. Rodehorst and A. Koschan. Comparison and evaluation of feature point detectors. In *Proc. of the 5th International Symposium Turkish-German Joint Geodetic Days "Geodesy and Geoinformation in the Service of our Daily Life"*, March 2006.

[202] P. Brand and R. Mohr. Accuracy in image measure. In *S. El-Hakim (Ed.), Proceedings of the SPIE Conference on Videometrics III*, volume 2350, pages 218–228, 1994.

[203] T. Tuytelaars and L. Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *11th British Machine Vision Conference*, pages 421–425, 2000.

[204] W. Freeman and E. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, September 1991.

[205] J. Koenderink and A. Van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55:367–375, 1987.

[206] L. Van Gool, T. Moons, and D. Ungureanu. Affine/photometric invariants for planar intensity patterns. In *Fourth European Conference on Computer Vision*, pages 642–651, 1996.

[207] S. Clarke and P. Willett. Estimating the recall performance of search engines. *ASLIB Proceedings*, 49(7):184–189, 1997.

[208] D. Pelleg and A. Moore. X-means - extending k-means with efficient estimation of the number of clusters. In *17th International Conference on Machine Learning*, pages 727–734, 2000.

[209] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

[210] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer, 2001.

[211] Y. Rubner, C. Tomasi, and L. Guibas. The earth mover's distance as a metric for image retrieval. In *International Journal of Computer Vision*, volume 40(2), pages 99–121, 2000.

[212] K. Grauman and T. Darrell. Efficient image matching with distributions of local invariant features. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 627–634, 2005.

[213] G. B. Dantzig. *Application of the simplex method to a transportation problem*. John Wiley and Sons, 1951.

[214] O.G. Cula and K.J. Dana. Compact representation of bidirectional texture functions. In *Computer Vision and Pattern Recognition*, volume 1, pages 1041–1047, 2001.

[215] Y. Rubner and C. Tomasi. Texture-based image retrieval without segmentation. In *International Conference on Computer Vision*, pages 1018–1024, 1999.

[216] J. Puzicha, Y. Rubner, C. Tomasi, and J. Buhmann. Empirical evaluation of dissimilarity measures for color and texture. In *International Conference on Computer Vision*, volume 2, pages 1165–1172, 1999.

[217] A. Noulas and B. J. A. Krse. Unsupervised visual object class recognition. In *Advanced School of Computing and Imaging Conference*, Lommel, Belgium, 2006.

[218] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley, 2001.

[219] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–407, 1958. (Reprinted in Neurocomputing (MIT Press, 1988)).

[220] V. Vapnik. *The Nature of Statistical Learning Theory*. NY:Springer-Verlag, 1995.

[221] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Machine Learning Research*, 2:265–292, 2001.

[222] W. Zhao, Y.G. Jiang, and C.W. Ngo. Keyframe retrieval by keypoints: Can point-to-point matching help? In *5th Int'l Conf. on Image and Video Retrieval*, pages 72–81, 2006.

[223] M. Blighe, S. Sav, H. Lee, and N. O'Connor. Mo músaem fíorúil: A web-based search and information service for museum visitors. In *International Conference on Image Analysis and Recognition (ICIAR)*, 2008.

[224] M. Blighe and N. O'Connor. Myplaces: Detecting important settings in a visual diary. In *ACM International Conference on Image and Video Retrieval (CIVR)*, July 2008.

[225] M. Blighe, N. O'Connor, H. Rehatschek, and G. Kienast. Identifying different settings in a visual diary. In *9th International Workshop on Image Analysis for Multimedia Interactive Services*, May 2008.

[226] H. Ling and K. Okada. An efficient earth movers distance algorithm for robust histogram comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5):840–853, May 2007.

[227] W. Lin and A. Hauptmann. Structuring continuous video recordings of everyday life using time constrained clustering. *SPIE Symposium on Electronic Imaging*, January 2006. San Jose, CA.

[228] J. Nielsen. *Usability Engineering*. Morgan Kaufmann, 1993.

[229] B. Shneiderman and C. Plaisant. *Designing the User Interface*. Addison Wesley, 2005.

[230] J. Nielsen and H. Loranger. *Prioritizing Web Usability*. New Riders, 2006.

[231] J. Preece, Y. Rogers, and H. Sharp. *Interaction Design*. John Wiley & Sons, 2007.

[232] G. Gay and R. Rieger. Tools and techniques in evaluating digital imaging projects. *RLG Diginews*, 3(3), June 1999.

[233] I. Witten and E. Frank. *Data Mining*. Elsevier, second edition, 2008.

[234] A. Olivia and A. Torralba. Building the gist of a scene: The role of global image features in recognition. In *Visual Perception, Progress in Brain Research*, volume 155, 2006.

[235] J. Beis and D.G. Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *Conference on Computer Vision and Pattern Recognition*, pages 1000–1006, 1997.

[236] D.H. Ballard. Generalizing the hough transform to detect arbitrary patterns. *Pattern Recognition*, 13(2):111–122, 1981.