# A Lifelogging System Supporting Multimodal Access

By

**Zhengwei Qiu**

A dissertation submitted in fulfilment of the requirements for the award of

Doctor of Philosophy (Ph.D.)

to the

Dublin City University

School of Computing

Centre for Digital Video Processing (CDVP)

CLARITY: Centre for Sensor Web Technologies

Supervisor: Dr. Cathal Gurrin

September 2013

## DECLARATION

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of a degree of Doctor of Philosophy is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, 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: ________________ (Zhengwei Qiu)

ID No. 58120378

Date: ______________________

# DEDICATION

Dedicated to:

My wife Na and two little daughters, Anna and Ava.

# ABSTRACT

Today, technology has progressed to allow us to capture our lives digitally such as taking pictures, recording videos and gaining access to WiFi to share experiences using smartphones. People’s lifestyles are changing. One example is from the traditional memo writing to the digital lifelog. Lifelogging is the process of using digital tools to collect personal data in order to illustrate the user’s daily life (Smith et al., 2011). The availability of smartphones embedded with different sensors such as camera and GPS has encouraged the development of lifelogging. It also has brought new challenges in multi-sensor data collection, large volume data storage, data analysis and appropriate representation of lifelog data across different devices.

This study is designed to address the above challenges. A lifelogging system was developed to collect, store, analyse, and display multiple sensors’ data, i.e. supporting multimodal access. In this system, the multi-sensor data (also called data streams) is firstly transmitted from smartphone to server only when the phone is being charged. On the server side, six contexts are detected namely personal, time, location, social, activity and environment. Events are then segmented and a related narrative is generated. Finally, lifelog data is presented differently on three widely used devices which are the computer, smartphone and E-book reader.

Lifelogging is likely to become a well-accepted technology in the coming years. Manual logging is not possible for most people and is not feasible in the long-term. Automatic lifelogging is needed. This study presents a lifelogging system which can automatically collect multi-sensor data, detect contexts, segment events, generate meaningful narratives and display the appropriate data on different devices based on their unique characteristics. The work in this thesis therefore contributes to automatic lifelogging development and in doing so makes a valuable contribution to the development of the field.

# ACKNOWLEDGMENTS

I wish to acknowledge everyone who helped me, encouraged me, and mentored me during my PhD study at Dublin City University in Ireland.

First, I would particularly like to express my gratitude to my PhD supervisor, Dr Cathal Gurrin for his invaluable advice and constant support. Sincere thanks are given to my mentor Professor Alan Smeaton who is always so kind and helpful.

I would like to Dr Aiden Doherty, Dr Hyowon Lee and all other colleagues in CLARITY Research Center who have been both friends and advisors to me. I would also wish to thank Science Foundation Ireland (SFI) and Enterprise Ireland (EI) for funding this research.

Finally, a big thank you to my beloved wife, Na Fu, and my little daughters Anna and Ava who have shared my sadness and happiness at all times. A big thank you also goes to my parents for their constant emotional and physical support. All of the aforementioned people are the strongest ties and motivators throughout these memorable and fascinating times.

# TABLE OF CONTENTS

CHAPTER THREE
RESEARCH METHODOLOGY



CHAPTER FOUR
DETECTING CONTEXTS



CHAPTER FIVE
SEGMENTING EVENTS

CHAPTER SIX
GENERATING NARRATIVES



CHAPTER SEVEN
SUPPORTING MULTIMODAL ACCESS

## CHAPTER EIGHT
## DISCUSSION

## List of Tables

## List of Figures

## List of Abbreviations

ANFIS = Classification and Regression Tree
API = Application programming interface
CART = Adaptive Neuro-Fuzzy Inference System
Cell-tower ID = Cell-tower identification
CPU = Central processing unit
DCT = Discrete cosine transform
FBF = fuzzy-basis-function
GPS = Global positioning system
HCI = Human-computer interaction
HSV = Hue, saturation, value
IEC = International Electrotechnical Commission
ID3 = Iterative Dichotomiser 3
iOS = iPhone operation system
IR = Information retrieval
ISO = International organisation for standardisation
JSON = JavaScript object notation
MAC address = Media access control address
MPEG-7 = Multimedia content description interface
OS = Operation system
PC = Personal computer
PDA = Personal digital assistant
PBE = Password-based encryption
RFID = Radio frequency identification
SD = Standard deviation
SD card = Secure digital card
SIM card = Subscriber identity module card
SMS = Short message service
SVM = Support Vector Machine
UI = User interface
XML = Extensible markup language
UWB = Ultra-wideband
WLAN = Wireless local area network

# CHAPTER ONE

# INTRODUCTION

## 1.1 Overview

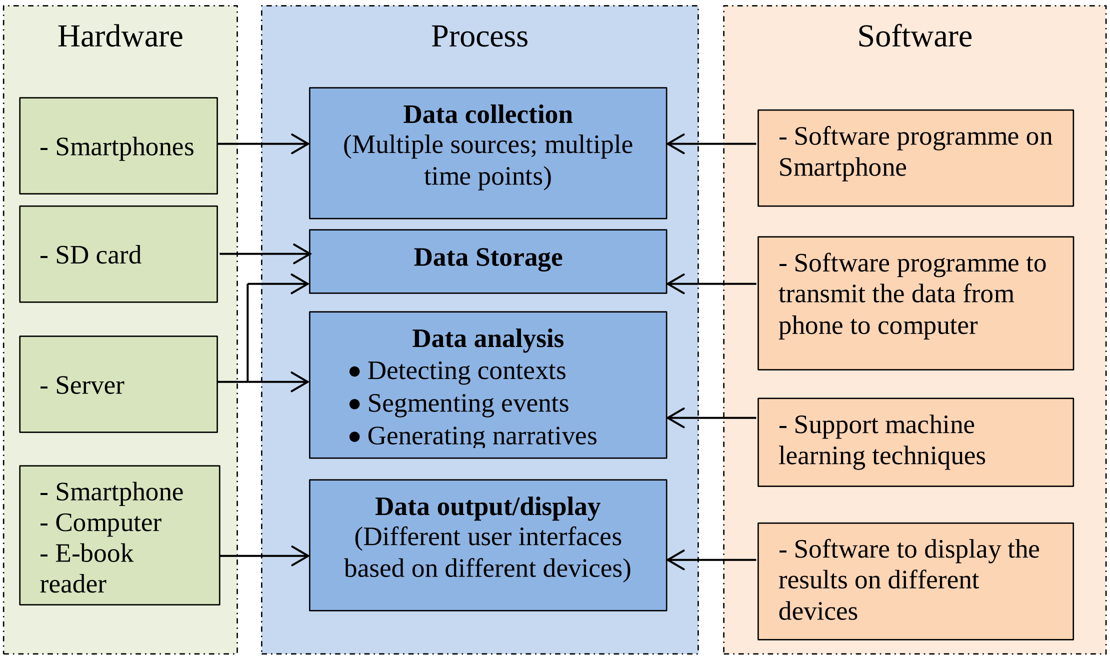
Many people write a diary in order to memorise their daily life and to prevent forgetting. Indeed, there is a growing movement of people called the Quantified Self who record and log aspects of their lives for reasons such as healthcare, efficiency, or general understanding of the self (http://quantifiedself.com). For example, a quantified self-enthusiast may estimate how much they drive every year, how many times they went shopping every month, what their daily blood glucose levels are, and so on. However, maintaining a diary or supporting detailed Quantified Self-analysis requires substantial investment of time and efforts. It is not typically achieved by the majority of people.

Fortunately, in the past decade, we have seen the introduction of numerous new devices such as the smartphone. The widespread usage of smartphones results in the accumulation of vast amounts of personal data. As sporadic capturing of data moves towards "always-on" capture, it becomes necessary to develop a lifelogging system which can segment the lifelog data stream into some manageable units and extract human readable content automatically. According to Smith et al. (2011: 1), lifelogging is "*in essence, the collection of data in order to illustrate a person's life*". A lifelogging system enables a user's life experience to be recorded automatically, and totally captured, allows for future access, and replays data via computing devices (Allen, 2008; O'Hara et al., 2008a; 2008b).

This study adopts the idea of "total capture", i.e. capture as detailed an archive of life experience as possible (O'Hara et al., 2008a; 2008b). Total capture lifelogging is possible with meaningful output and minimal cognitive overhead.

In this thesis, a lifelogging system is developed to investigate the lifelog activities. This system mainly includes four components namely data collection, storage, analysis and display as shown in Figure 1.1. The data in this system is from multiple sources, i.e. collected from different sensors. It is also called data streams. It provides more useful, future-proof and flexible personal life archives.

**Figure 1.1: The work in this thesis**




Source: The author (2013)

Indeed O'Hara et al. (2008b: 6) give a good overview on what motivates us to investigate lifelog activities *"... Every piece of information is such that it is very unlikely, but just possible, that it is valuable. Before technology allowed comprehensive storage, our strategy was usually to try to estimate which information is likely to be more valuable and to keep that. Now there is no reason to stick to that* *philosophy...”*. In other words, we should collect whatever we are able to, i.e. “total capture”. In our research, we argue that the more data we gather, the richer the resultant lifelogs will become. However, because we are still in the early days of lifelogging, many computational techniques that mine knowledge and patterns from our personal life archives may not be known yet. It is more useful to capture as much data as possible now, because it is impossible to re-capture data after the fact.

Lifelogging has many promising applications, e.g. transmitting professional knowledge (Bush, 1945), supporting the data owner’s memory (Sellen et al., 2007) and health monitoring (Lane et al., 2011). There may be many other potential yet undiscovered areas in which lifelog can be exploited since users can be creative in utilising products.

There are many challenges in terms of lifelogging system development. These include the multiple sensors’ data collection tool, large volume data storage, data analysis (combining data from multiple sources, segmenting events and generating narratives), and displaying analysis results using appropriate user interfaces based on different devices. The challenges and our work on how to address these challenges are presented as follows.

- **Multiple sensors’ data collection tool: From SenseCam to Smartphone**

  Existing lifelogging tools are customer-based devices or are combined with some external hardware, e.g. SenseCam (Hodges et al., 2011; Hodges et al., 2006) and DejaView (de Jager et al., 2011). SenseCam is a wearable camera worn via a lanyard around the neck as presented in Figure 1.2. It was designed to take photographs passively without user’s intervention. The SenseCam is expensive and difficult to buy. It is also not easy for people to carry them permanently as people may forget. Sometimes, other people around the carriers are quite cautious about it and may be concerned about the ethics. In addition, these devices can only collect picture data and cannot totally capture a person's life. Therefore, in this research, we make use of the ubiquitous smartphone as our lifelogging device.

The focus of this research is not to develop new hardware devices, to gather a minutely detailed lifelog, or to explore new and novel approaches to machine learning for data analysis; it is to explore how we can use the readily available tools to develop new approaches of managing and presenting detailed lifelogs. In this study, we use the smartphone to collect sensor data, the support vector machine learning (SVM) technique to analyse data. For presenting lifelog data, we use three devices as computer, smartphone and E-book reader.

**Figure 1.2: SenseCam**


Source: Hodges et al. (2011)

The smartphones allow us to capture and access data/information in a ubiquitous manner (Smith et al., 2011), because they are embedded with heterogeneous sensors and always-on networking capability. Furthermore, the smartphone can have a robust operation system (OS) such as Android or iPhone OS (iOS) which are compatible with computer systems. Most importantly, we are able to develop new programmes (apps) to enhance the functionality of the smartphone. This makes it much easier to popularise the lifelogging system on all smartphones employing the same OS without any code change after one lifelogging app is developed on the smartphone. We have already shown that the smartphone is an ideal device to replace (and enhance) the SenseCam, which was the principal lifelogging device of the past decade (Gurrin et al., 2013). Finally, the smartphone becomes very intimate to people's life, because it is one of the typical devices used in people's life. Based on these features of the smartphone, this study will develop a *lifelogging app* running on the Android platform. The details will be described in Chapter 3.

- **Large volume data storage**

Battery and storage are a smartphone's most important resources in the lifelogging system. Even though most smartphones support external SD storage cards, it is quite a small space compared with a computer's storage capacity. On the other hand, the collected data cannot be sent to the server directly, because wireless is not available all the time and wireless connecting cost a lot of battery life. Therefore in the lifelogging system designed and described in this study, all sensor data is stored on the phone SD card temporarily and then is transmitted to the server only when the smartphone is being charged. This is because data transmission using wireless is a battery consuming action, whether it is by WiFi or mobile network.

- **Data analysis by combining data from multiple sources: from image only to multiple data sources such as location, activity and environmental noise etc.**

  For lifelogging users, most lifelog data is not easy to access directly because it can come from different sensors, each of which has a different data format and a different value range. We have not been able to find a data management tool such as a search engine that can be applied directly on lifelog data because of a semantic gap between the user's query and the system. This study will show how six different kinds of contexts may be detected from different sensor data. They are personal, time, location, activity, social and environment.

- **Data analysis on segmenting events from the context change detected by sensors**

  Our lives are continuous; there is no concept of a self-contained document. Our days are viewed as being composed of one chronological and continually unfolding document. The current generation of data management software (database and search engines) are not designed to handle such data streams; hence it is necessary to segment the data stream into a sequence of events that can later serve as retrieval and linkage units to help users to access their lifelog data (Doherty and Smeaton, 2010). This is not a typical challenge for information retrieval as in most cases the document segmentation naturally pre-exists (e.g. WWW pages) or is relatively easy to generate. In our lives, however, there is one clue that can help us to segment life into different events; the change of context can signify a new event. In Chapter 5, we propose an approach to segment lifelog data into events using support vector machine learning (SVM) techniques because of their excellent performance on binary classification.

- **Data analysis on generating narratives for building digital diary**

  Biologists have found that "*Narrative*" or "*Paragraph*" activate more brain regions than "*Word*" and "*Sentence*" which helps humans to recall the past events (Xu et al., 2005). The narrative/paragraph describing daily events have five essential attributes which are *"When"*, *"Where"*, *"Who"*, *"What"* and *"How"*. There are mainly three processes to generate narratives from the segmented events, namely fabula, sjuzet, and discourse generation. In this study, fabula is a series of sentences based on the detected contexts and segmented events; sjuzet is a paragraph of narratives generated from the fabula without the repeated sentences; and discourse is a paragraph of narratives with an illustrated picture/keyframe taken during the event. More details of generating narrative will be described in Chapter 6.

- **Displaying analysis results**

  The availability of new devices such as the smartphone and the E-book reader indicates that computers are not the only way to access personal electronic data. However it also presents new challenges on how to represent data on these different devices during the design of the lifelogging system. These new devices have different features, i.e. smartphone has a smaller screen with colour while E-book reader has a larger screen with black and white. In order to investigate how to display lifelog data more appropriately on these devices, we surveyed users' experience and reviewed the display performance of different devices on eight interfaces for displaying lifelog data. The eight interfaces are images, images and annotations, images and icons, images and narratives, animations, diaries, icons, and narratives. More details and results will be described in Chapter 7.

## 1.2 Hypotheses

To address the new challenges in lifelogging research as multiple sensors' data collection, large volume data storage, data analysis and representing lifelog data on different kinds of devices, a lifelogging system is developed in this dissertation. Three hypotheses are proposed based on the literature reviewed in Chapter 2 and are summarised as follows.

***Hypothesis 1: Event segmentation can be performed effectively by detecting the changes in sensor data.***

***Hypothesis 2: A meaningful textual narrative that accurately represents an event can be generated automatically.***

***Hypothesis 3: Different access devices benefit from different representations of lifelog data.***

## 1.3 Deliverables from Dissertation

The major deliverables resulting from the work carried out during this dissertation include:

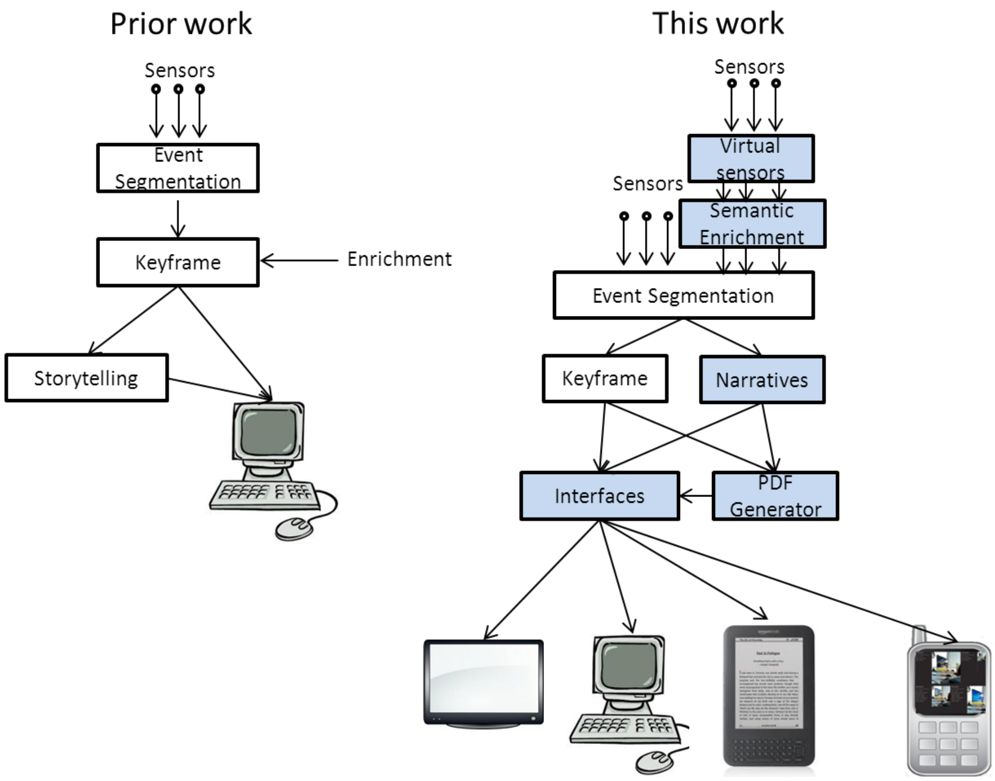
1) Firstly, in this study, a new generation of lifelogging tool has been developed to collect, store, analyse and display lifelogging data automatically. It does not require any user input to operate. A user just wears the smartphone and may the lifelog at any point afterwards. Therefore, this could reduce the

barriers-to-entry into lifelogging for a user who may not have been engaged in the process before.

2) A lifelogging tool which can collect a full range of sensor data from a smartphone in a power-efficient manner. Since the system is context aware, it can learn the user's situation and decide which sensor needs to be turned on or off in order to maintain all-day data capture.

3) Approaches to extract semantic concepts from raw sensors which help to bridge the semantic gap between the human and machine.

**Figure 1.3: The comparison between the most similar work (Byrne et al. 2011) and this work**




Note: The blue parts are new compared with Byrne et al. (2011).

Source: The author (2013)

4) A real-time lifelogging system which can analyse lifelog data such as face detection and upload data to a server in real-time. With that, users can easily browse and share their status using a web browser.
5) A first mapping of lifelog events through representation on multiple popular devices. In the existing research, Byrne et al. (2011) is the most similar one to the model in this study as shown in Figure 1.3. The model in this study is more comprehensive than that of Byrne et al.'s (2011). This work offers suggestions on the most suitable representations to enable fast access to lifelog data using different devices.
6) Methods for obtaining a user's location using a fusion of GPS, WiFi, Bluetooth and Base Station in order to complement their strengths and weaknesses. Compared with using GPS alone, locating a user by WiFi and Bluetooth will dramatically extend the smartphone battery life.
7) A new approach using support vector machine learning (SVM) to segment lifelog stream data into events.
8) A new user activity generation tool based on real-world accelerometer.
9) An approach to generate a narrative of event using all the concepts extracted from physical and virtual sensor data.

## 1.4 Contribution of this Study

Lifelogging is believed likely to become a well-accepted technology in the coming years (Doherty, 2008). The manual logging, i.e. writing paper diaries to record every daily event, is not possible for most people and not feasible in the long term. The automatic life logging, i.e. the lifelogging system in this study, is needed.

Lifelogging has many promising future applications. Examples include transmitting professional knowledge (Bush, 1945), supporting the data owner's memory (Sellen et al., 2007), health monitoring (Lane et al., 2011) and the wide range of usages suggested by Bell and Gemmell (2009) in their book "*How the e-memory revolution will change everything*". Many other potential yet undiscovered areas exist where lifelogs may be exploited by users in future generations. This study introduces a new lifelogging system which includes multiple sensors' data collection, large volume data storage, data analysis through detecting contexts, segmenting events, generating narratives and representing results to users. By doing so, this study has shown that event segmentation can be performed effectively by detecting the changes in sensor data; a meaningful textual narrative that accurately represents an event can be generated automatically; and different access devices benefit from different representations of lifelog data. Along with the deliverables indicated in the previous section, the work in this dissertation therefore contributes to automatic life logging development.

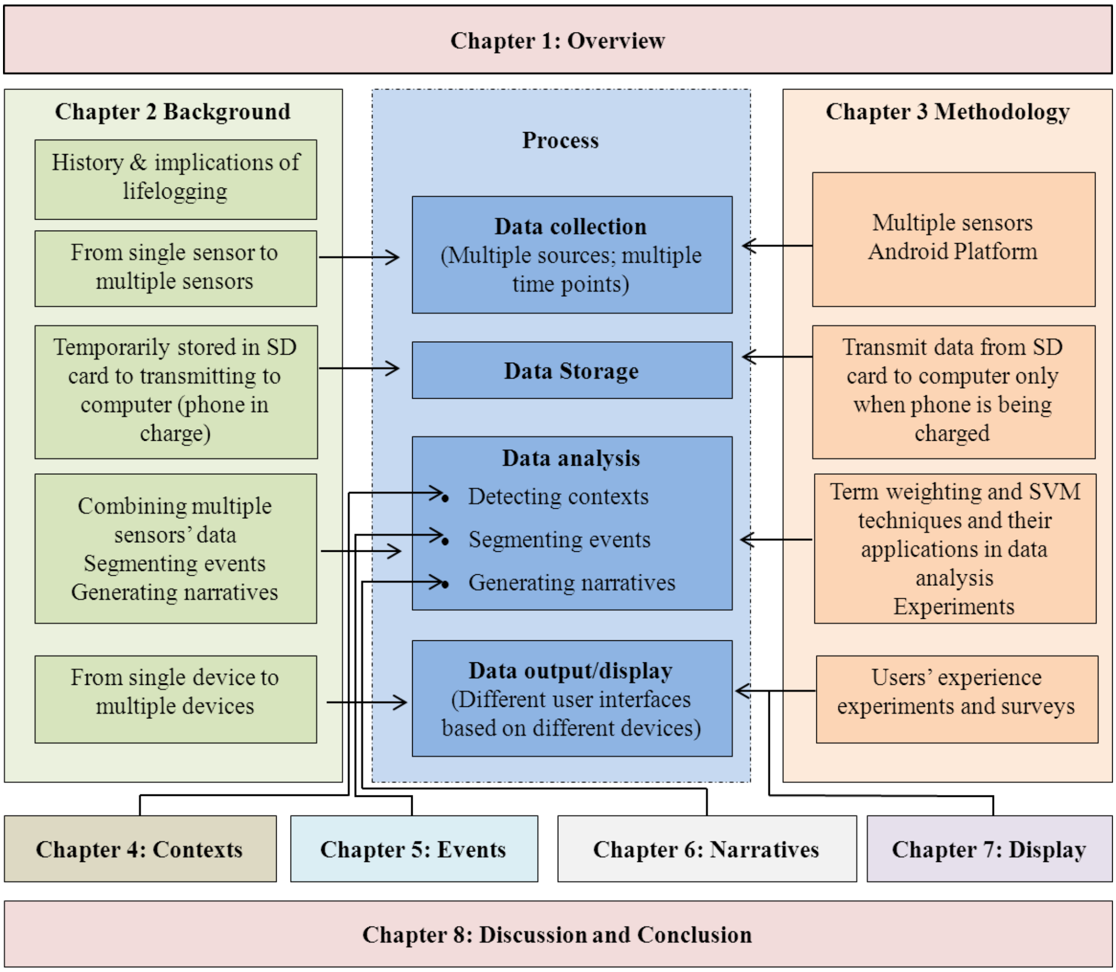
## 1.5 Thesis Structure and Outline

Figure 1.4 presents the structure of this thesis. Chapter 1 introduces an overview of this dissertation which includes the background, hypotheses, expected deliverables, contribution of the study, and the structure of the thesis.

In Chapter 2, a comprehensive overview of the relevant literature is provided in relation to the history and implications of lifelogging. Thereafter, it introduces the evolution of the lifelogging tool from SenseCam to smartphone for data collection. It also investigates the previous research on event segmentation and narrative generation approaches employed in other domains apart from lifelogging. Finally, this chapter exploits how other researchers summarised and visualised large amounts of sensor data.

Chapter 3 describes the methodology employed in different processes of the proposed lifelogging system. It firstly describes the data collection based on multiple sensors and the Android platform. Then the data analysis techniques employed in this study such as term weighting and SVM are introduced. Finally, it presents all of the experiments used in data analysis and data representation process of the system.

**Figure 1.4: The structure of this thesis**




Source: The author (2013)

Chapter 4 presents the details on combining different data streams, i.e. data collected from multiple sensors. Following that, it discusses the need to and explains how to convert raw sensor data into semantic contexts. The process of implementing virtual sensors for different contexts is described. Compared with the physical sensors, it is shown that the virtual sensors can be easily employed with better fault tolerance.

Chapter 5 focuses on the event segmentation in the data analysis process. It firstly provides a detailed explanation on how to segment sensor data into events. It also demonstrates the attributes to be extracted from raw sensor data and the contexts used to segment sensor data into small events. It then describes the experiments for testing hypothesis 1 which proposes that the event segmentation can be performed effectively by detecting changes in contexts. Finally, to select a good keyframe for each event, three contexts' (social, activity and environment) effect on selecting users' keyframe for their daily life are investigated where the best keyframe selection method (combining face detection and image quality) is found.

Chapter 6 provides an approach to generate a narrative for each event in the proposed lifelogging system. The concepts of fabula, sjuzet and discourse used in previous research are used to generate meaningful and effective narrative summaries of events (Cheong and Young, 2008). These concepts are used in testing hypothesis 2 which proposes that a meaningful textual narrative that accurately represents an event can be generated automatically.

Chapter 7 presents the last process of the proposed lifelogging system – representation of results. To assist a user's access to their lifelogs, an investigation was conducted on which user interface is more appropriate on different devices for accessing lifelog data. Results from the user experience experiment indicate the different data displaying performance on different lifelogging devices. This allows us to test hypothesis 3 which proposes that different access devices benefit from different representations of lifelog data.

Finally in Chapter 8, a brief summary of this study is firstly provided. The contribution and limitation of this study is discussed. Recommendations are made for future research.

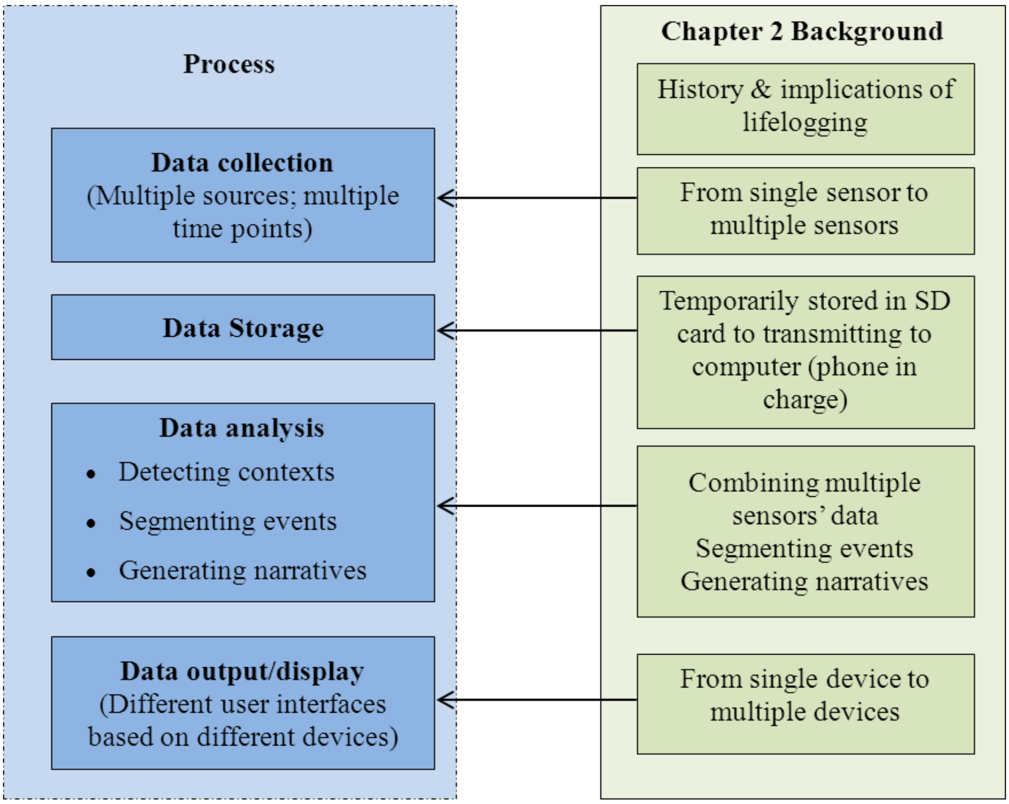
# CHAPTER TWO

# BACKGROUND AND HYPOTHESES

## 2.1 Introduction

In this chapter, the history and implications of lifelogging are introduced. Thereafter, the evolution of lifelogging tools for data collection from SenseCam to smartphone is summarised. The previous research on context detection, event segmentation and narrative generation employed in other domains apart from lifelogging is then reviewed. Finally, how other researchers summarised and visualised large amounts of sensor data are provided. Figure 2.1 presents the work in this chapter.

**Figure 2.1: Work in Chapter 2**




Source: The author (2013)

## 2.2 The History of Lifelogging

Throughout human history, people have tried to record daily events to enable these events to be recalled at a later stage, i.e. from cave painting to writing paper diaries. For this work, we are concerned with lifelogging practice, i.e. digitally capture and display user's life. As far back as the end of the Second World War, Vannevar Bush introduced the Memex which is a life knowledge organisation hypermedia system operating as a desk-based device (Bush, 1945). Memex introduced new concepts such as information links or trails, which are created by the individual or by others. Memex was described as an "*enlarged intimate supplement to one's memory"* (Bush, 1945). In these words, Bush (1945) identified some of the key issues for maintaining a personal lifelog: enlarged (store as much information as possible), intimate (private to the owner) and supplemental (working in synergy with one's memory). As defined by Smith et al. (2011: 1), lifelogging is "*in essence, the collection of data in order to illustrate a person's life*". The concept of lifelogging in this study focuses on the digital tools for data collection, storage, analysis and display.

Only in recent years has the Bush equal vision of lifelogging become feasible [which occurred] through advances in sensor data, data storage and, and information access, which are indeed the focus of this thesis. Ten years ago, the hardware to support lifelogging was not readily available.

Early pioneers such as Steve Mann (Mann, 1997) developed customised lifelogging and ubiquitous computing hardware, called EyeTap Digital Eye Glass. It is the earliest recorded lifelogging tool. Since the development of the Microsoft SenseCam about ten years ago (Hodges et al., 2006), lifelogging has attracted increased attention from scholars and practitioners. There is now a suite of new technologies that can monitor and log aspects of a person's life, ranging from single sensor devices to the multi-sensor smartphone apps and custom devices such as Oxford Metrics Group (OMG) Life Autographer (www.autographer.com).

Beginning with the single sensor devices, a good example is the Fitbit Tracker, a health monitoring lifelogging tool (www.fitbit.com). It is a wearable device that utilises an on-board accelerometer to measure data such as the number of steps walked, energy burn and the quality of sleep. The author would consider this to be a basic lifelogging device. At the other end of the scale are full-featured lifelogging devices such as the Autographer and Memoto which use a wide range of sensors to log many aspects of a person's life, e.g. camera, accelerometer etc. Applying semantic context extractor tools, such as the one in this thesis can turn these devices into extensive lifelogging tools. Memoto (memoto.com) is similar to Sensecam but is a smaller wearable lifelogging camera which can take two pictures per minute and includes a reduced sensor set (e.g. no accelerometer). At the time of writing (September 2013), it is still in development by Memoto AB based in Sweden. Autographer is an evolution of the original Sensecam that now incorporates a relatively high quality camera, GPS and many of the original SenseCam sensors. The Autographer has gone on sale in August of 2013.

Other devices worth mentioning include wearable video cameras (single sensor life capture devices) such as the Looxcie camera. Looxcie (www.looxcie.com) is a wearable video camera that facilitates live streaming of video and on-board capture of many hours of video data. Finally, there are the general purpose computing devices that can be programmed to act as lifelogging devices. One example is the smartphone, which we use in this work. Another example is Google Glass.

Developed by Google Inc., Google Glass (www.google.com/glass/start) is a wearable computing device that can be programmed to capture video/photos of what the wearer sees as well as providing real-time information and feedback via an optical head-mounted display. What is important to note here is that the range of lifelogging devices is increasing in recent years.

Until now, lifelogging has appeared to be an extreme activity carried out by only a small number of people (e.g. Bell and Gemmell, 2007; Doherty et al., 2011; Mann, 1997). There are three main reasons why personal lifelogs have not been used for the general population: 1) privacy and ethical concerns (Nguyen et al., 2009); 2) overwhelming amounts of data (Sellen and Whittaker, 2010); and 3) limited device availability.

On 1) the privacy and ethical concerns, it is our conjecture that personal lifelog is likely to follow the experiences of smartphone cameras, Facebook and other social networking sites and smartphone location tracking services. Once the personalised experience (Nakamura et al., 2010), wellness (Lane et al., 2011) and memory capture and sharing (Mathur et al., 2012) present a wide range of benefits to end users; these concerns will likely become of secondary importance. In other words, society will develop an acceptable usage policy and embrace the new technology. In this study we assume that there will be willing lifeloggers in the future who will benefit from the findings of this research.

Secondly, for the overwhelming amounts of data, existing research has indicated that lifelog data can be summarised into useful knowledge through:

1) segmenting it into a series of distinct events or activities (Doherty et al., 2011),

2) automatically labelling those events from both the content (Doherty et al., 2011) and context (Jain and Sinha, 2010),
3) automatically detecting faces to identify those episodes that are more interesting to reflect on (Doherty et al., 2012),
4) presenting segments of this personal lifelogs to the user as required,
5) analysing this data to provide new knowledge to the user.

This is our driving force. We aim to identify the challenges and build a semantically rich lifelog and evaluate it using various access approaches.

Thirdly, we have limited device availability. Current lifelogging data collection tool, e.g. SenseCam, helped bring lifelogging work to memory and to public health researchers. However, it is still confined to a tiny segment of the population. Based on our work (Qiu et al., 2011; Qiu et al., 2012) as well as the work by de Jager et al. (2011), this situation is about to change. In this work we have demonstrated that the smartphone is the new supporting technology for generating personal lifelogs with real-time ability and supports human prospective memory. Using smartphone significantly reduces the costs and provides lifelogging data analysis in a real time manner.

## 2.3 The Potential Implications of Lifelogging

Lifelogging has many promising future applications. Examples include transmitting professional knowledge (Bush, 1945), supporting the data owner's memory (also called memory aids) (Sellen et al., 2007; Wood et al., 2012), health monitoring (Doherty et al., 2013; Lane et al., 2011), a mental health tool (Rennert and Karapanos, 2013; Son et al., 2013), a social network analysis tool (Sueda et al., 2012), and even as an urban design tool (Ihara et al., 2011). Lifelogs can help objectively supply data and reveal potential errors inherent in self-reporting (Doherty et al., 2013). There are likely to be many other potential yet undiscovered areas where lifelogs may be exploited by users in future generations. They may invent some unexpected creativity in utilising products.

The potential for personal lifelogs is enormous. We do acknowledge that there are challenges to overcome, such as privacy concerns, data storage, data security and the development of a new generation of search and organisation tools. However, it is believed that these challenges will be overcome. This research will represent an important contribution and turning point for society. The quantified individuals know more about themselves than ever before, have more knowledge to improve their lives, and can share life events and experiences in richer detail with friends.

## 2.4 The Evolution of the Lifelogging Tool for Data Collection

Lifelogging research is encouraged by the arrival of new technologies. The initial lifelogging research used individual devices such as SenseCam (Hodges et al., 2006) and DejaView (de Jager et al., 2011). More recently lifelogging tools include smartphones embedded with more sensors and new platforms. We will begin with a discussion of the SenseCam.

### 2.4.1 SenseCam

The SenseCam was developed by Microsoft Research in Cambridge U.K. It is a wearable camera worn via a lanyard around the neck as shown in Figure 1.2 (page 4). The SenseCam is designed to take photographs passively without a user's intervention. It was initially created as a tool to visually record life experience and has been extensively used by memory professionals. It is about the size of a four-pack of AA batteries. None of the technologies included in the SenseCam are in themselves a great advancement, but it has become the pre-eminent passive capture device in academic research. The SenseCam captures an image approximately every 22 seconds when triggered by onboard sensors (Hodges et al., 2011). Because the SenseCam is a wearable device, these images are oriented with the wearer and thus capture life activities first-hand. Unlike a digital camera, the SenseCam has a fisheye lens, to maximise the field of view, and incorporates multiple sensors including light sensors (intensity and colour), a multi-axis accelerometer, a thermometer and a passive infrared sensor to detect the presence of a person. All these mean that a SenseCam can normally capture up to 5,000 images per day, depending on the wearer's activities. More social interactions will instigate more image captures (Hodges et al., 2011). This concept of automatic capture is essential for any form of personal lifelogging data collection.

Although SenseCam is quite small and easy to carry, its primary limitation is its off-line analysis (Hodges et al., 2006). A typical SenseCam wearer takes the camera off and uploads the content at the end of the day. Although the SenseCam gathers thousands of photographs per day (powerful memory cues and sources of evidence for semantic computer vision technologies) and tens of thousands of sensor streams, it is limited in on-board sensing and off-line processing.

Furthermore, the main obstacle to the widespread adoption of SenseCam is that it requires users to purchase, maintain and operate a dedicated hardware device; it cannot integrate users' lives seamlessly (Rawassizadeh et al., 2012). It is proposed that greater adoption would be greatly facilitated if the SenseCam functionality could be integrated with a device that users already own and are accustomed to charging and maintaining such as smartphone (Gurrin et al., 2013).

### 2.4.2 Smartphone

Due to the limitations of SenseCam (such as high cost, on-board sensing and off-line processing), there have been some attempts at integrating the lifelogging tool on different devices. For example, Abe et al. (2009) implement a lifelogging tool on a remote control which can record a user's browsing history on TV, temperature, location, and acceleration. Gellersen et al. (2002) implemented a lifelogging tool on a cup to monitor users' drinking behaviours. However, neither a remote control nor a cup is easy to carry every day and everywhere.

One obvious solution is the increasingly ubiquitous smartphone. Moving from the SenseCam to the smartphone as a lifelogging tool is important because a smartphone has more sensors and new platforms (e.g. Belimpasakis et al., 2009; Cheng et al., 2010; Chennuru et al., 2012; Lu et al., 2010). In our own research, we have shown that the smartphone can act as a real-time SenseCam equivalent (Gurrin et al., 2013). However, most lifelogging research groups are facing one big challenge: battery life.

Battery life directly affects how much a user can benefit from the mobile device. This is because the lifelogging applications/systems implemented on a smartphone usually drain the battery very fast (e.g. Hansen et al., 2009). For example, the wireless communication in smartphone utilises more battery than any other functionality apart from the CPU. In a lifelogging system, the wireless connection needs to be turned on as all location information comes from it (Rawassizadeh et al., 2012). To overcome the battery consumption issue, a lot of research uses the context-aware technique in lifelogging systems (e.g. Doswell, 2006; Gurrin et al., 2008). Context-aware techniques are efficient as they automatically turn the sensor on/off to save the battery life based on the environmental conditions. For example, SensLoc developed by Kim et al. (2010) will turn WiFi on automatically when there is no GPS signal.

In this study, we develop a context-aware lifelogging system based on the Android smartphone. The proposed lifelogging system doesn't require any specific hardware or additional devices. In addition, it enables the capture of a full day's data with a smart power managed strategy. Full details will be discussed in Chapter 3.

## 2.5 Lifelog Data Storage

Lifelog data storage is not the most important issue for users as mobile devices have increasingly large storage (Coughlin, 2008). However, it should be kept in mind that lifelogging tools are developed to capture a person's day-to-day activities in the longer term. Even large capacity devices can be filled very quickly. For example, a SenseCam can capture up to 5,000 photographs per day. Its onboard memory is enough to store one week's data. Unless the data is removed from the device in time, SenseCam may corrupt the data collected during the whole period. For example, it may overwrite the old file or stop collecting data (Byrne, Kelly & Jones, 2012).

Rawassizadeh et al. (2012) propose that a server with enough storage capacity must be the main location for storing data, while local storage on the lifelogging tool can only be used as the temporary storage. This is because there is less disk space available on smartphones and other portal devices than on data servers. Similar solutions have been used in other relevant projects. For example, iRemember can immediately transmit this form of data via a high-speed wireless network to a large-capacity server where the data is archived (Vemuri et al., 2006). However, in the real world, the high-speed wireless internet is not available everywhere. For example, in many urban areas, there is no WiFi hotspot available at all. The mobile network is available most of time, but it will cost the user too much financially to upload the big volume of data to the server. Additionally wireless connecting costs a lot in terms of battery life.

In this thesis, we propose a strategy to cope with the above issues. The strategy is to store all sensor data on the phone’s SD card temporarily and transmit it to the server when the smartphone is being charged. This will address both real-time access and optimising battery consumption

## 2.6 Lifelog Data Analysis

In the past decade, a number of lifelogging tools were developed to facilitate research. However, all researchers in the lifelogging community are facing the same challenge regarding the efficient access of information which is useful to the user (Aizawa et al., 2004; Tancharoen et al., 2006). We will introduce six contexts used in data analysis before introducing the data analysis techniques employed in this study. They are personal, time, location, activity, social and environment contexts.

### 2.6.1 Detecting Contexts

Most lifelog data is in numeric, image or other multi-media format. How to efficiently access the information using an easy and useful way for users is the common challenge facing most researchers in the lifelogging community (Aizawa et al., 2004; Tancharoen et al., 2006). One approach is to transform the sensor data into the context which is easy for users to read. For example, GPS data contains three-dimension information: latitude, longitude and altitude. The digits indicating latitude, longitude and altitude make no sense to users in their raw format. If the three-dimension GPS data is transformed to a location context, such as a home address, it will be very easy for the user to read and understand. Besides location context, we propose that there are five other commonly used contexts including time, personal, activity, social and environment which could be detected from sensors (Chen and Kotz, 2000; Kern et al., 2003).

The traditional approach for using the lifelog data is to scan/browse the lifelog data and find the event of interest by users. However, it is very time consuming and has low accuracy. Take the photograph as one example; photographs have been used mostly as lifelog data source. To generate useful description, users need to scan/browse a large volume of lifelog photographs to find the content with interest. Due to the extremely large volume of photograph data, it is obviously a time consuming activity. In addition, the accuracy is very low. As shown in Doherty et al. (2012), it takes on average more than ten minutes for a normal and healthy individual to locate an event of interest from a two and half-year lifelog and this is with only a 25% success rate. If one cannot recall the date or time as an access mechanism it would take several days or weeks to watch an entire lifelog for one year, using a photograph browser to discover the experience of interest (Aizawa, et al., 2004a). Another example is textual search engines. Textual search engines have been used for decades and are considered as an efficient and effective method of information retrieval (Baeza-Yates and Ribeiro-Neto, 1999). However, they can only search text documents which require a human to annotate first (Hatcher et al., 2004). Even though multimedia information retrieval techniques can search multimedia information using a search engine, most retrieval comes from related textual media data, such as description, subtitle, caption and comments which are mostly generated manually.

Context is widely accepted in computing science to semantically add meaning to non-textual data. Schilit et al. (1994: 85) stated that the constituents of context are; *"the location of use, the collection of nearby people, hosts, and accessible devices, as well as to changes to such things over time"*. On a conceptual level, it is also argued that further issues such as temperature, environmental noise level and social situation are of interest and can be usefully employed as additional sources of contextual metadata.

Dey (2001: 3) provided the following general definition, which is probably the most widely accepted: *"Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between the user and the application, including the user and the applications themselves".* This definition clearly states that context is always bound to an entity. In other words, the information describing the situation of an entity is context. However, in using indefinite expressions such as *"any information"* and *"characterize the situation",* the definition is very general. In another paper, Dourish (2004: 21) gave more detail about the notion of context. He stated that "*context is a form of information, context is delineable, context is stable"* and "*context and activity are separable"*. Dey et al. (2001: 106) extended their definition of context with the statement that *"Context is typically the location,* *identity and state of people, groups, and computational and physical objects"*. This study adapted this definition. The author has taken all the available sources of context and utilised them in this work as a source of semantics and evidence for semantic enrichment. The six contexts used in this study include personal, time, location, activity, social, and environment. How to detect these contexts from lifelog data will be presented in Chapter 4.

#### 2.6.1.1 Personal Context

Personal context is widely used in context-aware systems for describing the user's profile (Cheverst et al., 2000). Personal context not only contains gender and age, but also personal habits and personal life patterns which can be derived from activity and social contexts. The existing lifelogging systems identify different behaviours based on different user profiles. For example, the context-aware guide system can recommend to the user a route based on their interests (Abowd et al., 1997). Mazhelis et al. (2011) developed a recommendation system which determines the probability of a route the user will drive based on the historical data. Personal context also contains and decides other contexts. For instance, different people have different home addresses and different social relationships. It is difficult to separate the personal context from other contexts. People's activities and travel behaviours might be decided by personal context (Liu et al., 2013). For example, a user's age can influence his location pattern and social activity. Most elder people do not work full time and spend much of their time at home. Their activity intensity is quite low compared to that of a young person.

#### 2.6.1.2 Time Context

Time is part of the measuring system used to sequence events, to compare the durations of events, the intervals between them, and to quantify rates of change such as the motions of objects. The temporal position of events with respect to the transitory present is continually changing; events happen, then are located further and further in the past (Dowden, 2011). There are two catalogues of time; absolute time and relative time. According to Newton (1802) absolute time exists independently of any perceiver and unlike relative time progresses at a consistent pace throughout the universe. As humans, we can only understand relative time, which is the measurement of perceivable objects in motion (like the moon or sun).

Previous lifelogging tools used different time clocks which makes it difficult to integrate data. For example, in previous research, SenseCam was used to capture photographs, and a GPS receiver was used to collect location information. However, the devices used different system clocks. We have focused on identifying a way to synchronize the time on different devices. The SenseCam will lose the time setting when it is out of power. This makes it very difficult to integrate location with visual data (Gemmell et al., 2005). In this work, all sensor data collected on a smartphone is timed with one clock. The system also records satellite time when it captures GPS data. It is thus very easy to identify any data with the wrong timestamp by comparing the two timestamps. The server can also correct data using satellite time automatically.

#### 2.6.1.3 Location Context

Location is one of most commonly used contexts. It not only enables many location-based services (Abe et al., 2009; Azizyan et al., 2009), but can also be used for the identification of a user (Shi et al., 2011). Many studies have been conducted to extract location context through two stages: 1) investigating suitable techniques to acquire a user's location such as GPS and WiFi; and 2) transforming the location information to some meaningful information for users, such as transforming GPS data to a semantic meaningful text annotation, e.g. home.

In the first stage, most of the research focuses on investigating suitable techniques to acquire a user's location. GPS is widely used in lifelogging research because it is the most accurate and direct method for acquiring a user's location. However, GPS is not flawless. Compared with other ways (e.g. WiFi), GPS needs specialist hardware and requires a certain amount of time to start from a "cold-start" mode which consumes a lot of battery power (Rekimoto et al., 2007). Furthermore, it cannot work inside big buildings where the indoor environments are typically complex (Gu et al., 2009). For example; walls, equipment and humans influence the propagation of electromagnetic waves which leads to multi-path effects for GPS. Some interference and environmental noise sources from other wired and wireless networks degrade the effectiveness of GPS positioning. Meanwhile, most human activities occur indoors. For example, one study by Klepeis et al. (2001) has shown that people spend approximately 89% of the time indoors and 5% in a vehicle with the remaining 6% spent outdoors.

The history of using cell-tower ID to identify location is another commonly used location technique. Compared with other sensory data such as GPS and WiFi scans, using cell ID to acquire location has no extra cost because it is available on most handheld devices (Trevisani and Vitaletti, 2004) and it can even work in buildings. Recently some research groups employed it to locate users (e.g. Liu et al., 2013; Shi et al., 2011). However, its accuracy is the lowest of all locating sensors. To get more accurate indoor location, some researchers have tried using infra-red, ultrasound, radio-frequency identification (RFID), wireless local area network (WLAN), Bluetooth, sensor networks, ultra-wideband (UWB), magnetic signals, vision analysis and sound (e.g. Bargh and de Groote, 2008; Kaleja et al., 1999; Pahlavan et al., 2000). The existing studies all used stations whose locations are known and are in environments which are already set-up for such contextual data collection. Therefore, they are not suitable for general use in real world lifelogging.

Recently, WiFi locating techniques have started to gain more attention. Compared with GPS-based locating, using WiFi for locating takes less time to acquire accurate location. More importantly, it can work inside a building (Rekimoto et al., 2007). To utilise the advantages of different location sources, Kim et al. (2009) demonstrated that their system which combines GPS and WiFi data sources is more efficient than using a single source. Inspired by Kim et al.'s (2009) findings, this study will use GPS, Cell ID, WiFi entry and Bluetooth as the location context sources.

Compared with location, users are more interested in "places" such as the home and the work place. This is because a place is a locale which is important to a user and carries important semantic meanings for them (Liu et al., 2013). This encourages the second stage of location context research: transforming location to a place that is meaningful for users. Concerning the actual analysis of location data to mine important concepts, Wolf et al. (2001) reported a technique which analyses GPS trips in order to automatically identify trip purpose and to maintain travel diaries. Liao et al. (2007) identified significant places by using a trained modelling technique on a small dataset of four people. Ashbrook and Starner (2003) described a clustering technique to identify meaningful locations and received positive evaluation from multiple users. Finally, Kang et al. (2004) described an approach to automatically identify significant places using the clustering technique. They received positive evaluations on short (two day) logs from a small number of people.

This research differs from all the above. It reports an experiment for mining significant locations using a fast processing, threshold-free approach which is based on text retrieval. This approach is evaluated on a multi-year location archive which will be described in Chapter 4.

#### 2.6.1.4 Activity Context

Automatic recognition of human activities, such as sitting, driving, lying and walking, is one of the most important and challenging areas in lifelogging research (Takata et al., 2008). Different from other contexts such as time and location which is detected at one time point; the activity context requires combining data collected from many sources during a period. To mine and extract useful information from the extremely large volume of data, a machine learning technique is adopted by researchers to address this issue (e.g. Fitzgibbon and Reiter, 2003). As shown in Table 2.1, a lot of research based on machine learning has been conducted in the past decade on detecting semantic activity context.

Detecting activity context using machine learning is a nascent research area. By reviewing the results of past research on activity detecting, it was found that no single algorithm could produces the best overall performance. For example, Lu et al. (2010) found that the optimal performance could be achieved by different machine learning techniques for different contexts. For example, they found that support

**Table 2.1: Examples of detecting activity context from lifelogging data using machine learning techniques**

| Source | Contexts | Algorithms |
|---|---|---|
| Bao and Intille, 2004 | Walking, Walking carrying items, Sitting & relaxing, Working on computer, Standing still, Eating or drinking, Watching TV, Reading, Running, Bicycling, Stretching, Strength-training Scrubbing, Vacuuming, Folding laundry, Lying down & relaxing, Brushing teeth, Climbing stairs, Riding elevator, Riding escalator | C4.5 Decision tree and Naive Bayes |
| Berchtold et al., 2010 | Walking, No-movement, Standing, Lying, Climbing stairs, Cycling, Holding, Talking on phone, Typing text message | Fuzzy Logic |
| He and Jin, 2008 | Running, Standing still, Jumping, Walking | SVM |
| Jatobá et al., 2008 | Lying, Standing, Jogging, Walking, Climbing upstairs, Climbing downstairs | ANFIS, CART decision tree, ID3 decision tree, Nearest Neighbor, k-Nearest Neighbor and Naive Bayesian |
| Kao et al., 2009 | Brushing teeth, Hitting, Knocking Working at a PC, Running, Swinging, Walking | FBF-based |
| Lara et al., 2011 | Walking, Running, Sitting, Ascending, and Descending. | Naive Bayes, Bayesian Network, C4.5 Decision tree, Neural Network, Decision Stump and etc. |
| Lee et al., 2011 | Lying, Standing, Walking, Going-upstairs, Going-downstairs, and Driving | Artificial neural networks |
| Maurer et al., 2006 | Sitting, Standing, Walking, Ascending stairs, Descending stairs, Running | C4.5 Decision Trees, k-Nearest Neighbor, Naive-Bayes and the Bayes Net classifier. |
| Tapia et al., 2007 | Lying down, Standing, Sitting, Walking, Running, Ascend stairs, Descend stairs, Cycling | C4.5 Decision tree and Naive Bayesian |
| Yan et al., 2012 | Stand, Slow Walk, Sit Relax, Sit, Normal Walk, Escalator Up, Escalator Down, Elevator Up, Elevator Down, Down Stairs | C4.5 Decision tree |
| Zhu and Sheng, 2009 | Walking level, Walking upstairs, Walking downstairs, and Running | Markov models |

**Table 2.2: Comparing machine learning algorithms**

| | Decision Trees | Neural Networks | Naïve Bayes | kNN | SVM | Rulelearners |
|---|---|---|---|---|---|---|
| Accuracy in general | ** | *** | * | ** | **** | ** |
| Speed of learning with respect to number of attributes and the number of instances | *** | * | **** | **** | * | ** |
| Speed of classification | **** | **** | **** | * | **** | **** |
| Tolerance to missing values | *** | * | **** | * | ** | ** |
| Tolerance to irrelevant attributes | *** | * | ** | ** | **** | ** |
| Tolerance to redundant attributes | ** | ** | * | ** | *** | ** |
| Tolerance to highly interdependent attributes (e.g. parity problems) | ** | *** | * | * | *** | ** |
| Dealing with discrete/binary/continuous attributes | **** | *** | *** | *** | ** | *** |
| Tolerance to noise | ** | ** | *** | * | ** | * |
| Dealing with danger of over fitting | ** | * | *** | *** | ** | ** |
| Attempts for incremental learning | ** | *** | **** | **** | ** | * |
| Explanation ability/transparency of knowledge/classifications | **** | * | **** | ** | * | **** |
| Model parameter handling | *** | * | **** | *** | * | *** |

Note: **** stars represent the best and * star the worst performance

Source: Kotsiantis et al. (2007)

vector machine learning (SVM) is best for cycling, the Gaussian Model for driving and walking, and the Decision Tree for running. Furthermore, even for the same contexts, the different optimal algorithms were found in different studies (Lara et al., 2011). Kotsiantis et al. (2007) compared the best-known classification algorithms, and found that there was no one machine learning algorithm whose performance can be greater than all other classification techniques as shown in Table 2.2.

Based on the above description on “no best” machine learning technique, we decided to adopt a conventional and proven technique - support vector machine learning (SVM) as our machine learning methodology. As can be seen from Table 2.2, SVM provides generally good performance (Kotsiantis et al., 2007). More details on SVM and its applications in this research are presented in Chapter 3.

#### 2.6.1.5 Social Context

Social context indicates “who”, for example (“who” is talking to the user). Bluetooth can be used as a sensor to collect social context. It is a short-range wireless protocol which enables the exchange of data among two or more devices. It is increasingly routinely included in a wide variety of electronic devices from home computers to portable laptops, smartphones, tablets, keyboards, mice, mp3 players and headphones.

By gathering and analysing the presence of nearby Bluetooth devices, Nicolai et al. (2006) defined the concept of familiarity within the Bluetooth space. Their work demonstrated that social context could be drawn from general encounters with devices. They defined three types of social relationship: “familiar”, “familiar strangers” and “strangers”. A “familiar” can be detected as a person familiar to the user, typically the user’s friends, family or work colleagues. “Familiar strangers” are the people encountered on a somewhat regular basis, for example, in a pub or a shop. Typically interaction will never occur with these people. Finally a “stranger” is outside the friends/family/work-colleagues group. In the work by Lavelle et al. (2007), the authors presented an approach for utilising a mobile device’s Bluetooth sensor to automatically exploit the familiarity of other devices in order to decipher which users are important to individual users in the real world.

In the lifelogging domain, Doherty and Smeaton (2008b) have investigated whether face-to-face conversation would affect the importance of the events. In a paper by Aizawa et al. (2004a), the authors emphasised that detecting a conversation’s importance benefits efficient retrieval from a lifelog. This thesis will employ www.face.com, which is a technology platform with best-in-class facial recognition software to detect faces from photographs (Kotsiantis et al., 2007).

In previous research, authors have tried using phone call data to recognise people’s relationships (Quercia et al., 2010; Ye et al., 2009). For example, if a smartphone calls another smartphone rarely during working hours, but often in the evenings and during the weekend, then it is likely that these calls are from a friend or family, rather than a work colleague.

It is the combination of these three sets of information (i.e. Bluetooth, face detection and phone call) that gives a much stronger indication of the social connections to the people we encounter.

#### 2.6.1.6 Environment Context

Environment contexts are widely adopted in the context-aware field to indicate “what the user is doing”. The context-aware system can change device’s behaviours when the environment changes. The environment around a user such as temperature, noise and light is constantly changing. There has been initial research investigating how important the environment context can aid user recall of past experiences (Hirakawa, 2007). For example, Siewiorek et al (2003) have classified environmental noise level into three states: low, medium and high. A low value describes a quiet environment, whereas medium value identifies common situations such as talking and the high value states the environment such as a pub or bar. According to the environmental noise level, the system sets up the phone ring volume. In lifelogging, the environment context mainly includes that ambient temperature, ambient light brightness, environmental noise level and weather etc. To make the lifelog data more meaningful, Lee and Cho (2007) had considered collecting weather information. Doherty et al. (2007) described using the environmental data, i.e. the temperature and environmental noise level to segment lifelog data into events successfully.

### 2.6.2 Combining Multiple Data Streams

A sensor is a converter that measures a measurable attribute and converts it into a signal which can be read by an observer such as a computer or smartphone (Janardhan and Kumar, 2012; Majumder and Ray, 2012). For lifelogging users, most lifelog data is not easy to access directly because it can come from different sensors, each of which has a different data format and a different value range. There is no data management tool such as a search engine that can be applied directly on lifelog data because of a semantic gap between the user's query and the system data (Smeulders et al., 2000). This semantic gap indicates the lack of coincidence between the knowledge that sensor data can give a user and the actual meaning of the sensor data. Most of the sensor data, such as GPS, is in numeric form which does not support very well most conventional user queries. To enable user searching and exploration of lifelog data, the raw data needs to be transformed into semantic contextual annotations that the user can understand and use. For example, GPS needs to be transformed to an address or a meaningful place such as home.

Physical sensors are the most frequently used. These are hardware sensors and their function is usually very simple. This is because they are designed to be linear or linear to some simple mathematical function of the measurement (Janardhan and Kumar, 2012). They simply collect data; the consoles will read data with specific protocols and store the data using a specific data format (Majumder and Ray, 2012). However, physical sensor data is not typically designed to be human-readable, but in a form suitable for transmission and processing (Albertos and Goodwin, 2002). In this thesis, when sensors are mentioned, they mean physical sensors from which most lifelog data is collected.

Because the outputs of different physical sensors are in different formats, a way must be found to fuse them in order to acquire a united interface. To address this problem, we will introduce the concept of virtual sensor in Chapter 4.

Generally, the virtual sensor is defined as a software sensor which has three functions: 1) combining different format data collected from multiple psychical sensors; 2) transforming the combined different format data into a natural language which could be easily understood by humans; and 3) sharing the data through the web. Virtual sensors provide indirect measurements of abstract conditions by combining sensed data from multiple physical sensors, but can be used just like a physical sensor (Costantini and Susstrunk, 2004; Kabadayi et al., 2006). As mentioned above, the outputs of virtual sensors are in natural language which can be understood by humans (Lu et al., 2010). In addition, it enables researchers with different technical background to dig the data. For example, a researcher in healthcare can explore the link between sitting length and obesity from the virtual sensors outputs when a researcher in computing science shares data. The current data sharing function is less developed. A preliminary study was conducted by Lu et al. (2010) where they implemented a virtual sensor to export the outputs which can be used by other application on the phone. However, the outputs are not easily accessed by other researchers. Another study was conducted by Kim (2011) who used web services to share the outputs from the virtual sensor. Kim's (2011) work enables researchers to collaborate via the Web, but it is only open to researchers who have a background in programming languages such as SQL to make enquiries. This study will implement a set of virtual (software) sensors to enrich the raw sensor streams with semantically meaningful annotations. In addition, we implement the data sharing function through the web. More details will be discussed in Chapter 4.

## 2.6.3 Segmenting Events

### 2.6.3.1 What is an Event?

The definition of an event varies among research groups (Zacks and Tversky, 2001). According to Zacks and Tversky (2001: 3) *"an event is a segment of time at a given location that is conceived by an observer to have a beginning and an end".* However, the timescale of an event could be from a few seconds to a few hours. For example, picking up a pen from the ground is an event while a four-hour drive can also be considered as an event. Furthermore, an event is also conceived as having a hierarchical structure, as being composed of parts, or subevents (Hard et al., 2006).

Events depend on context and occur at different granularities or resolutions (Jain, 2008). When asking people to describe what they did during the day, they might list a few of events, such as *"I got up at seven, then prepared my breakfast. After I had my breakfast, I began to drive to work..."*. Such sentences are typical descriptions of events. People segment the complex dynamic world and the continuous flow of lifelog information into a modest number of meaningful units (events) (Kurby and Zacks, 2008). To discover what an event is, one approach is to ask for users' opinions directly (Newtson, 1973). Following this instruction, Zacks and Swallow (2007) asked users to watch short movies of single actors performing everyday activities and to segment them into events by pressing a button whenever they believed a boundary occurred. Although Zacks and Swallow (2007) did not give users any special instructions, the segmentation results were remarkably consistent. Observers agreed with each other on the event boundaries. Inspired by this, the current study attempts to automatically locate the event boundaries by training dataset using support vector machine learning (SVM) technique based on human decision boundaries.

#### 2.6.3.2 Why Use Events?

Since there is typically a huge amount of data in a lifelog, it is unlikely that individuals can organise every piece of information and remember where they put it. An organisational structure is needed to summarise the data into meaningful chunks of information for a user to read. To achieve this, it is necessary to understand how people organise information and items in their memory.

Studies on autobiographical memory have suggested that we organise our experiences into events. Brown and Schopflocher (1998) developed an event-cueing technique to explore the nested structure. They found that clusters of events people recall together are usually causally related, temporally adjacent, or similar in content. Kurby and Zacks (2008) found that people observe the complex dynamic world in terms of events. They also found that organising ongoing activity into events will integrate information over the recent past to improve predictions about the near future. Therefore, the context-free event segmentation used in lifelogs is a harder problem as it is unguided. It is noted that unless otherwise stated, the author is using the context-free, i.e. the 'artificial', non-psychological sense of 'event'. The motivations for using event segmentation are summarised as follows.

- **An event is a natural unit for human memory.**

  Zacks et al. (2006: 466), who studied how representation in the brain works, states that humans store memories as events: *"... segmenting ongoing activity into events is important for later memory of those activities... "*. This research suggests that human event segmentation is context-related, whereas lifelog event segmentation is context-free.

- **An event is a reasonable unit for lifelog data management.**

  The lifelog dataset consists of many types of sensor data that are continuous and lack natural breakpoints. Lifelog data must be organised into logical and meaningful units. For example, an image browser uses one image as one unit. A search engine considers the document as its unit. This work on lifelog typically considers the event as the unit of retrieval.

- **There is no major context changing in one event.**

  Newtson et al. (1977: 858) noted that *"... breakpoints [between events] tend to correspond to points at which the most physical features of the action are changing..."*. Therefore the most natural way to segment an event is to find the breakpoints. When some context changes, it will generate a breakpoint. Accordingly the big change usually happens at the breakpoint which separates two events. A single event usually shows little or no change in major.

#### 2.6.3.3 Approach to Event Segmentation

Obeid et al. (2010: 132) noted the following about events.

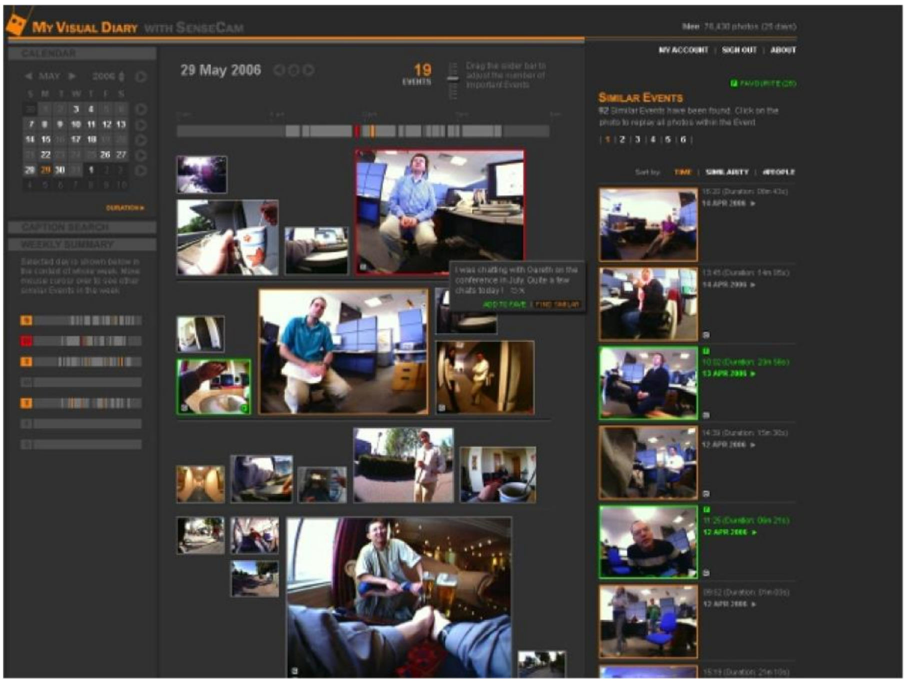
> *"we take events to be associated with patterns of change. An event must involve at least one object over some stretch of time or involve at least one change of state…"*

In the lifelogging domain, Loui and Savakis (2000) mentioned segmenting users' album photographs using date and time. Additionally, the images content could be considered if the image's date and time are not available. Lin and Hauptmann (2006) described a *k*-means clustering algorithm to cluster images together using their visual features. A weakness of this approach is that the number of events/clusters must be fixed to a value *k*, but in the real life, the number of everyday events can vary from day to day (Doherty and Smeaton, 2008*a*). Furthermore, this algorithm couldn't recognise the boundaries of events. Wang et al. (2006) segmented their lifelog video into five-minute clips, however the real life could not be segmented based on a fixed time length; we need something more flexible than a fixed duration or fixed number of event segmentation approaches.

Considering change of context has been used to segment events. For example, Cheng et al. (2010) segmented lifelog data by detecting peaks of activity change. Osada and Yoshino (2012) segmented an event using the change of address. In their research, Chen et al. (2011) segmented a long-term lifelog into events; but they mostly focused on grouping lifelog data into events using the Textilling algorithm without considering the continuity of lifelogs. Segmenting events based on one single data source is not the only approach. Doherty et al. (2011) separated a day's images into some event groups using the context from photographs and also used a combination of accelerometer, light, passive infrared, and temperature sources.

Figure 2.2 shows a first generation WWW interface to a large SenseCam image archive that implements such an event segmentation approach.

**Figure 2.2: An event-based browsing tool for visual SenseCam archives**




Source: Doherty, Moulin and Smeaton (2011)

In their experiments, Doherty et al. (2011) only considered changes in visual features. In real life, some activities could not be easily segmented. For example, when a user is driving, the landscape may change dynamically and constantly. Obviously, the driving activity would be segmented into many events using image data only because the similarity of adjacent images is very low when the user is driving. In their study, Smith et al. (2011) talked about using location to segment lifelog data into events, such as home and the work place. Inspired by this, the current research considers the change in every physical and virtual sensor as a source data to segment events.

Based on the above description, we hypothesise the following:

***Hypothesis 1: Event segmentation can be performed effectively by detecting the changes in sensor data.***

#### 2.6.3.4 Keyframe Selection

After events are segmented from lifelog data streams, each event may contain numerous pictures and hence users are therefore still faced with large volumes of data. One approach is to select one most relevant picture representing the event, called "keyframe selection" based on the histogram intersection distance (Doherty et al., 2008).

$$H(X,Y) = \sum_{i=1}^{n} \min(X^{(i)}, Y^{(i)}) \qquad \textbf{(Equation 2.1)}$$

where $X$ and $Y$ are histograms with $n$ attributes, and $X^{(i)}$ denotes the count of the $i^{th}$ attribute of $X$.

Keyframe selection is a very common task in the video analysis domain. However in the lifelogging domain, there are very few researchers working in this area. Six keyframe selection methods were investigated by Doherty et al. (2008): 1) middle image, 2) within event, i.e. the closest to all the other images in the event; 3) cross event, i.e. the closest to all the other images in the given event, but also is most distinct (histogram intersection distance, see Equation 2.1) from all the other images in the other events (Grauman and Darrell, 2005); 4) image with the highest quality (the contrast and salience quality measures); 5) within event and image quality fusion, i.e. the closest to all other images, and also has a good quality score (the contrast and salience quality measures); and 6) cross event and image quality fusion, i.e. with a good image quality (the contrast and salience quality measures), and also the most distinguishable from the keyframes in the other events.

In those six approaches, the image quality method achieved the best results (Doherty et al., 2008). Therefore, the method of image quality is adopted in this study. One limitation in the existing keyframe selection is that researchers only considered the image itself but did not consider the effect of other contexts such as the social context. The social context is an important indicator for keyframe. For example, Doherty and Smeaton (2008b) found that the photographs with faces are identified by users to represent important events. Based on this idea, this study investigates the effect of different contexts (i.e. social, location and environment) on keyframe selection as described in Chapter 5.

## 2.6.4 Generating Narratives

### 2.6.4.1 What is a Narrative?

"*When somebody tells you his life... it is a narrative achievement*" (Bruner, 2004: 692-693). Hardy (1968: 5) stated the relationship between life and narrative as *"we* *dream in narrative, daydream in narrative, remember, anticipate, hope, despair, believe, doubt, plan, revise, criticize, construct, gossip, learn, hate, and love by narrative".* As described by Riedl and Young (2004: 5), "*A narrative is a sequence of events that describes how the story world changes over time*".

Indeed, narrative is one of most important approaches to spreading knowledge (Carr, 1986; Tuffield et al., 2005). From the earliest times, humans began to transmit knowledge as narrative, such as telling stories or cave writing. Narrative is not only a combination of concepts, as Niehaus and Yong (2009: 75) stated, but "*rely on the readers to use narrative conventions and reasoning to complete their understanding*" and thereby bring their own meaning into the topic. Biological research shows that reading a narrative activates more brain regions than reading just words and sentences (Xu et al., 2005). Furthermore narratives can also be used to support keyword text searches using information retrieval techniques (Chen, 2009; Jaimes et al., 2004), which is the natural method that users have to locate knowledge using WWW search engines.

Our life is a sequence of events and actions. Everything we see, learn, and do becomes part of a story. This is the way we learn about the world around us and this is the way we should present our knowledge (Schank, 2000). A narrative may thus be seen as a way of presenting the captured events using all contexts of what had happened (Fatah gen Schieck et al., 2003).

In the lifelogging domain, some researchers have noticed the importance of generating narratives from lifelog data. In the existing research into lifelog, the textual annotation has been used to describe each event (Aizawa, 2005). As indicated by Xu et al. (2005), narratives are known to have cognitive processing advantages when compared with annotations.

In a more recent study, Niehaus and Young (2009) mentioned that the user would like to add information in order to complete his understanding by using inferences when reading narratives. With these inferences, the facts and events presented follow a smoother logic, and they seem much more cohesive. To the reader, narrative is not only the combination of annotations; but more than that; narrative can give more information to users when they read it by adding more relative knowledge.

#### 2.6.4.2 Approach to Narrative Generation

Narrative has shown potential to help people recall their lives; however, users do not want to spend time editing or authoring their narratives, if there is no tool to aid their efforts (Appan et al., 2004). For lifelogging, the narrative should describe how the life experience has progressed and changed over time. Rather than being a sequence of '*I did < something > at that < time >', 'I did < something else > at the < time > at that place',* the narrative generation process should aim to represent life experience in a more natural manner, for example, the '*I did something else at the next event time'* narrative could be represented more naturally as *'After arrival, I started to < something > at about < time >'*. The narrative generation in this study consists of three sub-processes, namely fabula, sjuzet, and discourse generation. The fabula[1] and sjuzet[2] are from Russian words and have been "*described by modem*

---

[1] Fabula: "фабула" means that the sequence of events, or history, as they apparently happened in the Story (McVeigh, 2008).

[2] Sjuzet: "сюжет" narrative ordering of the plot, or story itself (McVeigh, 2008).

*literary theorists as, respectively, the timeless and the sequenced aspects of story*" (Bruner, 2004: 694). In other words, the fabula is the raw material for a story and the sjuzet is how the story is told, i.e. the structure. According to Cheong and Young (2008), a fabula is a story world that includes all the events, characters, and situations in a story. In this study, fabula is a series of sentences based on the detected contexts and segmented events; sjuzet is a paragraph of narratives generated from the fabula without the repeated sentences; and discourse is a paragraph of narratives with an illustrated picture/keyframe taken during the event. A detailed example for fabula, sjuzet and discourse is provided in Table 6.1 in Chapter 6.

Narrative generation is a process that involves the selection of narrative content (the events that will be presented to an audience), ordering of narrative content, and presentation of narrative content through discourse (Riedl and Young, 2010). When describing a narrative approach to data representation, Harper et al. (2007: 3) concluded that *"the ability to juggle-up the narrative of life to create evocative stories was also a bonus".* Gemmell et al. (2005) described how to use many types of data collected by the MyLifeBits system to construct stories as a response to queries. With the additional captions of images and audio clips, the narrative can be generated as shown in Figure 2.3. In their experiments, Gemmell et al. (2005) mostly used time and location contexts, hence only simple narratives can be generated.

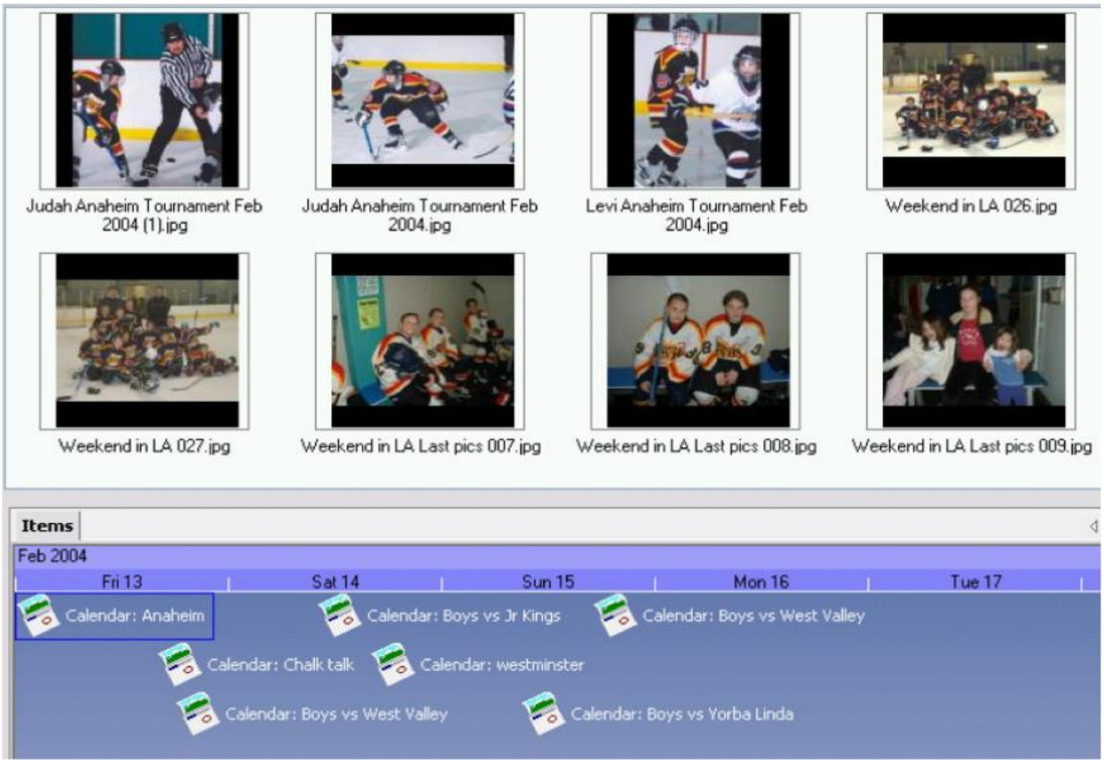
Additional research has taken place on generating narrative presentation on lifelogs. For example, Byrne et al. (2011) asked participants to manually choose multimedia data collected by lifelogging tools and to organise them to generate stories. Therefore, it is not real narrative generation but storytelling. In Hammerl et al. (2012), the narratives they generated were a list of activities ordered by time.

Inspired by and extended these existing studies, this study generates narratives based on multiple types of contexts and the segmented events. The details will be described in Chapter 6.

Based on the above description on the concept (what), advantage (why) and the process (how) of narratives, we propose the following hypothesis.

***Hypothesis 2: A meaningful textual narrative that accurately represents an event can be generated automatically.***

**Figure 2.3: The interface of telling stories with MyLifeBits**




Note: All images associated with a narrative are generated using time and location.

Source: Gemmell et al. (2005)

## 2.7 Displaying Results

With the development of new devices such as the smartphone, data representation on multiple devices has become an important research topic. However, the existing research mainly focuses on "universal accessing". For example, Hess et al. (2011) investigated the user requirements and preferences across device media systems. They argue that it is required for a universal access on different devices. The content needs to be shared and accessed between personal and shared devices. Fleury et al. (2012) reported an approach to transform video content from a mobile device to a television. Recent software solutions such as Apple's Bluetooth File Exchange allow for over-the-air transfer of content such as photographs or documents between paired (Source from Wikipedia).

More attention is needed to consider the differences in devices such as size of the screen and the availability of a keyboard. Ubiquitous computing scenarios not only enable the "ubiquitous accessing", but also bring a challenge for system and user interface designers. Different types of devices have different interaction expectation standards. For example, the user interface based on keyboard and mouse has become very successful among PC users. The user interface based on multi-touch screen allows intuitive ease in navigation and catches the fancy of millions of smartphone and tablet users (Tambe, 2012). It becomes increasingly difficult to optimally represent content across the myriad of devices currently available, due to different size, resolution, interaction mechanisms and environment of use.

Human-computer interaction (HCI) experts have realised the importance of characteristics of output devices on interface design (Robertson et al., 1996). As Tambe (2012: 24) stated in his work, "*Clearly, different products representing different paradigms require different solutions. When the needs and habits for a product are different in another medium, a paradigm shift is needed*". Lifelogs are made of different kinds of contexts. They can also be presented in different ways.

There might be no single optimal way to present lifelog data on every device. The best way to display lifelog data is using different representation techniques on different devices for different usage scenarios.

Therefore, we propose the following hypothesis.

***Hypothesis 3: Different access devices benefit from different representations of lifelog data.***

## 2.8 Summary

In this chapter, we reviewed the history of lifelogging and the current status of research in this area. In comparison with other research topics, lifelogging is quite a young and open research area. In the first stage of lifelogging research, most researchers focused on developing suitable lifelogging tools which needed less user effort and could be widely used in the wider population. Having been studied in other research (e.g. Doherty et al., 2011), the SenseCam could not achieve this goal. The arrival of smartphones not only changes the concept of a phone, but also gives a new hope to lifelogging research because of its online computation and networking capability. This motivates the current research to develop a new generation of lifelogging tool based on the Android smartphone which is one of most popular smartphone OS's. Furthermore, smartphone can be seamless integrated into daily life, is resource efficient, secure, and facilitates long-term digital preservation.

Because the data collected from many sensors are in different numerical formats, it needs to be transformed into meaningful and understandable contexts. In this chapter, previous research on context detection and event segmentation was reviewed. Six contexts are detected as personal, time, location, activity, social and environment.

Narrative is the most acceptable form of knowledge for humans and its advantages were documented in other domains. In this chapter, we reviewed the relevant literature on the definition of narrative (what), the reason to use it (why) and the approach to generate it (how).

In the last part of this chapter, we reviewed the existing work on how to display the lifelog data analysis results. Most of research focuses on the display in only type of device. Based on the unique characteristics of different devices, we propose that different access devices benefit from different representations of lifelog data.

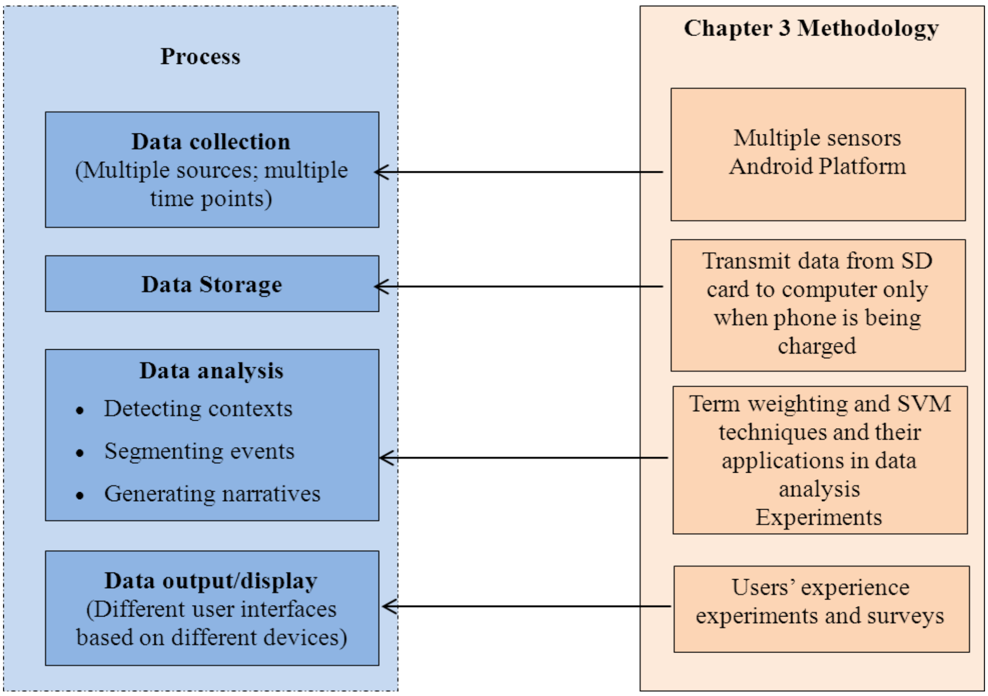
# CHAPTER THREE

# RESEARCH METHODOLOGY

## 3.1 Introduction

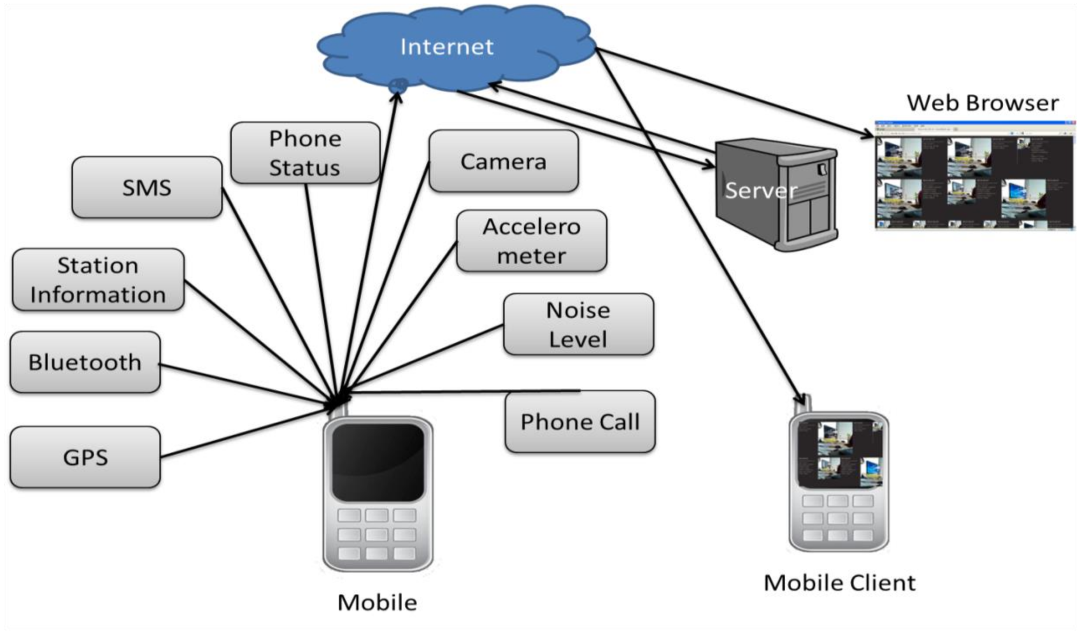
In this chapter, the methodology employed for different aspects of the research is presented. It firstly describes the data collection based on multiple sensors and the Android platform. Then the data analysis techniques such as term weighting and support vector machine learning (SVM) and their applications in data analysis are introduced. Finally, it presents the methodology employed for experiments in data analysis and data representation processes of the system. Figures 3.1, 3.2 and 3.3 show the main points addressed in this chapter, the overview of the proposed lifelogging system and its architecture.

**Figure 3.1: Work in Chapter 3**




Source: The author (2013)

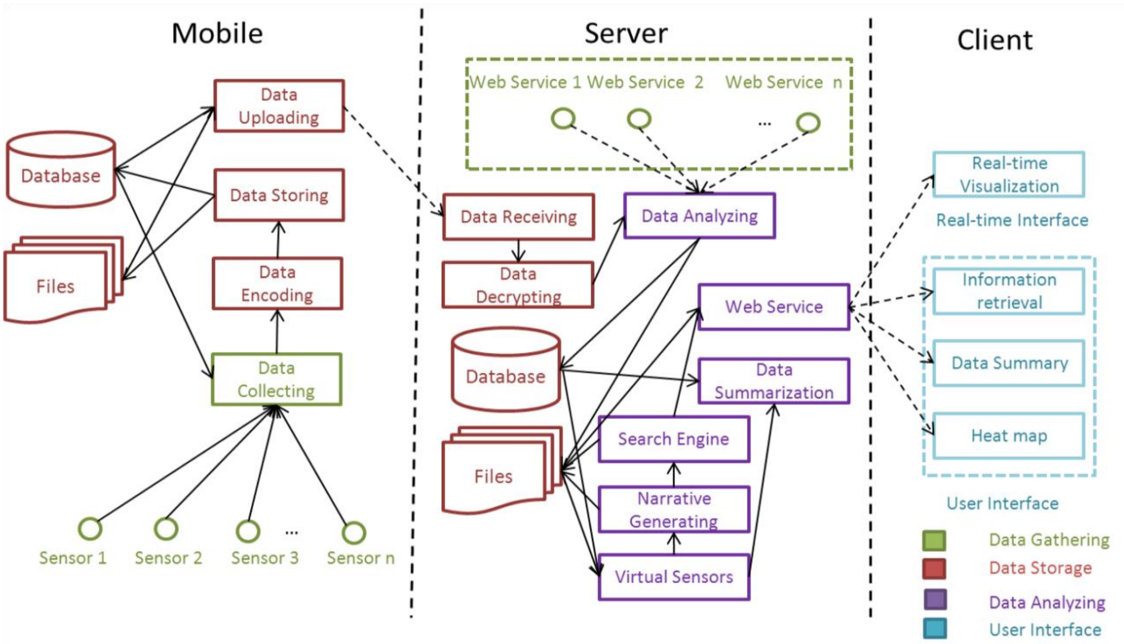
**Figure 3.2: The lifelogging system overview**




Note: The system can automatically collect data and upload it to a server. Users can access their data through a web browser.

Source: The author (2013)

**Figure 3.3: The lifelogging system architecture**




Source: The author (2013)

The lifelogging system developed in this study not only contains data collection but also includes data storage, data analysis and the user interface to represent the results. Figure 3.3 shows a new architecture for lifelogging systems. The data is collected by a smartphone and then sent to a server when the phone is being charged. The server reads and processes all the users' data. On the user side, data can be accessed using a web browser. Once the user is logged into the website, data can be accessed by browsing through information retrieval tools. The details are shown in the following subsections.

## 3.2 Lifelog Data Collection

In this study, lifelog data is collected from multiple sensors based on the Android platform, without requiring any user input.

### 3.2.1 Multiple Sensors

The smartphones used for data collection have multiple sensors. They include the accelerometer, GPS receiver, Bluetooth, and camera. These sensors are good data sources for researchers to investigate a user's activities. For example, a GPS receiver provides the user's location information; a camera provides picture/image information.

For data collection, the lifelogging system not only collects a user's personal information using physical sensors, but also collects related information such as weather information from the internet and a location gazetteer. With the location and time information, the weather information for the user can be found by some online weather service. All the data captured is shown in Table 3.1.

**Table 3.1: A full set of available sensors and default sampling rate**

| Sensor Name | Data | Granularity |
|---|---|---|
| Accelerometer | 3 Directions' Acceleration, Time | 5 readings/second |
| Bluetooth | MAC address, Name, Type, Time | Every 2 minutes |
| Base Station | Base Station ID, Country Code, Area Code, time | event-driven |
| Camera | Photograph, Time | 2 photographs /minute |
| GPS | Latitude, Longitude, Elevation, Time | Every 20 meters |
| Noise | Noise Level, Time | Every 30 seconds |
| Music | Track name, Artiest name, Time | event-driven |
| Phone Call | Phone Number, Start Time, End Time, Type | event-driven |
| Power Status | Charging Status, Time | event-driven |
| Screen Status | Screen on/off | event-driven |
| SMS | Phone Number, Time, Send/Receive | event-driven |
| Taken Photograph* | Photograph, Time | event-driven |
| WiFi | MAC Address, Name, Signal Strength, Time | Every 2 minutes |
| Weather | Weather, Distance (km), Cloud (%), Temperature(℃), Discomfort index (%), Rainfall (mm), Snowfall (cm), Humidity (%),Wind direction, Wind speed(m/s), Sea-level pressure (hPa) | Every 30 minutes |

Note: The frequency can be modified using a configuration file. * indicates that the system records the user's photograph taken event.

### 3.2.2 Android Smartphone Platform

In this study, the lifelogging system is developed based on the Android smartphone platform. Google releases the Android code as open-source under the Apache License. Any smartphone complying with the license requirements can run the Android system. Because it is a customer-based virtual machine, the same app can be used on smartphones with different versions of the Android operating system. It means that the lifelogging tool can be deployed on any smartphone which is running the Android system. Based on the Android platform, the proposed lifelogging system has the following features.

- **It does not disturb the user's interaction.**

  The system runs like a background service. It has no interface. Once the user installs and finishes the configuration, it can be started automatically when a user turns the smartphone on. It can detect screen status. It will stop collecting data and release related the resource when the screen is on or when the user is on a phone call. When the phone's screen is off, it will awake and run in the foreground to maximise use of the phone's resource.

- **It does not require the user to organise lifelog data.**

  It doesn't ask the user to organise data manually. It uploads and organises data automatically when the smartphone is being charged. Because most lifelogging users are not experts on computing, they can easily make mistakes which affect the data quality. On the other hand, updating and organising data can be boring and time consuming work, due to which the users may stop using the lifelogging tool. The automatic organisation of data solves this issue.

- **It is context aware.**

  Collecting more data means consuming more battery resources. For example, if a smartphone keeps the GPS on, the battery can only work for 5 to 8 hours. Collecting more data also means using more bandwidth and processor time when it sends data back to the server. It also costs more, in terms of battery life because wireless networking consumes a significant amount of battery resources during data transmission. Furthermore, additional data increases the amount of data which needs to be analysed. The analysis processing, such as using SVM machine learning to segment events, takes a longer time when more data is applied.

In the proposed lifelogging system, we only begin to collect some information when it is necessary. For example, it the GPS sensor will be turned off automatically when a user sits down to have coffee, and turn the GPS receiver on when a user leaves the coffee shop. It can also learn the user's location pattern automatically and judge if it is necessary to turn on the GPS sensor. It can reduce sampling frequency and extend battery life yet retain the same detailed lifelog capture. It can choose an optional sensor to collect data if there are more than one data source. For example, if WiFi and GPS are available in the area, it will choose WiFi as a location source when the WiFi's location is known in the system. Using WiFi for location information would consume less battery life and return a comparably accurate result.

- **It implements data security.**

It has higher data security since all data is encrypted on both the smartphone and server sides. When the system collects some sensitive data such as phone numbers, it encrypts the data using the password-based encryption (PBE) method (Ab Rahim, 2004). For photographs, the system will not encrypt whole image files, but only the header of the JPG file (Huang et al., 2009). On the other hand, a user needs a password to access the file via a web browser or user application. The password can be set up or changed only when using the smartphone. In our system, we use the SIM card series number as the encryption key. This ensures that the user will not lose any information, even when he or she changes their phone. The user can also access their data on any Android smartphone with the same SIM card.

- **It does not require specific hardware.**

  The software can run on any device which is installed with Android later than version 2.2. It can automatically detect available sensors. If some sensors are not available, the system can also run normally, simply not gathering information from the sensors.

- **There is no time gap among sensors.**

  There is no time gap among the sensors. A globally synchronized clock is crucial when several sensors work together (Martincic and Schwiebert, 2005). The previous lifelogging tools were implemented with different devices working independently of each other. For example, in previous research, a SenseCam was used to capture photographs, and a GPS receiver was used to collect location information (Gemmell et al., 2005). However, the devices used different system clocks. There is no effective way to synchronize the time of different devices. In addition, SenseCam will lose the time setting when it is out of power. This makes it very difficult to integrate location and visual data (Gemmell et al., 2005). In this work, all the sensor data collected are timed with one clock. The system also records satellite time when it captures GPS data. It is very easy to identify any data with the wrong timestamp by comparing the two timestamps. The server can also correct data using satellite time automatically when data is uploaded.

- **It ensures longer battery life.**

  As reviewed in Chapter 2, the smartphone is an appropriate lifelogging data collection tool. However, in past research, very few successful lifelogging applications have been developed on the smartphone as it still not feasible due to considerable battery limitations (Doherty et al., 2011). The proposed lifelogging system employs a clever power management strategy (i.e. context aware). The smartphone's battery can work for at least a full day. It can even work longer after it has learned the user's regular life pattern.

- **It supports real time mode.**

  Real time networking is getting more and more popular. People are very interested in sharing information in real time. If a user activates real time mode on, the software will detect the important moments and upload to the server immediately. For example, if a face is detected from a photograph, the photograph will be uploaded to the server immediately. Other related information such as location and environmental noise level will also be sent. In this study, to extend the battery life, the system is not running in the real-time mode, but it can be turned on easily when necessary.

- **It is extensible.**

  In the future, more new sensors may be integrated into smartphones. In order to deal with that, the software in this project implements a flexible architecture based on the Android API which can automatically detect a new sensor and read its data. On the server side, a flexible framework which can identify sensor type with the header of the data file is also implemented. For example, the current experimental smartphone does not have a light sensor and does not collect light intensity information. However, when the system is installed on a smartphone with a light sensor, the system will automatically find the light sensor and collect light level information. When the server receives the data file, it will read the header of the file in order to discover the sensor type and decide where the data will be stored. If it is a new sensor, the system will create a new table for the new sensor automatically.

## 3.3 Lifelog Data Storage

The battery and storage are among the smartphones' most important resources. Even though most smartphones support external SD storage cards, they have quite small memory compared with a computer's storage. The collected data could not have been sent to the server immediately, because access to a wireless network is not available all the time and everywhere for transmitting data. In addition, using wireless for data transmission has a high cost in terms of battery consumption. Therefore, we design the system to store all sensor data on the phone SD card temporarily and transmit it to the server only when the smartphone is being charged. By doing so, the system ensures enough battery usage for data transmission.

Furthermore, this system reduces the data file size which helps to avoid large volume data loss when the wireless network becomes disconnected during uploading. For instance, this study didn't adopt any of the most commonly used data formats such as XML and JSON, because XML and JSON tags waste a lot of storage and bandwidth (Wang, 2011). Another reason is that photographs are binary files and they can’t be stored using XML or JSON data format file. The data structure of our system is shown in Figure 3.4.

**Figure 3.4: Data file structure**

```
UUID-File Type- Device ID – Simcard ID
Timestamp,, Attribute 1,, Attribute 2…
Timestamp,, Attribute 1,, Attribute 2…
Timestamp,, Attribute 1,, Attribute 2…
Timestamp,, Attribute 1,, Attribute 2…
```

Note: The accelerometer data file contains 30 readings, each line containing one time-stamp. The server will associate a timestamp with each reading when it is read.
Source: The author (2013)

The data file contains “UUID”, “Data Type”, “Device ID” and “Sim card ID” on the first line. The “UUID” is the unique ID of the file; it will be stored in the database when the file is read. The system does not read it again if the ID is already in the database. It thus avoids re-reading the data file under a multi-thread situation. “Data Type” can give system information about which sensor this file is generated from. The system decides to start related proceedings to process it, because it is made up of several services which can process different types of data files. “Device ID” and “Sim card ID” can be used to identify the smartphone and user. The remaining lines are data lines. For most kinds of data lines, each line contains one time-stamp and one reading of values. But for accelerometer data files, each line contains 30 readings and one time-stamp to save storage space. Because wireless is not stable as a wire data transmission method, it is easy to lose data. This study sets a maximum data file size of 200k for photograph files, because it decreases the data risks during data transmission. Once the data file reads the limitation, it creates a new file and encrypts the old file into binary format. All new data is then written into the new file.

Normally, the smartphone generates about 300Mb of data every day in gathering data. A smartphone with a 4 GB SD card can store about 12 days' lifelog data. With regard to battery life, the system does not upload all data in real time. It only uploads data when the phone is being charged. At the same time, it sorts all data files by time and sends them to the server. When the data integration file is checked, the server will send back a flag which confirms the success of transmission. Then the system will remove the file from the SD card. The detail of pseudocodes on transmitting data from smartphone to server at both phone and server sides are shown in Figures 3.5 (phone side) and 3.6 (server side).

**Figure 3.5: Pseudocode for data transmission from phone to server (phone side)**

```
WHILE true
        Let system sleep for 1 second
        SET POWER_STATUS= get charging status
        IF POWER_STATUS THEN
                SET NETWORK_STATUS= Wifi status
                IF NETWORK_STATUS THEN
                        List all the data file by time
                        SET fileCount to 0
                        SET fileNumber to the number of data files
                        FOR i=0 to fileNumber
                                Post file to the web service with fileType and SimcardID,
                                return true if it is sent to server and delete the data file
                                from SDcard
                        Next
                END IF
        END IF
END WHILE
```

Source: The author (2013)

## 3.4 Lifelog Data Analysis

The lifelogging system in this study not only collects data, it also gives feedback to users which might be of benefit to them. However, the bulk of the data collected by the system is numerical such as latitude and longitude which does not make sense to the users.

**Figure 3.6: Pseudocode for data transmission from phone to server (server side)**

```
SET fileType= getPostParameter(fileType)
SET simcardID=getPostParameter(simcardID)
SET dataFile=getPostFile()
SET fileHead=read(dataFile)
SET fileID= getPhoneID(fileHead)
SET userID= getUserIDFromDatabase(simcardID)
IF fileID in database THEN
        Delete the file and end function
END IF
SET fileIntergrity= checkFileIntergrity(dataFile)
IF fileIntergrity is false THEN
        Send fail flag to client and end function
END IF
SWITCH(fileType)
        CASE imageFile:
                Decrypt the file header with simcardID
                Write the file path in database
                Move the image to the image folder
                BREAK
        CASE sensitiveFile:
                Decrypt the file header with simcardID
                BREAK
        DEFAULT:

END SWITCH
Send a success flag to client and end function
```

Source: The author (2013)

To enable the user to view this information, such data should be something that users can understand. In our system, the server receives and stores data sent by the phone side. It also collects additional information (such as weather information) from the internet. It translates raw sensor data into the contexts which are understandable by the user. For example, it translates the latitude and longitude to the address, and defines the semantic meaning of the address for the specific user, such as “home” and “workplace”.

The data analysis component is based on the server side which includes detecting contexts, segmenting events, and generating narratives which will be introduced in more detail in Chapters 4 to 6. The main functions are:

- **Decoding and verifying data file:** The data files sent back by the smartphone may not be integral due to wireless failure. This component would decode the entire received file and verify them. If the file is complete, it will be stored in the database or in a specific format file. A flag indicates the successful transmission will be returned or it will remove the failed file and send an error message to the phone user/client.
- **Collecting external data:** In some cases, external data sources can provide valuable information. With a user's time and location information, the server can collect information such as weather information from on-line services.
- **Segmenting events:** Typically, in a full day, a person encounters more than 20 individual events, with each event lasting 30 minutes on average (Doherty and Smeaton, 2008*a*). This work employs the machine learning approach to identify events from sensor streams by detecting changes of contexts.
- **Semantic analysis software (virtual sensor) for sensor streams:** The output of the sensors consists of raw sensor streams. To support real-time analysis, semantic analysis tools are needed at both server and smartphone sides. These act as virtual (software) sensors to enrich the raw sensor streams with semantically meaningful annotations. For example, using raw accelerometer values, we can identify the physical activities of a user, e.g. walking and driving (Qiu et al., 2011). The following virtual sensors are used: semantic date/time, meaningful location, personal physical activity, social interactions, environment context, semantic visual concepts automatically identified from the photographs and personal context of the user's life patterns. Using these sources, we can

semantically enrich the annotations of events and construct a narrative to describe each event needed for both search and presentation.

- **Indexing and retrieving events:** In order to retrieve life experiences, the users' experiences and their annotations are indexed. This work employs an off-the-shelf search engine to index the narratives for every piece of life experience data. It also provides keyword search through the e-memory archive, ranking and presenting the multimedia life experience data to the user through a web interface.

## 3.4.1 Data Analysis Techniques

In this study, a large quantity of sensor data, also referred to as data streams, was collected by participants, most of which was not readable by humans. To mine and extract useful information from data, term weighting and support vector machine (SVM) techniques were employed in this study. Term weighting approaches are widely used in information retrieval to help users to find the highest related document according to their query. Term weighting occurs through analysing the statistical occurrence of terms in the natural language as well as in the document itself. SVM is one of most commonly used classification tools in the data mining area (Quackenbush, 2001).

### 3.4.1.1 Term Weighting

It is noted that this subsection on term weighting is mostly based on one of our papers as Qiu et al. (2010) which is an output of author's PhD studies.

In text retrieval, one of the initial challenges faced by researchers in the field was how to identify the most important terms in a piece of text and in a language as a whole. Finding a solution to this problem would allow for retrieval of ranked lists of documents and not just sets of documents based on Boolean logic. The solution lies in the work of Luhn (1958) which claims that the frequency of word occurrence in an article furnishes a useful measurement of word significance. This means that term frequency information from both documents and language as a whole can be used to identify and weight highly the important terms in a language.

This is achieved using various approaches to term weighting, with the most well-known being the TF*IDF ranking technique (Robertson et al., 1997).

- TF refers to Term Frequency. It's a measure of how important a term is to a document by simply counting its frequency of occurrence in the document.
- IDF is a global document collection score that identifies how important the term is to the document collection as a whole.
- TF*IDF ranking associates term importance weights with terms in documents by employing two term frequency components;
- TF and IDF. The more a term occurs across all documents, the less discriminating it is as a query term and the lower its IDF value is. Consequently the less a term occurs the more discriminating it is and hence the more desirable it is as an aspect of document ranking and so it will have a higher IDF value.
- IDF is basically the inverse of a score called DF (Document Frequency) which is a count of the number of documents that a term occurs in.
- TF*IDF weighting allows for the calculation of a term importance weight for the occurrence of a unique term in a document and is calculated using the following equation:

$$w_{ij}=tf_{ij}*log\left[\frac{N}{df_j}\right]$$ (**Equation 3.1**)

where: $w_{ij}$ = the weight assigned to a term$T_j$ in a document $D_i$., $tf_{ij}$ = frequency of term $T_j$ in document $D_i$, N = number of documents in collection and $df_j$ = number of documents where term $T_j$ occurs at least once.

The conjecture is that, since the distributions of locations (a naturally occurring phenomena) follows a similar distribution law to the distribution of words in natural language, it should be possible to utilise term weighting techniques, such as TF*IDF to automatically identify important locations in a person's location log (Mitzenmacher, 2004).

It is however noted that where location log analysis differs from text IR is that in text IR, the least useful words are the words that occur most often, for example 'and' and 'the'. In location log analysis the most frequently occurring terms are likely to be the most important places in a person's life i.e. home and work locations. In addition, for this research the concept of a document needs to be defined in terms of location logs. It is assumed that a document is a month of location logs. Identifying an individual trip unit as a document does not make sense because a trip will typically not contain an extended time period anchored in one location.

#### 3.4.1.2 Support Vector Machine Learning (SVM)

It is noted that this subsection on SVM is mostly from one of our papers as Qiu et al., (2011) which is an output of author's PhD studies.

In this study, some tasks could be seen as a classification process. For example, event segmentation could be seen as identifying event boundaries from all units.

There are only two classes in the dataset, event boundary and no-boundary. For context detection, it can be seen as the process of selecting certain data units containing the context attributes. In real life, when people select a good apple to eat, they usually select the ripe or nearly ripe apple with full colour, no discoloration, good shape and smooth peel, because their experience tells them that such an apple tastes good. For a computer, selecting a good apple is a simple task as well, if the apple's colour, shape and smoothness could be described in a suitable way, such as by numbers which can be processed by the computer. For example, the apple could be labelled as a good one using some thresholds if its colour C $\in [C_m \ldots C_n]$, shape S $\in [S_m \ldots S_n]$, and its smoothness P $\in [P_m \ldots P_n]$.

However, in some areas, the process could be very complicated, such as using gene expression phenotype for identification and classification (Duan et al., 2005). Genes may have thousands of features; and identifying gene combinations to distinguish and separate the healthy patients from the sick ones could not be simply identified using some numbers, because even humans do not have such acknowledge. To address that issue, scientists tried to make the computer learn from data, so called machine learning. It is defined by Mitchell (1997: 2) as:

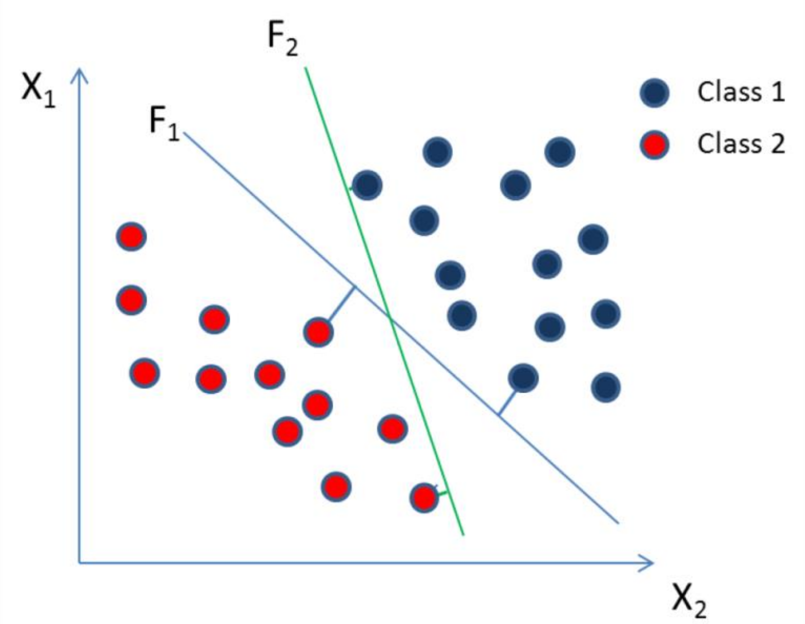
> "*A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E*".

Based on this theory, hundreds of machine learning algorithms are developed. SVM is one of them suitable for binary classification tasks (Tao et al., 2013).

**a. Classification Algorithm**

A SVM is a learning algorithm originally developed by Vapnik (1999). It is a technique suitable for binary classification tasks without loss of generality. One 2D example is shown in Figure 3.7, where several possible linear hyperplanes could be used to separate the two classes. For each two-class classification, an optimal hyperplane is searched that separates an *n*-dimensional feature space into two different classes: one class representing the category to be detected and the other one representing all other categories.

**Figure 3.7: An illustration of two hyperplanes ($F_1$ and $F_2$) which can identify two separate classes**




Note: $F_1$ and $F_2$ represent two separate classes: class 1 and class 2. The Hyperplane $F_1$ separates the data with the maximum margin. $F_2$ separates the data, but not with the maximum margin.

Source: The author (2013)

A perfectly separable hyperplane is considered optimal when the distance to the closest training samples is maximised for both classes. This distance is called the "margin". The margin is parameterised by the *support vectors* which are obtained during the training stage. By maximising the margin, we can search for the classification function that can most safely separate class 1 from class 2. For a two-class SVM, the decision function for a test sample x has the following equation:

$$g(x) = \sum_i \alpha_i y_i K(x_i, x) - b \quad \textbf{(Equation 3.2)}$$

Where, $\alpha_i$ is the learned weight of the training sample $x_i$. The training samples with $\alpha_i > 0$ are the so-called *support vectors*. $y_i$ is the class label of $x_i$ (+1 or -1); and $b$ is a learned threshold parameter; $K(x_i, x)$ is the response of a *kernel function* for the training sample $x_i$ and the test sample $x$, which measures the distance (or similarity) between the two data samples and maps the distance onto a higher dimensional space in which the hyperplane separator and its support vectors are obtained. Once the *support vectors* are known, the decision function for an unseen test sample $x$ is thus obtained.

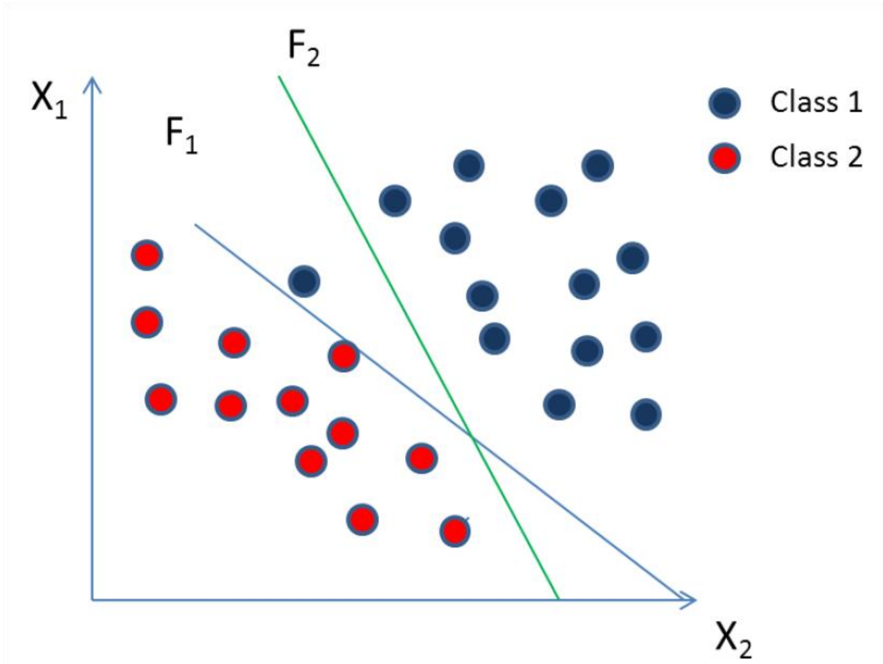
In some cases, the hyperplane cannot separate two classes perfectly. As shown in Figure 3.8, hyperplane $F_2$ cannot separate class 1 from class 2 completely; one class 1 point is misclassified; however, it has a better ability to generalise because it has a wider margin for both classes. In this case the margin is "soft". SVM maximises the margin width while minimising errors.

**b. Kernel Function**

The motivation behind mapping to a higher dimensional feature space (e.g. from 2D to 3D) is that this higher-dimensional space data could become more easily separated or better structured, referred to as the kernel function.

As shown in Figure 3.9, two classes in two dimensions can only be separated completely using a non-linear curve. However, when they are mapped to three dimensions, they can be separated simply by a linear hyperplane. The choice of an appropriate kernel function $K(x, y)$ is critical to the classification performance. $K$ should be positive, definite and symmetric (a.k.a. Mercer's condition) (Vapnik, 1999), to guarantee the convergence of SVM training.

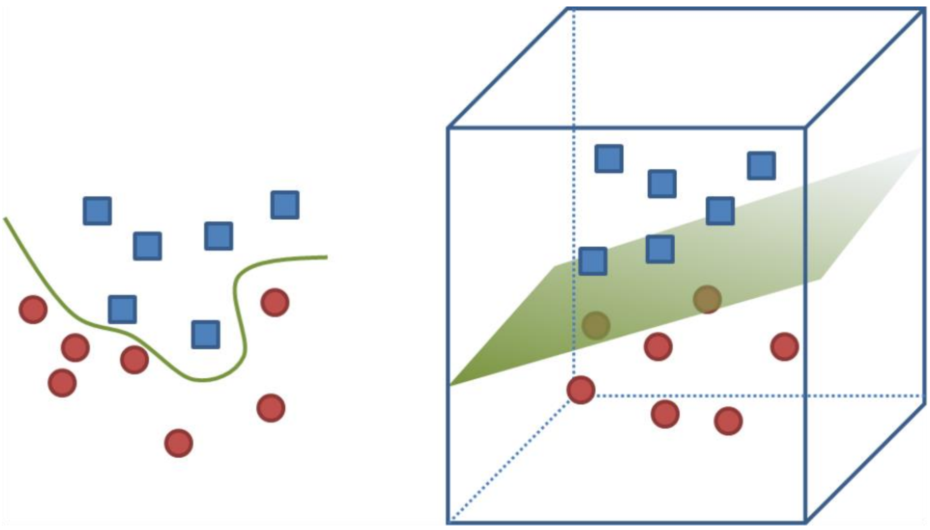
**Figure 3.8: An illustration of an imperfectly separable hyperplane *$F_2$***




Note: $F_1$ and $F_2$ separate two classes: class 1 and class 2. The Hyperplane $F_1$ has classified two classes completely without any classification errors, but it may cause over-fitting. $F_2$ has lower classification accuracy, but has the bigger margin than $F_1$. $F_1$ has high accuracy but not with the maximum margin. The margin of $F_2$ to the data point is "soft". It does not separated two classes completely.

Source: The author (2013)

**Figure 3.9: An illustration of how the kernel function works**



Note: In two dimensions, the two classes can only be separated by a curve. When two dimensional data points are mapped to three dimensions, the two classes can be linearly separated.

Source: The author (2013)

A number of general-purpose kernel functions have been based on the different distance metric of features. The most commonly used kernels are linear kernel, polynomial kernel, sigmoid kernel and gaussian radius basis function (RBF) (Zhang and Wang, 2011). This study employs the RBF kernel (see Equation 3.3) which is the most commonly used kernel function used in support vector machine (SVM) classification, because less parameters need to be set and the kernel values never go to infinity or zero even when the degree is large (Hsu et al., 2003).

$$K(x; y) = \exp(-\gamma \| x - y \|^2) \qquad \textbf{(Equation 3.3)}$$

Where $\gamma>0$, $r$ and $d$ are kernel parameters.

**c. Parameter Selection**

Usually, a SVM can work on its default setting. Some parameters can be set up to improve classification accuracy, to increase generalisation, and to avoid over-fitting (Auria and Moro, 2008)

In the above section, we have discussed the process used by SVM to find the optimal hyperplane by maximising the margin and minimising the overall risk. However, in some situations, maximising the margin may cause higher classification errors, while simply improving classification accuracy may reduce generalisation. In other words, the machine may work well on the training set but could perform inefficiently on a new sample. To address this issue, SVM models generally have a cost parameter, *C*, that controls the trade-off between allowing training errors and forcing rigid margins. The smaller *C* is, the wider the margin is, the higher the risk is in causing a classification error, and the higher the generalisation is.

Except for this global parameter, the RBF kernel has a specific parameter as $\gamma$. It defines how far the influence of a single training example reaches, with low values representing "far" and high values representing "close". If it is overestimated, the RBF kernel will work like a linear kernel. On the other hand, if underestimated, the function will lack of regularisation. For example, in the lifelogging system in this study, the decision on an event boundary will be highly sensitive to the environmental noise in training data.

The accuracy of a SVM model is largely dependent on the selection of the model parameters. To find the best combination of parameters, a common strategy is to separate the data set into two parts: a training part and a testing part. The prediction accuracy obtained from the testing part is used to ascertain the classification accuracy of an independent data set. This procedure is known as cross-validation. In *v*-fold cross-validation, one first randomly splits the training dataset into *v* disjoint subsets of equal sizes. Sequentially one subset is tested using the class trained on the remaining dataset. A model is trained *v* times using all the different subsets. The overall performance of that model is then calculated as the mean accuracy of the *v* classification runs (Hsu et al., 2003). The LibSVM software is adopted in this study. It provides a grid search which modifies the values of each parameter across the specified search range by using geometric steps to find the optimal parameters (Hsu et al., 2003).

**d. Advantages and Disadvantages of Using SVM for this Study**

According to Tao et al. (2013), support vector machine learning (SVM) is found to provide higher classification accuracies than other machine learning approaches. It has advantages in solving small sample learning, data nonlinearity and data high dimensionality problems mean that it has been used in a lot of classification research in recent years.

In this study, we choose SVM to classify our data, mostly due to the following five reasons:

- **SVM has relatively low sensitivity to the number of training samples.**

  SVM defines the classification model by exploiting the concept of margin maximisation (Melgani and Bruzzone, 2004). It appears to be especially advantageous in the presence of heterogeneous classes for which only few training samples are available. Because SVM can find the maximum gap between classes, most computational overhead resides in the training phase.

- **The same algorithm solves a variety of problems with little tuning.**

  By introducing the kernel, SVM provides flexibility in the choice of the threshold separating the instances (Auria and Moro, 2008).

- **SVM provides good out-of-sample generalisation.**

  Since the kernel implicitly contains a non-linear transformation (Auria & Moro, 2008), a good out-of-sample generalisation is provided by SVM if the parameters are appropriately chosen. This means that, by choosing an appropriate generalisation grade, SVM can be robust, even when the training sample has some bias (Auria and Moro, 2008).

- **The classification complexity in SVM does not depend on the feature space.**

  The performance of SVM is relatively insensitive to the number of data points and the classification complexity does not depend on the dimensionality of the feature space (Joachims, 1999).

- **The SVM is easy to use.**

  A lot of applications and libraries on SVM which are written in different programming languages are available online. There are a few parameters users need to deal with. For example, when the RBF kernel is used, the user only needs to adjust C and $\gamma$ to get the best accuracy. Some SVM applications such as LibSVM also provide tools to help the user find the optimal parameters automatically (Chang and Lin, 2011). Furthermore, SVM can be run on a smartphone (e.g. Zhao et al., 2011). This will reduce the effort required to move activity detection on phone side. It will decrease the server's load dynamically on processing lifelog data. Users will not depend on the server for information when the network is unavailable.

We acknowledge that SVM also has limitations which may affect the data analysis performance (Kotsiantis et al., 2007). For example, SVM has a slow speed in training data due to its cross-validation method, i.e. finding the best setting from a large range of potential ones. In addition, SVM methods are designed for binary classification, i.e. all of the results are true or false. To apply SVM in the case of multi-classification problems, users have to use a one-against-all strategy several times. SVM can solve a multi-classification problem but it is based on a single optimisation method where the parameters have low accuracy. However, these two limitations are not serious concerns. In our data, the training dataset is not extremely large. Indeed using a SVM does not require a great deal of time. In addition, we have found that our data only needs training once. The time is also not a serious issue in this study as the users do not need real-time processing for all data. We applied a one-against-all strategy rather than the multi-classification strategy which avoids the low accuracy issue.

We choose SVM to classify our data not because it is the best at machine learning, however there are no algorithms that can be the best on all the cases, but because we wanted to demonstrate that the machine learning technique can be applied in our case. Choosing a best algorithm from hundreds of classification algorithms and their variants would not be deemed so necessary in this study. The focus of this work is on utilising the outputs of the SVM, not on developing new or optimised machine learning approaches.

**e. Four Steps to Use SVM**

The process of using SVM to detect contexts include four steps; choosing training dataset, extracting the optional features of data, training the classification model, and evaluating the classification. These four steps are described in more detail below.

- **Step 1: Choosing training dataset**

  Typically, there are two catalogues of data in the training set: positive and negative instances. Some SVM tools support multi-labelled classification but are typically solved by combining independently produced binary classifiers (Weston and Watkins, 1999).

  To detect concepts from multimedia, a set of target concepts must be chosen. Usually, similar concepts should be avoided, such as lake and sea. For example, Cusano et al. (2004) defined seven concepts, which are quite different to each other. They are *buildings, ground, skin, sky, snow, vegetation* and *water*. Usually, machine learning research has assumed that the class distribution in the training data is reasonably balanced, because it has been observed that a disproportional abundance of negative examples decreases the performance of learning algorithms (Brank et al., 2003; Kubat and Matwin, 1997). For example, it is not suitable to identify "working on computer" and "checking email" activities in a lifelog.

- **Step 2: Extracting the optional attributes of data**

In order to classify instances or to separate people into two or more groups, we need to know some information about them. For example, "skin" may be the key attribute to classify apples into two groups: good and bad. They are not the only attribute that can be used. Other attributes such as water percentage could be a key attribute for choosing a good apple. Sometimes, some attributes will be used together. For example, a good apple should have a good skin and contain more water. However, to find optional attributes is very important.

The previous research has shown that the classification performance is determined by the way in which the attributes are selected (Rakotomamonjy, 2003). It seems that using more attributes provides more discriminating ability. With a finite training sample, a high-dimensional attribute space may be empty and many separators may perform well on the training data but few may generalize well (Bradley and Mangasarian, 1998).

Incorrect attributes may also bring more errors to the classification process. For example, to find a good apple the weight of an apple could be considered. Perhaps, co-incidentally in the training set all good apples are bigger than bad ones. Therefore size would be having more weight in the generated model. Obviously, such a model would not be the best model to classify good apples from bad ones.

- **Step 3: Training the classification model**

  In a lifelog, most concepts could not be detected using simple thresholds. The relation of attributes may not be linear. In general, the RBF kernel is a reasonable first option because it can handle the case when the relation between class labels and attributes is non-linear. There are two parameters for an RBF kernel: C and $\gamma$. It is not known beforehand which C and $\gamma$ are best for a given problem. To identify a good C and a good $\gamma$, cross-validation and grid-search would usually be used (Hsu et al., 2003). Some SVM libraries such as LibSVM usually provide the tools to search for optional parameters based on training data (Chang and Lin, 2011).

- **Step 4: Evaluating the classification**

  SVM usually evaluates itself using its training data with cross-validation. In $v$-fold cross-validation, a training set would be separated into $v$ subsets of equal sizes. Each subset is tested sequentially using the classifier trained on the remaining $v$-1 subsets. Thus, each instance of the whole training set is predicted once. The accuracy of cross-validation is calculated using a percentage of data which are correctly classified (Bradley and Mangasarian, 1998).

This study will improve lifelog information accessibility by using the above two techniques, i.e. the term weighting and support vector machine learning (SVM) on the three aspects. They are the location context detection using term weighting, activity context detection and event segmentation using SVM. More details on these aspects are described in Chapters 5 and 6.

### 3.4.2 Applications of the Term Weighting and SVM in Data Analysis

#### 3.4.2.1 Detecting Location Context using Term Weighting

It is noted that this subsection is mostly from one of our papers as Qiu et al. (2010) which is an output of author's PhD studies.

We used the term weighting technique to identify significant locations in this study. It is faster and less reliant on rules than other clustering based techniques. As mentioned earlier there are a number of components of a commonly used term weighting scheme such as TF*IDF. By employing the components of TF*IDF term weighting (TF, DF, IDF) four weighting techniques can be defined for identifying the important locations from personal location logs. In this study, the author was interested in locating a number of important location types:

- **Home/Work locations**

  The locations we most frequently visit, would be important for many locations applications. The volunteer user had purchased a new home, so was expected to locate in both homes.

- **Social locations**

  These are the locations that are most similar and the locations that we attend periodically, but not every day. It is expected that the system would be able to identify important social locations from the archive automatically. These social locations are those which the individual returns to again and again, such as family home, a relative's home and socialising locations.

- **Travel, extended visit locations**

  These locations are the places where the individual has spent some time, but visits do not reoccur so frequently. For example, holiday locations or work travel locations.

- **Pass-through locations**

  These locations are the places where we pass through often but rarely stop. These are exemplified by short linger/stay durations which occur frequently. An example of these locations is the shops that we pass through on our way to work every day.

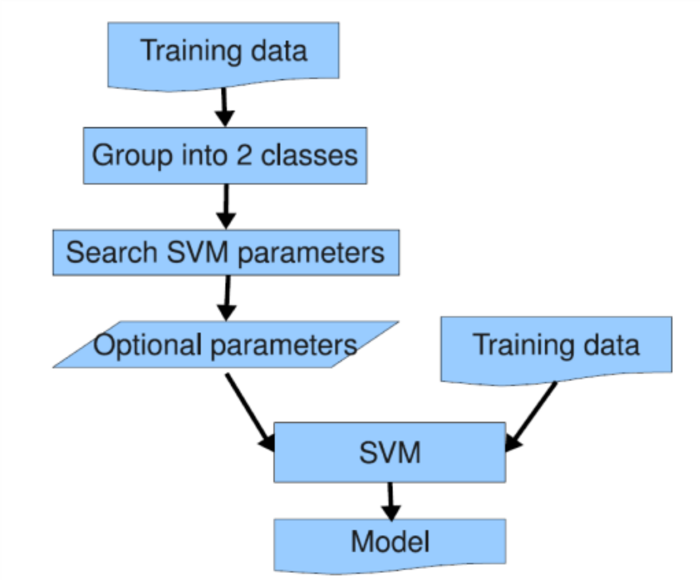
The location of user is important to classify users' activities. For example when a user is going to a shop, the activity is more than walking, but shopping. This experiment will combine the activity data and location data to get more detail of activities. In the experiments, the important location region and the time the user stays there will be considered to decide the important moments.

#### 3.4.2.2 Detecting Activity Context using SVM

It is noted that this subsection on the application of SVM in detecting activity context is mostly from one of our papers as Qiu et al. (2011) which is an output of author's PhD studies.

Four activity contexts detected automatically by the system are sitting/standing, lying, walking and driving. The steps for detecting activity context using SVM are shown in Figure 3.10. In the process, the training data is classified into two classes (binary classification) for each activity. Following that, the optimal parameters for each of the four activities are identified. The optimal parameters and training data are used to train the classification model for each activity. Each of the four models is then evaluated using five-fold cross-validation.

**Figure 3.10: Process of classifying raw acceleration data into user activities**




Source: Qiu, et al. (2011)

A number of attributes are used as input to the activity classifier, and these are described in detail below.

- **Raw acceleration data:** Raw data can be used to judge the posture of the mobile device. Due to gravity, the value of the accelerometer axis is about 1G. For example, when the user lies down, the horizontal axis' value decreases while the longitudinal axis' value increases.
- **Standard deviation:** This attribute is used to calculate the strength of activities. If the accelerations change rapidly, there is a strong likelihood that the user is walking or driving rather than sitting/standing or lying.
- **Range:** This attribute can be used to better distinguish driving from walking. When the user is driving, the *Standard Deviation* may be the same as walking.

However the range of values which change is smaller than for the walking activity. For example, when the user is walking, the maximum acceleration in the y direction can be 5 while it will be 3 when the user is driving.

Because accelerations were collected from a 3-axis accelerometer, a total of 9 attributes (raw acceleration data, standard deviation and ranges for each of the three axes) are used for one reading of acceleration.

#### 3.4.2.3 Segmenting Events using SVM

Following the four steps of SVM, the lifelogging system in this study segments the events from the raw data and the contexts extracted from sensors.

- **Step 1: Choosing training dataset**

  After collecting sensor data and uploading it to the server side, the participants were asked to annotate the event boundaries. The dataset with the annotated event boundaries is chosen to be the training dataset.

- **Step 2: Extracting the optional attributes of data**

  Based on the users' and researcher's own experience, some attributes are extracted from the contexts. Example attributes in this study are speed, signal strength change of WiFi hotpots, etc.

- **Step 3: Training the classification model**

  Lifelog data collected in this study was generated by different sensors. The attributes extracted from sensor data have very different value ranges. Before being used they must be standardised. Standardisation is a very important step before training data. The main advantage of standardisation is to avoid attributes in greater numeric ranges dominating those in smaller numeric ranges. After standardisation, all attributes are equal to the SVM. To identify a good C and a good $\gamma$, the two parameters for an RBF kernel we used the LibSVM to train our model.

- **Step 4: Evaluating the classification**

  SVM usually evaluates the training dataset itself. In *v*-fold cross-validation, a training set would be separated into *v* subsets of equal sizes. Each subset is tested sequentially using the classifier trained on the remaining *v*-1 subsets. In this study, we adopted five-fold cross-validation. To evaluate the effectiveness of SVM on event segmentation, three different metrics were used: precision, recall, and F1-Measure (i.e. a single measure that incorporates both precision and recall as defined in Section 5.6).

### 3.4.3 Approach to Generate Narrative

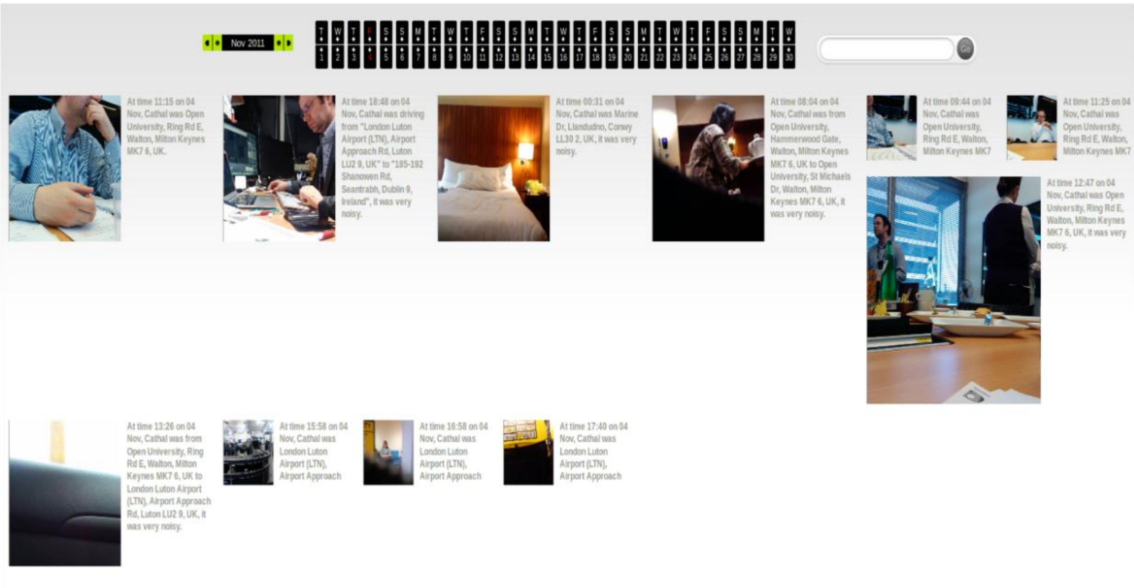
As Mateas and Senger (1999) suggested, narrative is a family resemblance concept, a cover term for a rich set of ideas expressed textually. Our model of an event follows from the fact that all data in a lifelog has relevance to real-life; the *"When"*, *"Where"*, *"Who"*, *"What"* and *"How"* to users. Therefore, to generate a narrative involves trying to answer those questions as accurately as possible, so as to accurately describe an event or a sequence of events.

As has been noted above, the narrative generation in this study consists of three sub-processes, fabula, sjuzet, and discourse generation. The fabula and sjuzet are from Russian words and have been "*described by modem literary theorists as, respectively, the timeless and the sequenced aspects of story*" (Bruner, 2004: 694). In other words, the fabula is the raw material for a story and the sjuzet is how the story is told, i.e. the structure. According to Cheong and Young (2008), a fabula is a story world that includes all the events, characters, and situations in a story. In this study, fabula is a series of sentences based on the detected contexts and segmented events; sjuzet is a paragraph of narratives generated from the fabula without the repeated sentences; and discourse is a paragraph of narratives with an illustrated picture/keyframe taken during the event. A detailed example for fabula, sjuzet and discourse is provided in Table 6.1 in Chapter 6.

## 3.5 Lifelog Data Representation

The user interface is one of the most important parts of the system for users. This study conducts research on finding the best user interface to show users' lifelog data on different devices. User interfaces (UIs) are provided to enable users to browse and search their lifelog data. One example of a UI is shown in Figure 3.11

**Figure 3.11: Web interface of event view**




Note: The uploaded data can be segmented into events by an event segmentation model, and the user can view them by events.

Source: The author (2013)

When lifelog data is uploaded to the server, the server will check data integrity, detect contexts, segment data into events, select a keyframe, generate a narrative for each event, and store all information in the database. Once those narratives are generated, the background service will index them. The user can access them using a web browser. Because the narratives are indexed by search engine, users can search them by query as well.

With the presence of mobile devices (e.g. smartphone), accessing digital data does not have to be taken place at a fixed place such as at home on a desktop. Users like to access data in any location and on different devices. Smartphones, digital TVs, E-book readers, hand-held tablets, etc. are part of the new generation of digital devices. They have the capability to access digital data even for a range of users such as the elderly people and young children. Compared with a computer or laptop, many of these portable devices are somewhat limited in computing power. On the other hand, they support different (not necessarily limited) input and output capabilities. Therefore, it cannot simply be assumed that the prior research on access to lifelogs, which was conducted mostly on computers, will be suitable to the new generation of devices. These devices support ubiquitous access, with different display capabilities and interaction methods. It is proposed that there is no one-for-all representation of lifelog data on different devices. Chapter 7 will explore what interaction methodologies work best across the range of modern computing devices. All of the semantics, segmentations and narratives generated to date in this evaluation are utilised. We tested the following three main categories of devices; the computer, smartphone and E-book reader.

## 3.6 Experiment Configuration

Lifelog study always involves privacy concerns (Allen, 2008). Even data that can only be accessed by the users themselves, some participants still feel uncomfortable about collecting and viewing their life experience. This means that lifelog studies can only be carried out using a very small sample. For example, in Smith et al. (2011) there was only one participant involved. Even the most famous lifelogging project “MyLifeBits”, had only one experiment participant (Gemmell et al., 2002).

In this study, seven groups of participants were recruited to carry out seven experiments. The participants employed in the experiments are working in different areas and don’t have prior knowledge of lifelog studies, although they are familiar with using computers. Information about the participants involved in the experiments is shown in Table 3.2. Further details of experiments are listed in subsequent chapters.

In this study, a survey approach is employed to gather users’ experience on the display performance of eight user interfaces (UIs) on three devices: computer, smartphone and E-book reader. The eight UIs are images, images and annotations, images and icons, images and narratives, animations, diaries, icons, and narratives. Four criteria are used to evaluate the display performance as visual appeal, subjective satisfaction, potential for errors and speed of use. Visual appeal is believed to dominate impression judgments (Lindgaard et al., 2011). The other three criteria (effectiveness, efficiency and satisfaction) were defined by the international standard ISO/IEC 9241-11 and have been used by Shneiderman and Plaisant (2005). More details are presented in Chapter 7.

**Table 3.2: The experiment configuration for all tasks in this study**

| Chapter | Experiment | Experiment Description | Participant(s) | Period |
|---|---|---|---|---|
| 4 | Activity Recognition | Recognise user's activities using one accelerometer | 1 | 2 weeks |
| 4 | Multi-Source Location Fusion | Calculating user's location using GPS, WiFi , Bluetooth and Base Station | 5 | 3 months |
| 4 | Identify Important Location | Separating personnel location history into 4 types | 1 | 39 months |
| 5 | Event Segmentation | Segmenting lifelog data stream into events | 5 | 2 weeks |
| 5 | Keyframe Detection | Find the photograph which can represent a whole event | 3 | 2 weeks |
| 6 | Narrative Generation | Generating narratives from users' lifelog data | 5 | 2 weeks |
| 7 | Multimodal Accessing Lifelog Data | Using different devices to access personal lifelog data which are represented using different user interfaces. | 17* | 8-30 minutes |

Note: * indicates that in this experiment, there was one data owner and 16 participants evaluating the displaying performance.

## 3.7 Summary

This chapter described the methodology employed in this study. Although SenseCam is the most common lifelogging tool (Hodges et al., 2011), it is already out of date. In this study, a smartphone is employed by us to develop a new generation lifelogging system. Lifelog data are made up of numbers. To support user's access, two data mining techniques are proposed and introduced: namely term weighting and SVM, to detect meaningful context from massive lifelog data and segment events. More details will be presented in Chapters 5 and 6. This chapter also generally described the experiments and survey employed in this study.

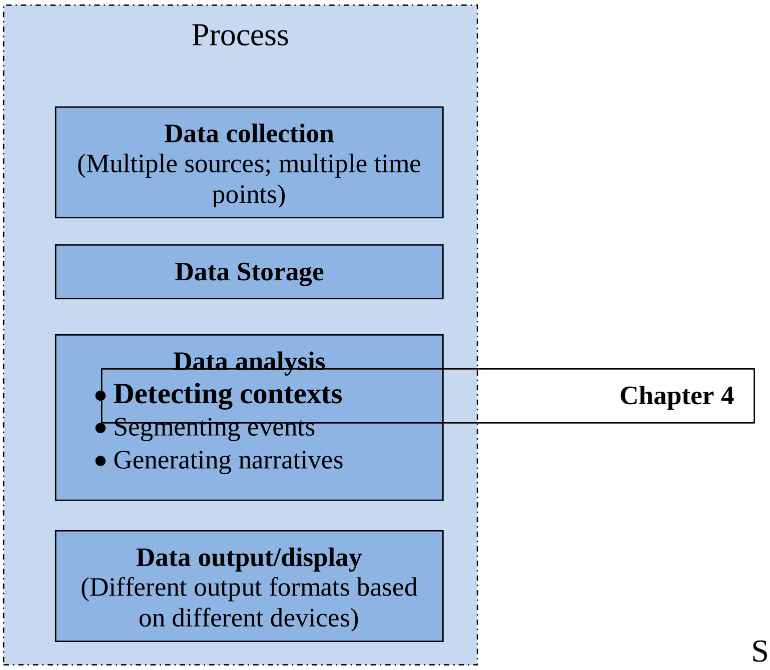
# CHAPTER FOUR

# DETECTING CONTEXTS

## 4.1 Introduction

The previous two chapters presented the background of this study and the methodology employed at the different processes of the lifelogging system. This chapter focuses on the context detection in the data analysis process. It firstly presents details on combining different data streams, i.e. data collected from multiple sensors. Following that, it motivates the need to convert raw sensor data into semantic contexts and describes how this is achieved. Finally the process of implementing virtual sensors for different context detection is provided. Figure 4.1 shows the position of this chapter's work in the whole model.

**Figure 4.1: Work in Chapter 4**

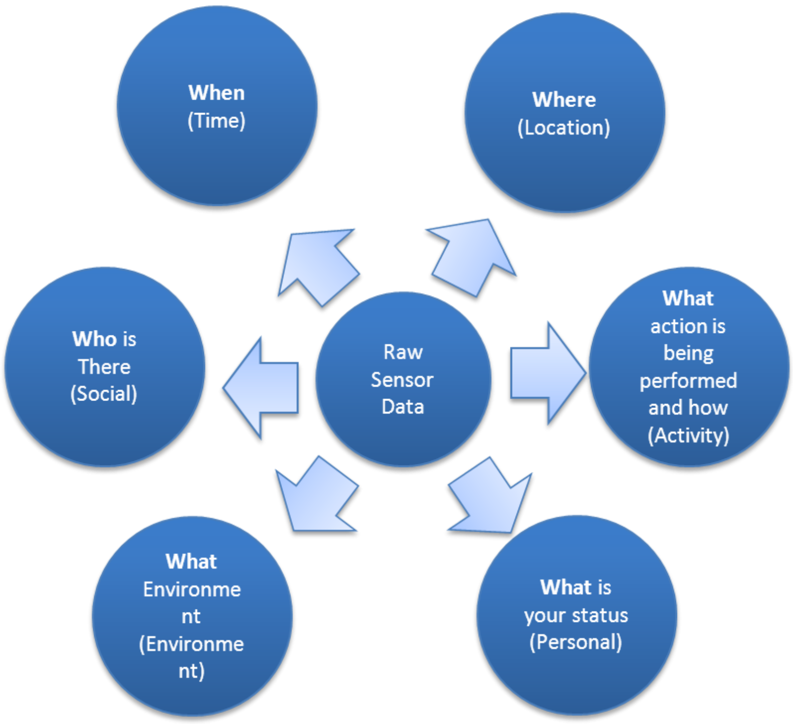



Source: The author (2013)

## 4.2 Using Six Contexts to Represent Life

To present a meaningful reflection of daily life, context needs to be detected and extracted from the heterogeneous sensor data source. Contextual data needs a lot of information to adequately characterise situations. This study adopts the widely used set of contexts (i.e. phrases) to indicate the *"When"*, *"Where"*, *"Who"*, *"What"* and *"How"*. "When" is about time; "Where" is about location; "Who" is about other people; "What" is about the environment; and "How" is about the manner in which the action is being performed. In reality, the user is the key to decide the contexts. To answer *"When"*, *"Where"*, *"Who"*, *"What"* and *"How"*, in this study, all contexts are grouped into six categories. They are the Personal, Time, Location, Activity, Social, and Environment Contexts, as shown in Figure 4.2. We will now discuss how to detect contexts from raw sensor data.

**Figure 4.2: The relationships between context sources**




Source: The author (2013)

### 4.2.1 Personal Context

Personal context indicates the user profiles, e.g. age, gender, and habits. Personal context can be manually created or extracted from other context sources. For example, the proposed lifelogging system in this study can learn the user's life patterns through the user's travel log history in a highly accurate and low battery consumption way.

The lifelogging system in this study will scan the user's location history for a period and then estimate the user's travel habit. After knowing the user's life patterns, the system will change the frequency for data collection. For example, if the system knows that the user usually stays at home during the night time, the GPS data may be turned off at night time. By doing so, the battery of the phone is extended. The pseudocode for this function is presented in Figure 4.3.

### 4.2.2 Time Context

Previous lifelogging tools were implemented with different devices which may use different time clocks (Gemmell et al., 2005). For example, in previous research, a SenseCam was used to capture photographs, and a GPS receiver was used to collect location information. However, the devices used different system clocks and presented different timestamps at the same time.

In this work, all the collected sensor data are timed with one clock. The system also records satellite time when it captures GPS data. It is very easy to identify any data with the wrong timestamp by comparing the two timestamps. The server can also correct data using satellite time automatically.

**Figure 4.3: Pseudocode for detecting personal context using location history**

```
SET base_rate=0.8
WHILE
IF user at home THEN
        SET isHome=true
        saveAtHomeTime(time,isHome)
ELSE
        SET isHome=false
        saveAtHomeTime(time,isHome)
END IF
        SET getAtHomeRate=getAtHomeRateBasedOnHistory(Time)

IF isHome and  getAtHomeRate >base_rate THEN
        change the sensor scan (WiFi, Bluetooth,GPS) to normal frequency
        // because radio consume most of battery
        IF accelerometer values have no change THEN
                Let the system sleep for 5 minutes
                // The system will keep the cpu running
        ELSE
                Wake up the system
        END IF
ELSE
        change the data scan to normal frequency
END IF

IF screenOn THEN // user begin to use phone
        change the sensor scan to normal frequency
END IF
LOOP
```

Source: The author (2013)

Time keeping is a critical element in coordinating activities. In this study, local time is used to coordinate the sensor data. The time is acquired from the timestamp on every physical sensor reading and virtual sensor reading. Time will also synchronise with mobile networks automatically. If a smartphone is unable to acquire the correct time, e.g. when the user is abroad, the system will correct the time using satellite time acquired by the GPS receiver. To optimise system performance it does not collect all of the sensor data at the same time. Some sensor readings (e.g. accelerometer) occur every second while others (e.g. photographs and Bluetooth) occur a number of times per minute. Furthermore, some sensors only return their values when some specific events occur. For example, phone call data is only available when the user receives a phone call. Therefore, time is chosen to be the key binding agent for sensor readings. The semantic interpretation of time adopted is shown in Table 4.1.

**Table 4.1: Detail of time contexts detected from time**

| Name | Description |
|---|---|
| Season* | Spring, Summer, Autumn, Winter |
| Day time* | Morning, Afternoon, Evening, Night |
| Week day* | Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday, Weekend |
| Month* | January, February, March, April, May, June, July, August, September, October, November, December |
| Relative Time | Yesterday, Last week, Last hour, Last minute, Last month |

Note: Time concepts can be separated into absolute time and relative time.
* indicates absolute time.

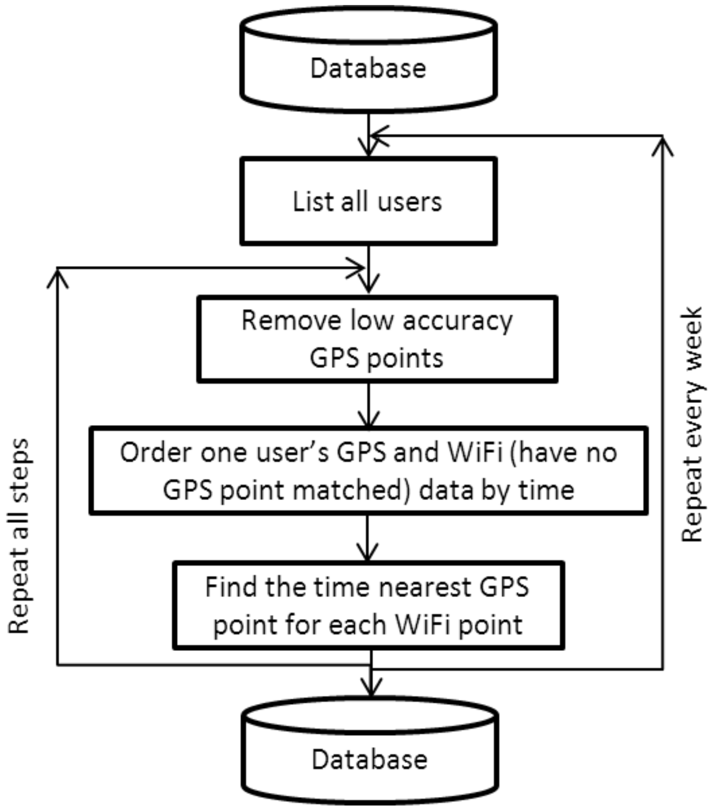
## 4.2.3 Location Context

### 4.2.3.1 Fusing Different Location Sources

To detect location context, WiFi is mainly used alongside GPS, Bluetooth and the Base Station data. Choosing these sensors is based on the consideration of cost, range, granularity and requirements by this study such as detecting location context using GPS sensor. WiFi consumes less power than GPS. We thus would like to use more WiFi than GPS. However, WiFi has less accuracy than GPS. To save the power and improve the accuracy, we designed an approach to combine the WiFi with GPS.

The Media Access Control address (MAC address) is a unique identifier for the WiFi network. In this study, we do not transmit WiFi MAC addresses to physical locations on smartphone, but only when they are uploaded to the server. The GPS receiver stays off when WiFi hotspots are available or the user is not moving. The system will turn the GPS receiver on when the user begins to move or when a new WiFi MAC address is detected. When the server receives the WiFi and GPS data, it will match the WiFi with the GPS automatically. The process of matching the WiFi and the GPS is shown in Figure 4.4. By applying the process on WiFi data, each WiFi will match a GPS point and the accuracy will be improved when more data is received.

**Figure 4.4: The progress to find coordinates for WiFi hotspots**




Note: The code is implemented as SQL storage function. It runs weekly on the server. It will achieve better accuracy for each WiFi when more data collected.

Source: The author (2013)

In a study of a total of 19,147 unique WiFi MAC addresses collected by five users in a three-month period in 2011, 7,136 WiFi hotspots were obtained with the latitude and longitude in less than one second and 3,754 in less than two seconds. A total of 14,628WiFi hotspots show less than one minute time gap with nearest GPS point and are used in this study.

There are two types of WiFi. One is fixed, such as the wireless network at one house. The other one is embedded in portable devices which can move, such as laptops and smartphones called “portable hotspot”. Obviously, the portable WiFi hotspots cannot be used for detecting location context. They need to be separated from fixed WiFi hotspots. In this system, two approaches are used to detect portable WiFi hotspots.

Firstly, we use the distance threshold. For each WiFi MAC address, GPS may be collected more than three times at different time points (usually on different days). Three distances are calculated between two GPS locations out of three. If two distances values are larger than 200 meters, this WiFi is seen as a portable WiFi hotspot. It is put in the portable WiFi list which will not be used for location context detection. That WiFi hotspot will be ignored when it appears next time.

Secondly, we compare the adjacent WiFi hotspots. For each WiFi hotspot, its adjacent WiFi hotspot list is collected twice at two time points (usually on different days). If its adjacent WiFi list is completely different from the other one, this WiFi hotspot is seen as a portable WiFi hotspot. Similarly, its MAC address will be put in the portable WiFi hotspot list and will be ignored when it appears next time.

In past research, some scientists have tried to use Blutooth to detect a user’s location (Mizuno et al., 2007). In this study, 17,090 unique Bluetooth data was collected as shown in Table 4.2. In the Bluetooth data, only 245 Bluetooth in desktops are used to detect location context in this study as they are fixed to a place.

9,784 Bluetooth data were found to be embedded on 4,114 smartphones. 570 laptops were detected. As smartphones and laptops are movable, these Bluetooth data are not used in this study.

**Table 4.2: The statistics of all types of collected Bluetooth data**

| Number | Device Type | Number | Device Type |
|---|---|---|---|
| 59 | 0(MISC) | 87 | 528(MODEM_OR_GATEWAY) |
| 1 | 1344(Unknown) | 1 | 16 (Unknown) |
| 3 | 768(NETWORKING ) | 12 | 1408 (Unknown) |
| 13 | 256(UNCATEGORIZED ) | 1 | 1024(AUDIO OR VIDEO UNCATEGORIZED) |
| 4 | 1600(Unknown) | 245 | 260(DESKTOP) |
| 108 | 1028(WEARABLE_HEADSET ) | 39 | 1664(Unknown) |
| 3 | 264(COMPUTER_SERVER ) | 1,738 | 1032(AUDIO OR VIDEO HANDSFREE) |
| 1 | 1796(WEARABLE_WRIST_WATCH ) | 570 | 268(LAPTOP) |
| 1 | 1044(LOUDSPEAKER) | 2 | 3584(Unknown) |
| 17 | 272(HANDHELD_PC_PDA) | 12 | 1052(PORTABLE AUDIO) |
| 1 | 4352(Unknown) | 24 | 276(PDA) |
| 2 | 1060(SET_TOP_BOX ) | 1 | 5544(Unknown) |
| 9,784 | 516(PHONE_CELLULAR) | 70 | 1084(DISPLAY AND LOUDSPEAKER) |
| 98 | 7936(UNCATEGORIZED) | 75 | 520(PHONE_CORDLESS) |
| 1 | 1280(PERIPHERAL) | 4,114 | 524(PHONE_SMART) |
| 3 | 1288(Unknown) | | |

Note: The "Unknown" types of data can be distinguished according to their ID number in the front. They were collected from the devices which are not defined on developer.android.com

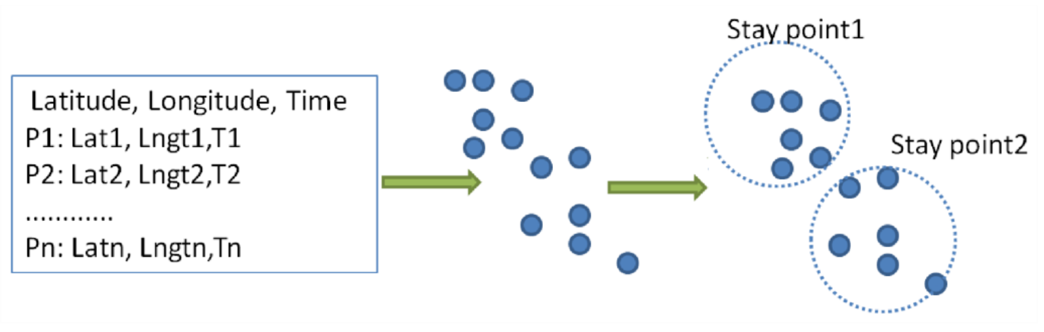
Previous research has shown that locating using mobile phone Base Stations is not very accurate especially when only one Base Station is acquired (Liu et al., 2007). However, mobile phone Base Station signals can cover most of a country. Therefore, in this study, when other locating sources are not available, mobile phones' Base Stations are used. The steps to identify the user's location using Base Stations are described below:

1. List all the location clusters (cluster algorithm will be described in the next section) which have the same mobile phone Base Station ID.
2. Order all selected clusters by the total duration of user linger time.
3. The centre point of the cluster where the user spent the longest time will be chosen as the location.

#### 4.2.3.2 Clustering Location Points

Those location points cannot be directly used because they carry little in the way of semantic knowledge. Actually two location points for the same location rarely have the same spatial coordinates because of the fusion of locations. For instance, at different times' changing GPS points for the place "home" are not identical, although they are very close to each other. Thus modelling is required to group different location points with the same semantic meaning. To address this, density-based clustering is applied as demonstrated in Figure 4.5.

**Figure 4.5: Clustering location points**




Source: The author (2013)

All of an individual's location points are put into one dataset and clustered into several geographical regions. When there is at least one point within an already clustered point, the new points are added to the cluster. In this way, a cluster is formed as a closure of points. Location points from the same place are directly clustered into a density-based closure. In this study, all addresses are acquired using Google geolocation. Clustering will dynamically reduce request frequency, because requests are received only once for each cluster.

#### 4.2.3.3 Detecting Meaningful Location Context

Detecting meaningful location context helps to generate a more understandable narrative. For example, saying "the user is at home (place)" is more understandable than saying "the user is at 580 Collins Ave (address)". Using the term weighting technique described in previous chapter, this section carries out an analysis of a three-year travel log to detect meaningful locations. The results confirm the validity of applying term weighting technique in location context detection, which provides support for using it in the proposed lifelogging system. The experimental dataset and the results are presented below.

**a. Experimental Dataset**

In this experiment, a rich dataset is used. It is a user's three-year travel log data collected from mid November 2005 to January 2009. Location data is collected from GPS every ten seconds. The user was very dedicated to turning on the GPS at most of the time and this recording achieved over 99.5% of all available days. Within this travel log, trips to work, holidays, and shopping were recorded. Walking from one building to another in one place was not recorded primarily due to the start-up time of turning on GPS. For example, the user might have arrived in another building before the GPS was turned on.

Using a gazetteer, the GPS data was transformed into three-tier location names (town, city and country). For example, GPS co-ordinate 30.299982, -97.591782 is converted to the following place name: *Walter E. Long Lake, Texas, United States*. By doing so, the data is more semantically meaningful.

When detecting location context, the last known GPS was employed for the GPS coverage breaks. For example, if the user entered into a building for an hour, and the GPS was not on, the last known GPS point, i.e. the building's location, was used as the location for this period.

The employed travel log contains location data from 43 countries. Every movement was logged by the user, including walking, driving and any airline flights taken during that period. The user's lifestyle is such that a reasonable amount of international travel was undertaken during that period, about twelve international trips per year. Since airline location is included, the number of countries visited seems artificially high, as flying over a country would result in it being given a location log. Table 4.3 shows the countries actually visited by the user (normal text) and the countries that the user simply passed-through while in an airplane (italic text). It is noted that the countries that the user visited for less than one hour and locations that were over sea and not associated with any one country were ignored (286 hours). Table 4.4 shows the number of GPS points logged, countries visited, actual named places visited and the average duration spent at each location, year by year, and in total. The reason the average time spent at each location dropped significantly was because the user bought a car in 2005.

**Table 4.3: Countries visited and hours spent there**

| Country | Hours | Country | Hours | Country | Hours |
|---|---|---|---|---|---|
| Ireland | 19,824 | Norway | 4,073 | China | 1,316 |
| South Korea | 750 | UK | 625 | US | 268 |
| Singapore | 253 | Hong Kong | 133 | Finland | 131 |
| France | 101 | Japan | 82 | Denmark | 56 |
| Germany | 54 | *Sweden* | 41 | *Russia* | 33 |
| *Holland* | 29 | *Poland* | 15 | *India* | 5 |
| *Belgium* | 4 | *Ukraine* | 3 | *Malaysia* | 3 |
| *Mongolia* | 2 | *Estonia* | 2 | *Afghanistan* | 2 |
| *Canada* | 1 | *Belarus* | 1 | *Turkmenistan* | 1 |
| *Latvia* | 1 | *Pakistan* | 1 | | |

Note: The countries with Italic are the countries the user had never been to.

**Table 4.4: Summarising the location log**

| Year | Points | Countries | Places | Avg Time (mins) |
|---|---|---|---|---|
| 2009 | 20,878 | 6 | 1,096 | 33 |
| 2008 | 361,312 | 36 | 14,527 | 36 |
| 2007 | 318,638 | 33 | 11,492 | 45 |
| 2006 | 305,869 | 11 | 7,091 | 74 |
| 2005 | 14,910 | 8 | 388 | 174 |
| **Sum** | 1,021,607 | 43 | 27,508 | 61 |

**b. Results**

Table 4.5 outlines the accuracy of identifying the four location types (e.g. home/work place, social visit, long visit and passing through) using each of the four algorithms previously described. To achieve these figures the precision at cut-off levels (1, 3, 5 & 10) for each month and the overall average precision were calculated. Table 4.5 shows the best performing algorithms to identify (in bold) the four location types.

**Table 4.5: Average precision for 1, 3, 5 and 10 locations using term weighting method**

| | **Home/Work** | | **Social Visit** | | **Long Visit** | | **Passing Through** | |
|---|---|---|---|---|---|---|---|---|
| TF*DF P@ 1/3 | **1.0** | **0.83** | 0.0 | 0.0 | 0.0 | 0.16 | 0.0 | 0.01 |
| P@ 5/10 | **0.59** | **0.51** | 0.06 | 0.07 | 0.23 | 0.24 | 0.22 | 0.32 |
| TF*IDF | 0.26 | 0.28 | **0.45** | **0.49** | 0.16 | 0.15 | **0.58** | **0.58** |
| | 0.28 | 0.29 | **0.38** | **0.35** | 0.15 | 0.17 | **0.58** | **0.59** |
| TF | 0.97 | 0.8 | 0.0 | 0.11 | 0.03 | 0.15 | 0.0 | 0.14 |
| | 0.57 | 0.51 | 0.16 | 0.17 | 0.19 | 0.19 | 0.26 | 0.32 |
| DF | 0.58 | 0.55 | 0.0 | 0.01 | **0.37** | **0.37** | 0.11 | 0.14 |
| | 0.50 | 0.46 | 0.04 | 0.07 | **0.28** | **0.25** | 0.28 | 0.36 |

Note: P@Rel = precision at total number of possible relevant items.

Clearly TF*DF is the best way to locate work/home locations, although it is only a little better than TF. DF shows promise in being able to locate places of long visits, although the precision values are not as high as expected. TF*IDF does find significant social and visiting (holiday) locations, as expected, although once again not as successfully as expected. Finally, a P@Rel (precision at total number of possible relevant items) evaluation gives a score of 1.0 for identifying the three home/work locations when using TF and TF*DF ranking techniques. Simple location frequency analysis (TF) is not effective at finding any location other than Home/Work, and in this case, we have shown TF*DF to be more effective than TF only.

One remaining issue with these results is that the place of work and the place of home have not yet been separately identified. To this end, a rule-based assumption is employed in the process. Home is dominant between 6pm and 6am, while work is dominant between 6am and 6pm. Assuming this, TF*DF ranking is employed to calculate home and work locations and it was found that P@1 for home was 1.0 and P@1 for work was 1.0, which is as expected. Since the user moved home during the logging period, p@2 is actually also 1.0, which illustrates robustness of the process and the proposed techniques.

### 4.2.4 Activity Context

The activity context covers the activities the user was currently involved in and answers the question "What did the user do and how?" It can be described by means of explicit goals, tasks, and actions. The proposed lifelogging system can record multiple sensors relating to activity. For instance, GPS is widely used to detect the speed of movement. However, GPS is not available in all situations such as indoors and is not always accurate. Compared with GPS, an accelerometer consumes less battery but can be used to detect more activities; namely sitting/standing, lying, walking and driving.

Accelerometers are the most promising motion sensors for physical activity assessment in free-living subjects (Meijer et al., 1991). This study employs the 3-axis accelerometer to classify the user's daily physical activities. Figure 4.6 illustrates the volunteer user who is wearing the experimental smartphone and the directions of the accelerometer axes. For this purpose, the frequency and amplitude characteristics of human body acceleration will ultimately determine the technical specifications of the accelerometer.

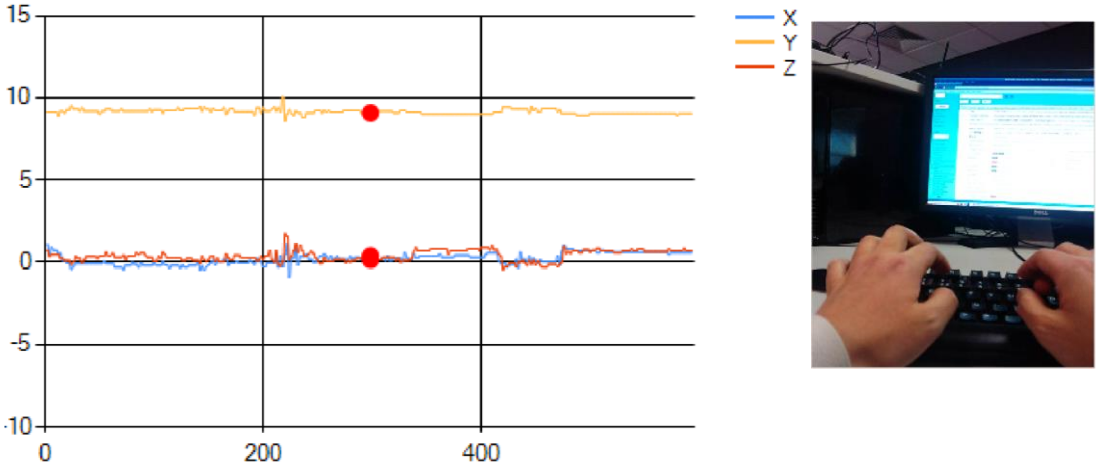
**Figure 4.6: A user is wearing our experimental smartphone**


Note: *X* axis points left, *Y* axis points down and *Z* points to the front.

Source: The author (2003)

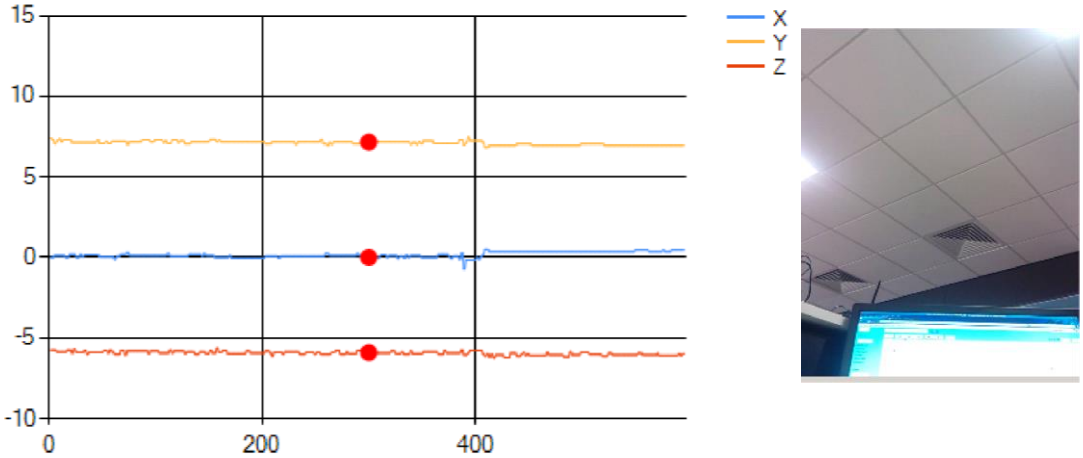
Figures 4.7 to 4.10 illustrate the 3-axis graphs for the user's different activities, i.e. sitting/standing, lying, walking and driving. Due to gravity, one acceleration of 3-axis is always about 1G, so if the value of this axis changes to less and another axis increases, the detection algorithm will note that the mobile phone's angle with the ground has changed. *Sitting/Standing* is quite straightforward to detect; when the user is sitting, all the surrounding accelerations exhibit little change (shown in Figure 4.7). *Lying* is the easiest activity to detect in this study. As shown in Figure 4.8, the acceleration on axis *Z* becomes negative value because of gravity. It is the very obvious feature to recognize lying from other activities.

**Figure 4.7: Graph of 3-axis accelerations for sitting/standing**



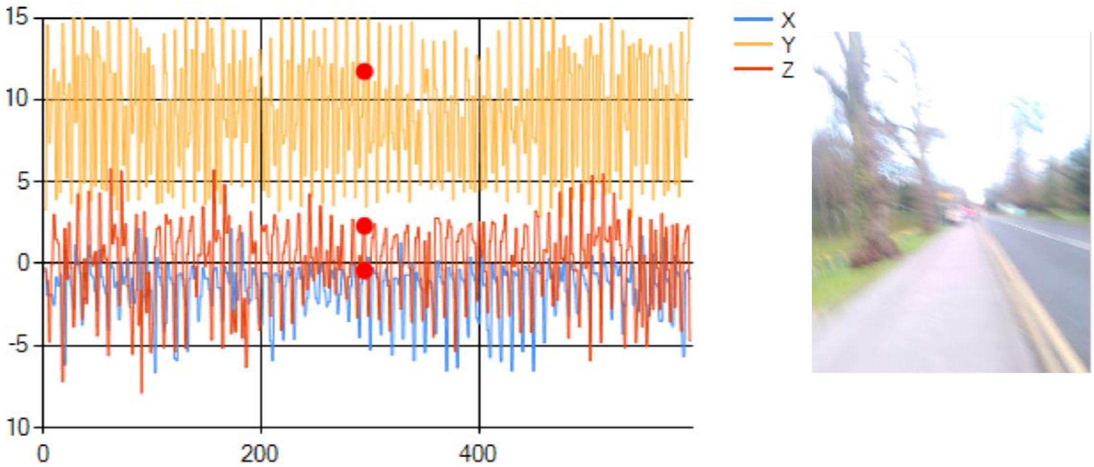

Source: The author (2013)

**Figure 4.8: Graph of 3-axis accelerations of lying**



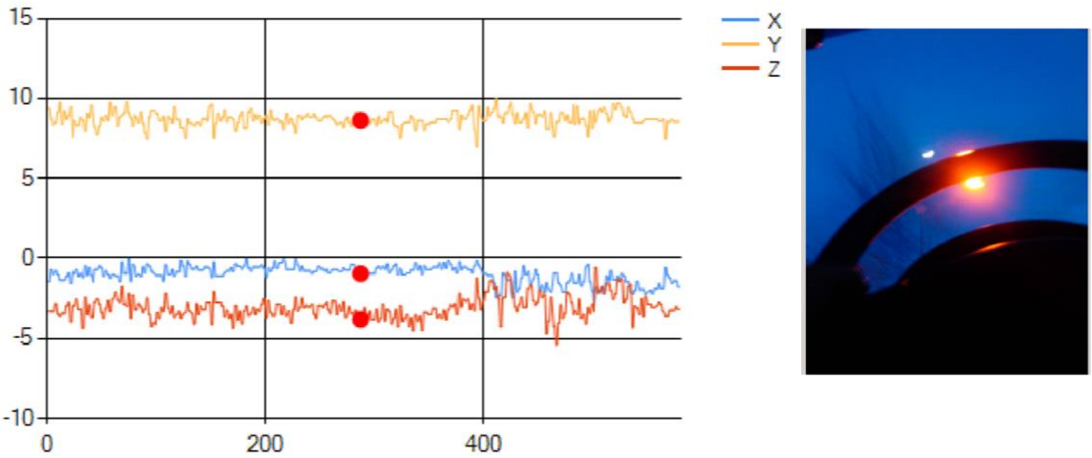

Source: The author (2013)

**Figure 4.9: Graph of 3-axis accelerations of walking**




Source: The author (2013)

**Figure 4.10: Graph of 3-axis accelerations of driving**

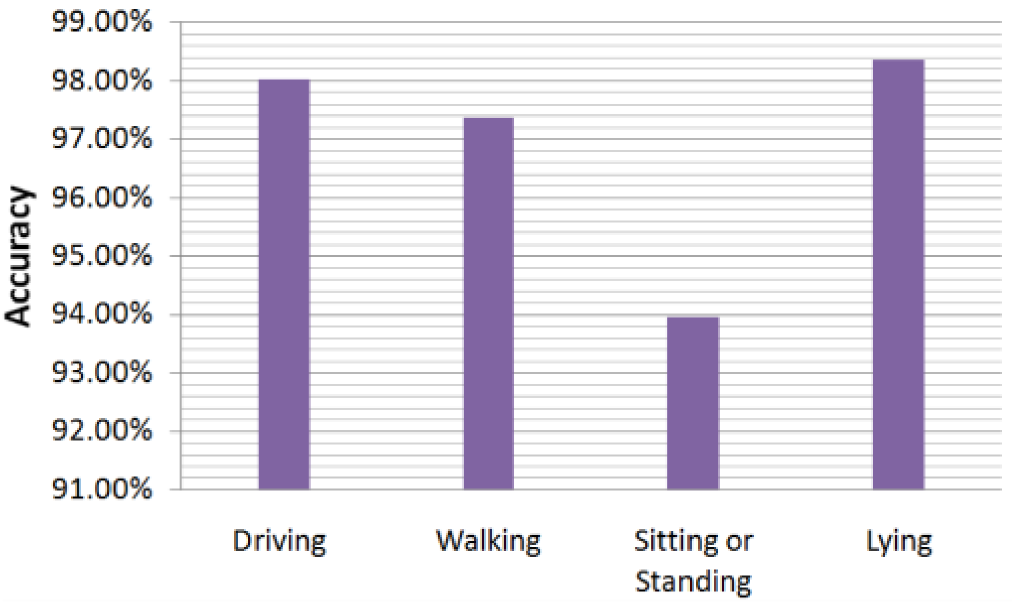



Source: The author (2013)

*Walking* is a very different activity to classify (shown in Figure 4.9), as all three accelerations change a lot. The accelerometer has more sensitivity than humans. When *driving* on the road, even if road is flat where people don't have detected movement, an accelerometer detects minor vibrations (shown in Figure 4.10). The SVM was employed as a machine learning tool to classify activities, given its widespread use in classifying accelerometer-based activity (Ravi et al., 2005). It can be used to classify multi-class data, but in this study a binary-class classification which has higher accuracy is adopted.

The resulting accuracy of detection for each activity is shown in Table 4.6 and summarised in Figure 4.11. It can be seen that all four activity contexts have a high accuracy rate ranging from 94% to 98%. Some wrong estimation (i.e. confusion) is explored. 20 *Driving* instances were classified as *Walking* because of sub-optimal road conditions. 144 *Driving* instances were also classified as *Sitting/Standing* because of red lights or stop signs. 105 *Sitting* instances were classified as *Driving* most likely because of peculiarities in some sitting or standing actions for unknown reasons.

**Figure 4.11: The accuracy of each activity model**




Note: The range is from range 94% to 98%.

Source: Qiu et al. (2011)

**Table 4.6: Confusion matrix of each activity model**

| | Driving | Walking | Sitting or Standing | Lying |
|---|---|---|---|---|
| **Driving** | **1,647** | 20 | 144 | 2 |
| **Walking** | 9 | **4,066** | 185 | 0 |
| **Sitting or Standing** | 105 | 207 | **6,557** | 73 |
| **Lying** | 7 | 5 | 234 | **3,949** |

Note: The data in each row is the user's annotation.

**Table 4.7: Data source for detecting activity context**

| Data Source | Activities |
|---|---|
| Accelerometer and Compass | Sitting/Standing, Driving, Walking, Lying |
| Phone Status | Talking on the phone, Charge Phone |
| Screen Status | Playing with phone |
| Camera | Taking a photograph |
| Music Player | Listening to music |

As behaviours can be changed between periods of photograph captures, 207 *Sitting/Standing* instances are classified as *Walking* and 185 *Walking* instances are classified as *Sitting/Standing*. Given the difficultly in distinguishing *Lying* from *Sitting/Standing*, 234 *Sitting/Standing* instances are classified as *Lying*. For many of these misclassifications, a simple smoothing step post-classification, would address most of these problems and this is planned for future work. Besides accelerometer data, the system collects other activities, such as the phone screen status and phone call status. The activity contexts are described in Table 4.7.

### 4.2.5 Social Context

The social context captures the relationships between user and others such as friends or family, as shown in Table 4.8. A relationship expresses a semantic dependency between two or more people that emerges from certain circumstances they are involved in. Additionally, a relationship is not necessarily static and may evolve and disappear dynamically. In this study, Bluetooth, phone calls & SMS, and face detection are used to detect social context.

**Table 4.8: Data source for detecting social context**

| Data Source | Social Context |
|---|---|
| Phone Call and SMS | Friends/ Family |
| Bluetooth | Colleague /Family/Stranger |
| Photograph | Face-to-face conversation |

Bluetooth is a short-range wireless protocol which enables the exchange of data among two or more devices. It is increasingly included in a wide variety of electronic devices including home computers to portable laptops, smartphones, tablets, keyboards, mice, mp3 players and headphones. By detecting the other Bluetooth signals around the user, the system can estimate the relationships between the users and others. This lifelogging system will scan the frequency the other Bluetooth signals appear during some period. For example, if a smartphone captures a Bluetooth signal rarely during working hours, but often in the evenings and during the weekend, then it is likely that this Bluetooth is from a friend or family, rather than a work colleague. This applies to the phone call and SMS. The pseudocode for this function is presented in Figure 4.12.

**Figure 4.12: Pseudocode for detecting social context using Bluetooth**

```
SET bluetoothList=getUniqueBluetoothPhoneListFromDatabase()
SET weekCount=getWeekNumber()
SET bt_count=length of bluetoothList
FOR i=1 to bt_count
        SET weekHourList=getWeekdayHourFromDatabase(bluetoothList.get(i))
        // select distinct
datepart(hh,time),datepart(dw,time),datepart(week,time),datepart(yy,time)
        //from BluetoothTable where mac_address=bluetoothList.get(i)
        SET number=getCountOfBluetoothAppearFromDatabase(bluetoothList.get(i))
        // select distinct datepart(dy,time),datepart(yy,time)
        //from BluetoothTable where mac_address=bluetoothList.get(i)

        IF bluetoothList.get(i) appear in work time AND number>weekCount THEN
                label the person "Colleague"
        END IF
        IF bluetoothList.get(i) appear in home time AND number>weekCount THEN
                label the person "Colleague"
        END IF
        IF bluetoothList.get(i) appear in home and work time AND number>weekCount
THEN
                label the person "Friend"
        END IF
        IF number==1 THEN
                label the person "Stranger"
        END IF
NEXT
```

Source: The author (2013)

Face-to-face conversation is one of the most essential forms of social activity in our daily lives and a means by which people convey and share information and emotions. This thesis employed www.face.com, which is a technology platform with "best-in-class" facial recognition software to detect faces from our photographs (Kotsiantis et al., 2007). The lifelogging system in this study applies the face detection technique to detect the social context.

It is the combination of these three sets of information (Bluetooth, phone call and SMS, and face detection) fused together that gives a much stronger indication of the social connections of the people we encounter.

### 4.2.6 Environment Context

Environment context in this study captures the noise level and weather information. As shown in Table 4.9, the current context set is detected from the photograph (camera), online weather service and environmental noise level (microphone). The photographs provide a unique insight into the user context by employing visual processing algorithms to visually analyse each image and to identify what is present in each image. Pre-existing visual processing tools are employed to identify the 'What' of an image, where what is a listing of concepts including vehicle, office, door, horizon, etc. In addition to simply providing user context, the photographs act as an integral aspect of the lifelog and are a key attribute of any interface into a digital lifelog.

**Table 4.9: Data source for detecting environment context.**

| Data Source | Environment Context |
|---|---|
| Photograph (camera) | car/bus/vehicles, indoor, outdoor, office, toilet/bathroom, stair, door, buildings, vegetation, road, sky, tree, grass, inside vehicle, view horizon |
| Weather service (online) | warm/cold, windy, sunny/cloudy/raining/snowing, wet / dry |
| Environmental noise level (microphone) | quiet, noisy |

## 4.3 Implementing Virtual Sensors

To fuse all sensor values, it is necessary to implement a function which can return the correct value at any point in time. The data is collected mostly from physical sensors and is not human-readable. In addition, it has different frequencies and generates a time gap. For example, available WiFi is scanned every two minutes. Physical sensors cannot return corresponding value immediately. WiFi can only generate one reading when the hotspot is found. One scanning action may take more than one minute. Accelerometers can generate more than five readings every second. But they do not work in real-time. There is always a time gap between readings. However, in some situations, the sensor value can be predicted. For example, a user's current location can be predicted using his previous locations and location patterns. Because the software gathers different sensor data at different frequencies, the data is returned with different timestamps. Furthermore, raw sensor data is not easily shared. If a virtual sensor supplies the user with raw sensor data, all applications using it will also need to implement the data mining module. That will not only waste time, but it is also difficult to ensure that all users' output is consistent.

In this study, we use five virtual sensors to solve above issues. They are time, location, activity, social and environment virtual sensors. As described in Chapter 2, a virtual sensor is a software sensor which has three functions: 1) combining different format data collected from multiple physical sensors; 2) transforming the combined different format data into a natural language which could be easily understood by humans; and 3) sharing the data through a web interface. To improve the speed of the virtual sensors all sensors data are pre-processed. Virtual sensors can generate the meaningful semantic knowledge to better fit with a user's expectations.

- **Time virtual sensor:** The time sensor will return a different format of time depending on the user's input parameters. For example, it can return "Morning", when the user queries it with specific parameters. In this study, the time sensor is implemented by a segment code. It does not contain any data.
- **Location virtual sensor:** The main function of the location sensor will supply the most accurate location to the user. Furthermore, the process is transparent to the user application. The output of the location sensor not only includes the address, but also other information such as home or work place. As mentioned above, the location information is calculated not only using GPS information, but also using WiFi and Base Station. For the location sensor, if GPS is available, the latitude, longitude and accuracy data will be returned to the user. The address and length of stay will also be returned. When the GPS is not available, the WiFi access point or Bluetooth will be transmitted to the GPS point. If the location information comes from a WiFi access point, the minimal time span will also be returned in case the user uses it to filter out some values. If GPS, WiFi and Bluetooth are not available, the Base Station will be considered as location information.
- **Activity virtual sensor:** As mentioned above, accelerometers can be classified into four different activities as sitting/standing, lying, walking and driving. For the activity sensor, when a new accelerometer sensor data is read to the database, the SVM classifier will be executed and the results will be stored in the database. By requesting the activity sensor, the user's activities such as

walking, sitting can be returned. For other activities such as phone calls, a message can be requested using a specific time range. For this sensor, user ID, time, time period and type are input parameters. The sensor will return different activities according to different type parameters.

- **Social virtual sensor:** Once a new photograph is received, the system will judge if the photograph is blank. If the photograph is not blank, it will be processed by a face detection process. If a face is detected from the photograph, the number of faces and corresponding face's position and size will also be stored. Bluetooth information was considered as an important social context data source. Almost every smartphone has a Bluetooth adapter. Bluetooth can be used to predict the relationships. Some users like to turn it off to save battery life. In our study, we found that 10% of users had Bluetooth switched on. During the pre-processing stage, the Bluetooth and location information were gathered for retrieval. The relationship between the device owner and user can be predicted. For example, if the devices appear in the user's home and also in other place the owner of the Bluetooth can be the user himself or his friend. Social virtual sensor can be used by inputting user ID, time, time range, type. The face, relationship and related information can then be sent back.
- **Environment virtual sensor:** The environment virtual sensor mainly supplies noise information such as "noisy" and "quiet", although it can also acquire location information from the location sensor. With time and location information, it can locate the weather information and return it to the user. To enhance the response speed, all the weather information was downloaded and stored in the database when any new location was detected.

## 4.4 Summary

Physical sensor data is easily collected but very difficult to semantically understand as most of it is made up of numbers. It needs to be converted into semantic contexts such as personal, time, location, activity, social and environment contexts. This chapter introduced how these contexts were detected. In particular, we applied term weighting into the location context detection and the SVM into the activity context detection. Lastly, this chapter presented how to implement the virtual sensor in order to combine different format data; transform the combined data into a natural language; and to share the data through the web.

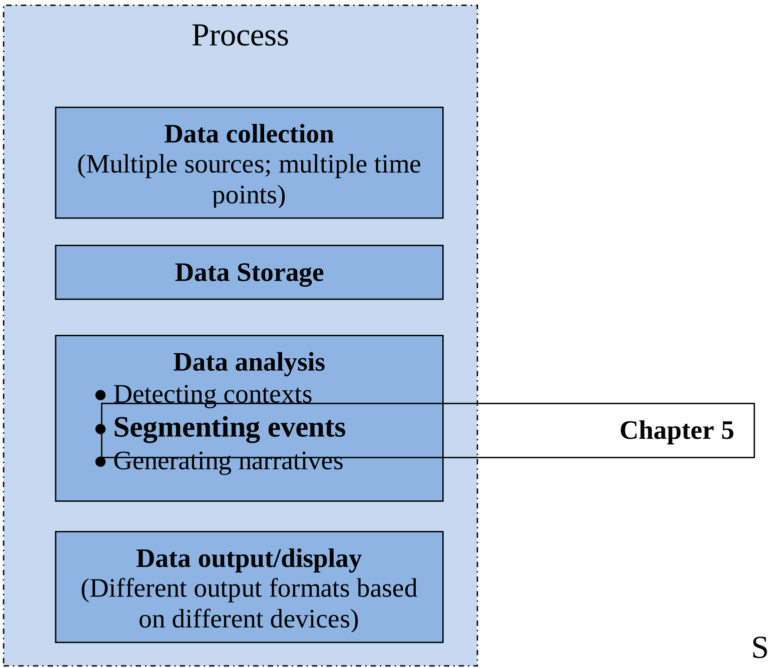
# CHAPTER FIVE

# SEGMENTING EVENTS

## 5.1 Introduction

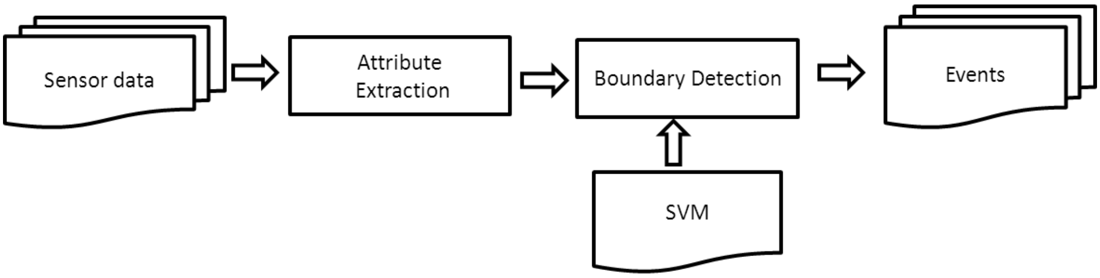
This chapter provides a detailed explanation of the approach used to segment sensor data into events. It shows what attributes are extracted from sensor data based on which events are segmented. The experimental process and results are presented for testing hypothesis 1 which proposes that event segmentation can be performed effectively by detecting changes in sensor data. Finally, to select a suitable keyframe for each event, three users' keyframe selection methods for their daily lives are compared. The context sources leading to the best keyframe selection are summarised. Figure 5.1 shows the position of this chapter's work in the whole model. Figure 5.2 presents the process to segment events.

**Figure 5.1: Work in Chapter 5**




Source: The author (2013)

**Figure 5.2: The process to segment events**




Source: The author (2013)

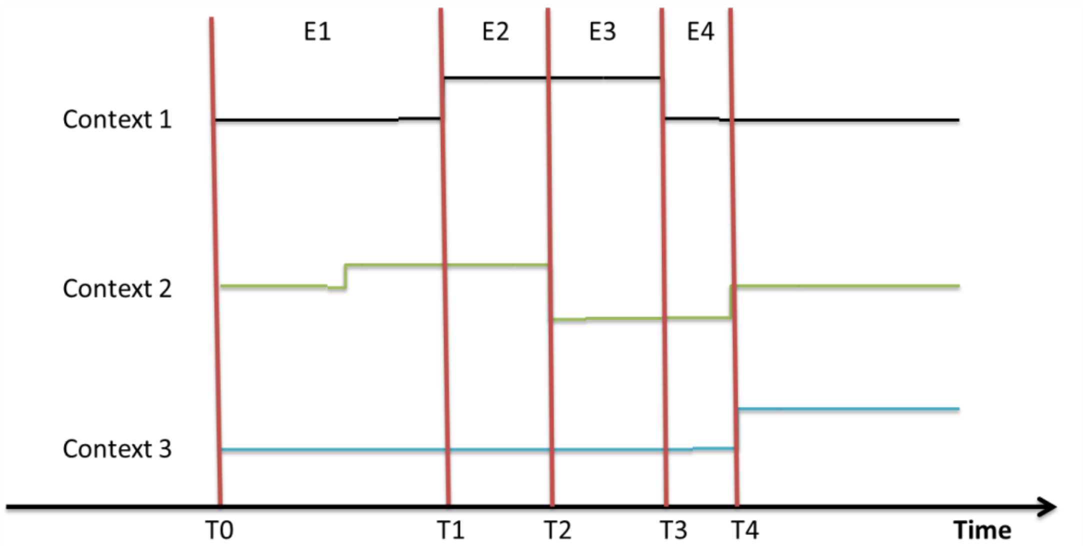
As shown in Figure 5.2, the event segmentation includes two key parts namely attribute extraction and boundary detection. Extracting attributes needs detecting the change of context, and standardising unit due to the different frequency in data sources. We begin with the change of context as follows.

## 5.2 Change of Context

As described in Chapter 2, event segmentation has been used to organise information by some researchers in lifelogging community (Doherty et al., 2012). Three main reasons for using events are: 1) an event is a natural unit for human memory; 2) an event is a reasonable unit for lifelog data management; and 3) there is no big context change in one event. Along with the existing researchers in lifelogging domain (e.g. Chen et al., 2010; 2013), the lifelogging system in this study adopts the method of context change to segment events.

As mentioned above, context pattern changes may generate a new event. However, not every context change will produce a new event. As shown in Figure 5.3, Event *E1* is from time *T0* to *T1*. At time T1, the change of *Context 1* produces a new event *E2* and the change of *Context 2* between *T1* and *T2* makes a new event *E2*. However, between time *T0* and time *T1*, *Context 2* has a small change but it does not create a new event. Between time *T3* and time *T4*, *Context 2* and *Context 3* change at the same time, making a new event. Take one real example, picking up a mobile phone to check the time will usually not generate a new event, but picking up a mobile phone and making a phone call may constitute a new event. Using machine learning techniques, this study will investigate what context changes will make a new event, and how much change is necessary to make a new event.

**Figure 5.3: Detecting event boundaries by detecting the context changes**




Note: By detecting changes of context it is possible to detect an event's boundaries. However, not every change of context will result in an event boundary. An event boundary may be made by changes of several contexts.

Source: The author (2013)

## 5.3 Standardising Unit into One Frequency

The data is collected at different frequencies due to the difference in sensor capability and practicality. For example, weather data is sampled at 1/1800 Hz (once every thirty minutes), while accelerometer data from an experimental smartphone is captured at 5 Hz. The less frequent weather information collection is because weather is more stable compared with accelerometer data. To segment lifelog data into events, those data or their values should be standardised to the same sampling frequency.

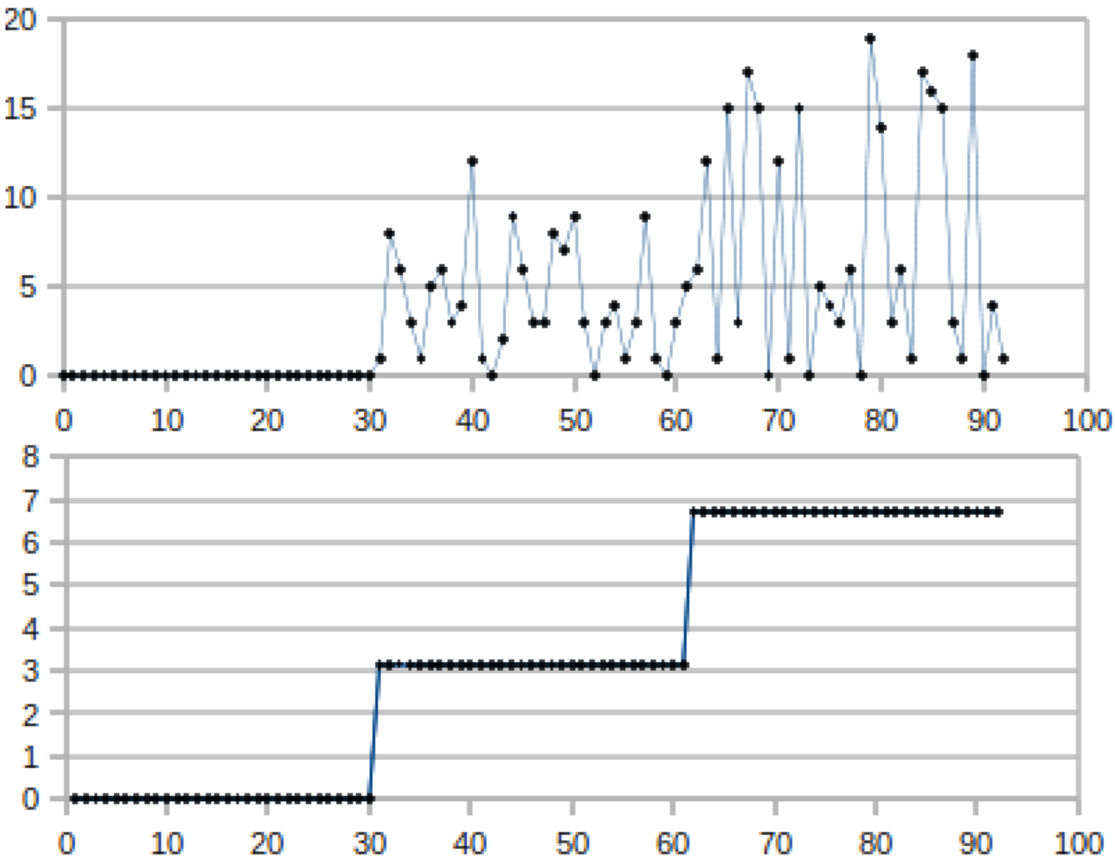
In this study, to generate ground-truth, volunteers were recruited to collect and annotate lifelog data. Practically, it is not possible to ask them to annotate every piece of data, but they were required to annotate a data section. This study chooses the 30 seconds period, the same frequency as a wearable camera.

This process orders all the photographs by capture time, then uses the time of each captured photograph as a reference point. All data collected 15 seconds before and after a photograph is captured and is considered as belonging to one unit. For high frequency sensors such as an accelerometer which generates five readings of data every second, some attributes were extracted from the 30 seconds period data (the detail will be discussed in following sections). For low frequency sensors such as the WiFi adapter which only scans WiFi access points once every two minutes, the WiFi hotspots collected one minute before and after the photograph are considered belonging to one unit.

## 5.4 Extracting Attributes for Event Segmentation

After standardising the frequency of data into units as described above, a subset of attributes (e.g. standard deviation) are chosen to calculate these data units while preserving or improving the discriminative ability of the classifier. The attribute selection affects generalisation performance, running time requirements, constraints and interpretational issues (Weston et al., 2001). As shown in Figure 5.4, 90-second accelerometer data is divided into three groups: 0-29, 30-59 and 60-89. It is not easy to identify value changes from the higher figure. However, if the standard deviation attribute for each group is extracted, the changes can be found in the lower figure. If the values are from an accelerometer, we can say that the user's activity strength is becoming stronger with time. The number of values that need to be processed decreases dramatically if those values can be substituted by their standard deviation. The lower graph has a better performance on computing than the upper one. It also shortens the running time. Therefore the lower graph can show the gradual trend of the graph.

**Figure 5.4: An example for extracting standard deviation attribute**




Note: Raw accelerometer data values are not used in event segmentation, but extracted attributes from group of data. That makes it easier to find the changing trend of values. The upper figure shows the raw data points; the lower figure shows the standard deviation ( $\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - u)^2}$ ) of per 30 data points.

Source: The author (2013)

### 5.4.1 Extracting Image Attributes

To facilitate the calculation of image similarity, this study uses three different low-level visual attributes to measure the similarity of images. They are scalable colour, colour layout and edge histogram. The attributes are based on the moving picture experts group (MPEG)-7 features (Manjunath et al., 2001). MPEG-7 features are a standard description of multimedia content (O'Connor et al., 2005).

- **Scalable colour:** A colour histogram in the hue, saturation, value (HSV) colour space is derived using a Haar transform coefficient encoding which can be calculated efficiently, allowing for scalable representation (Keogh et al., 2001). In this study, 64 numbers are extracted to represent each image.
- **Colour layout:** The compact descriptor captures the spatial layout of the representative colours on a grid superimposed on an image. This study first divides the photographs into 3x4 blocks to guarantee the resolution or scale invariance. Then the colour layout descriptor is obtained by the discrete cosine transformation of each block's dominant colour and non-linear quantisation of their discrete cosine transform (DCT)-coefficients (Kasutani and Yamada, 2001).
- **Edge histogram:** The edge histogram represents the spatial distribution of edges in an image (Won et al., 2002). It could be used to identify changes of scenes. The edge number in the photographs may increase when the user is walking from indoors to a park. In this dissertation, this attribute is described by 80 numbers.

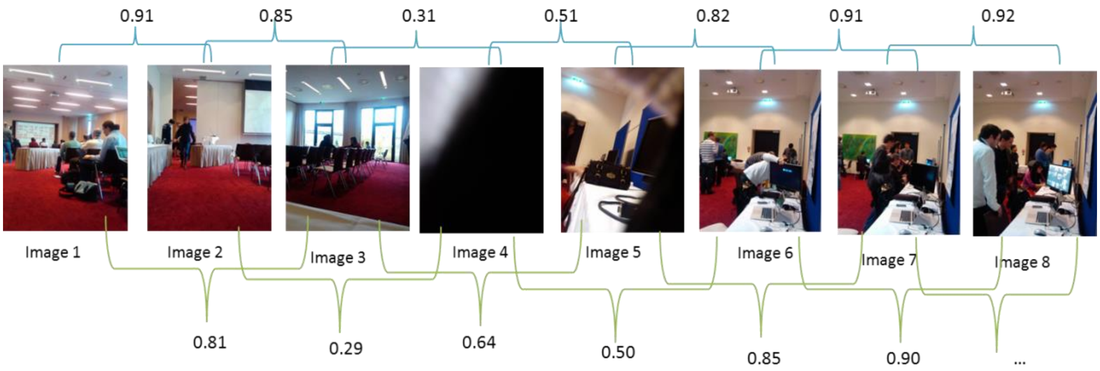
The MPEG-7 visual features are represented by vector values, and are used to calculate the similarity of images. Euclidean distance is chosen as this has been shown by previous researchers to provide intuitively reasonable results (Agrawal et al., 1993; Wang et al., 2005*a*). Euclidean distance is the *ordinary* distance between two points that one would measure with a ruler and is given by the Pythagorean formula. It is most commonly used due to its simplicity (Wang et al., 2005*b*). By using this formula as a distance measure, Euclidean space becomes a metric space. The equation for the Euclidean distance is shown in Equation 5.1.

$$d(x,y)=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2} \qquad \textbf{(Equation 5.1)}$$

Where *x, y* are two vectors and *n* is the number of features for each object.

If adjacent images are sufficiently dissimilar, event boundary may exist between the two images. However, this is not always the case. As the phone is hung up, the camera may occasionally be covered by the user's hand or clothes. To avoid misjudgement in this situation, this study groups five photographs as one unit in order to calculate the similarity of photographs.

**Figure 5.5: Image similarity calculation**




Note: Every image is calculated not only using its adjacent image, but also using the two preceding and succeeding images.

Source: The author (2013)

As shown in Figure 5.5, the similarities for *Image 4* are calculated between it and *Image 2*, *Image 3*, *Image 5* and *Image 6*. In this process, the similarity of *Image 2* and *Image 3, Image 5* and *Image 6*, *Image 3* and *Image 5, Image 3* and *Image 6*, *Image 2* and *Image 5*, *Image 2* and *Image 6* are also calculated. If there are no photographs before or after *Image 4*, the similarity is zero. All the similarity calculations will be applied on the three low-level features using the Euclidean distance equation. Those similarities will be used as attributes for *Image 4* as input to the event boundary calculation process.

### 5.4.2 Extracting Location Attributes

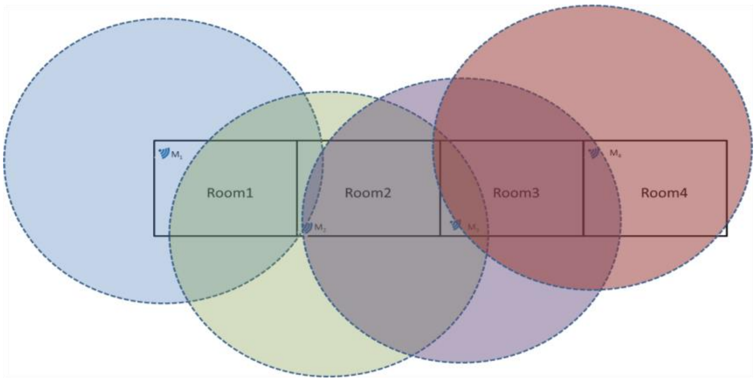
In this study, location is used as an important context source to identify the boundary between events, as it has been used as a key organisation methodology for lifelog data and a key clue to help user access their lifelog data (e.g. Aizawa et al., 2004*b*; Kurby and Zacks, 2008; Liao, 2006; O'Hare et al., 2005*a*).

The lifelogging system in this study collects location data from different sources, namely WiFi, Bluetooth, GPS and Base Station. The multiple sources provide more opportunities to improve the accuracy of the event segmentation technique. As mentioned above, to detect the event boundary is to detect the change of context. For GPS data, the changes can be measured by speed. The changes in WiFi and Bluetooth can be measured by the change of signal strength and MAC addresses.

For a Base Station, the change can be captured by the change in Base Station ID. However, not every change of location will create a new event. For example, people usually identify a period of driving or walking as an event, although location changes all the time. To identify the boundary of events, the rate of changes should also be investigated. The attributes extracted from location context include speed, current number, value and number of same WiFi MAC addresses.

- **Speed:** Speed is calculated from GPS data. In this study, machine learning can use this attribute to avoid misclassifying a driving or walking event.
- **Current number, value and number of same WiFi MAC addresses:** The changes of location may create a new event. The location changes will also cause the collected WiFi list to change. As shown in Figure 5.6, when the user is in room 1, the system can capture WiFi $M_1$ and $M_2$. In room 2, the system can capture WiFi $M_2$, and $M_3$. This study compares the changes of WiFi lists to identify location changes. If WiFi lists have no changes in a period, it may mean that the user hasn't been in other rooms. If there are some changes in the list. It means that the user moved to an adjacent room or moved to another side of room. For example, the user on the left side of room 2 can capture different WiFi hotspots compared to the right side of the room. WiFi changes can not only identify location changes in buildings but also location changes outside.

**Figure 5.6: Example of four rooms and four WiFi routers**




Note: There are four rooms and four WiFi routers. The routers' signal range is 30 metres, thus a mobile phone in each room will receive a different WiFi list.

Source: The author (2013)

### 5.4.3 Extracting Activity Attributes

In real life, an event boundary may occur when the user's activity changes. For example, it may be a new event when a user starts walking from sitting. Compared with other contexts, the activity is more dynamic, because humans can change activities in a very short time. In this study, other activity data sources such as answering a phone call are also considered. The following attributes are employed in this study.

- **Change of accelerometer's maximum and minimum values, and standard deviation:** These values are the key attributes to identifying the change of activity context. When a user begins to walk, the maximum value will increase in a short time. At the same time, the "Standard Deviation (SD)" will also increase because of dynamic changes to acceleration on the three axes. Some short term activity changes can only be identified by a combination of these attributes. For example, the user stands up and passes his friend a cup of water, then returns to his seat. That activity may only take 5 seconds, but that action will make the current maximum and minimum acceleration values quite different from previous values. In comparison to those values, the SD will change little. Their combination may remove the effects of some short period activities on event boundary detection. For example, picking up a cup may not be considered an event.
- **Change of screen status:** Mobile phones are playing an increasing important role in our lives. The change of mobile phone status will record a user's activity by phone. The change or no change to the screen status in a time unit is used to

classify the boundary of events. For example, a user may begin to play a game on the phone after coming back from a meeting.

- **Change of phone call status:** Phone calls are one of most important social activities. They are an important key for event boundary detection.
- **Change of music player and headphone status:** Many people listen to music when they walk, run or take the bus. The kinds of songs playing are important to identify a user's interests, but are not required for event segmentation. This attribute is not related to what song the user is listening to but to the activities such as plugging in or plugging out the headphones and also the music player's status. For example, the user may start the music player when he gets on a bus. It may be a starting point of one new event.

### 5.4.4 Extracting Environment and Social Attributes

Attributes based on environmental noise level and face detection are extracted. Most of the environment context sources, such as the weather, are relatively stable during the event. For example, the temperature might only change slightly in a whole day. Even if it does keep changing, people usually do not notice it very much. Another important source for environment context is the noise level. Siewiorek et al (2003) have classified environmental noise level into three states: low, medium and high. The low value describes the quiet environment, the medium value identifies common situations such as talking and the high value states the environment such as a pub or bar. Along with Ma et al. (2003a, 2003b), this study does not record voice but records the environmental noise level of the environment. According to the environmental noise level, the system sets up the phone ring volume. In this study, the following attributes are included:

- **Change of environmental noise's maximum and minimum values, and standard deviation:** Similar to activity context detection, those are key attributes in classifying different environments. In this study, the frequency of the environmental noise levels is very low. The changes of environmental noise levels are considered in a two-minute interval around the photograph.
- **Change of face numbers:** The beginning and ending of a conversation may be a beginning of a new event and ending of an old event. To detect the conversation, face detection is a key step. Change of face numbers is an attribute in this study.

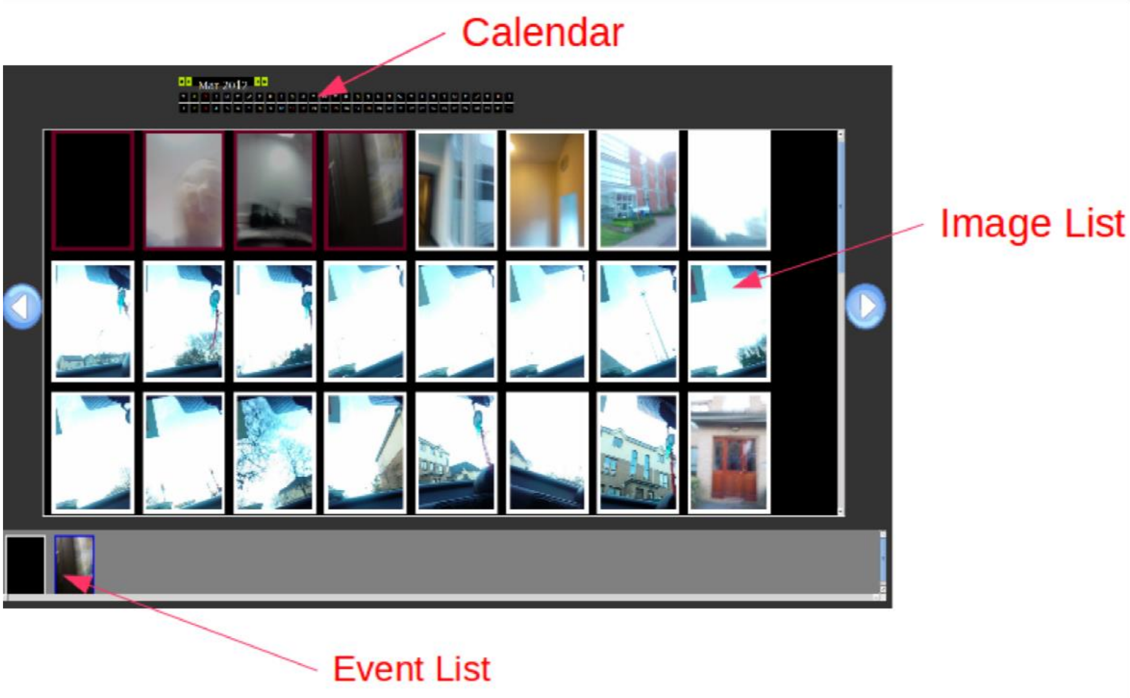
## 5.5 Dataset for Event Segmentation

Hypothesis 1 proposes that the event segmentation can be performed effectively by detecting changes in sensor data. To test it, 5 users were asked to collect their everyday lifelog using the proposed lifelogging system for two weeks (discussion on sample size will be presented in Chapter 8). The configuration of the experiments is show in Table 3.2 in Chapter 3. When all data is uploaded to the server, users can browse their photographs using a web interface to annotate their data. With the interface, they can simply select the beginning and end photograph of every event. To avoid forgetting, participants were asked to annotate the data on the evening of data collection or the next morning.

The annotated data is used as the training dataset for the SVM to segment events with a five-fold cross-validation. The results from SVM will be compared to the manually annotated data. The SVM's performance in event segmentation is evaluated based on the accuracy rate on event boundary detection.

As shown in Figure 5.7, there are three panels: Calendar panel, Image List panel and Event list panel. Once the user logs into the system with his user name and password, the latest data will be displayed on the web page. Users can choose to browse any day's image data by clicking the calendar. On the web page, the related day is highlighted if the users have uploaded that day's data. To segment the event, the user can choose any photograph as the beginning of one event, and then choose one as the end of event. After that, the event will appear in the Event list. If the user makes a mistake, the event can be deleted by double-clicking it.

**Figure 5.7: User annotation interface**




Note: Participants can choose the beginning and end of an event by clicking on the images.

Source: The author (2013)

**Table 5.1 Statistics on event segmentation data collected**

| User | Images | Events | WiFi | Bluetooth | Noise* Level | GPS | Accelerometer | Screen status | Base Station | Call | SMS | Charging | Music |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 22,246 | 276 | 78,699 | 7,428 | 11,088 | 1,825 | 667,481 | 482 | 265 | 42 | 8 | 18 | 6 |
| 2 | 24,115 | 230 | 25,0342 | 10,788 | 11,959 | 5,323 | 723,687 | 366 | 378 | 18 | 5 | 16 | 0 |
| 3 | 24,251 | 255 | 239,993 | 11,994 | 12,175 | 7,745 | 727,598 | 594 | 362 | 33 | 6 | 15 | 0 |
| 4 | 30,256 | 305 | 293,622 | 15,526 | 15,107 | 10,523 | 907,856 | 368 | 498 | 35 | 5 | 20 | 0 |
| 5 | 19,278 | 169 | 63,593 | 7,352 | 9,636 | 1,305 | 578,396 | 282 | 198 | 25 | 2 | 13 | 0 |
| total | 120,146 | 1,235 | 926,249 | 53,038 | 59,965 | 26,721 | 3,605,018 | 2,092 | 1,701 | 153 | 26 | 82 | 6 |

Note:* indicates environmental noise.

In this study, five participants annotated 1,235 events in their own two weeks' lifelog data. As shown in Table 5.1, a total of 120,146 photographs were captured by five users. On average, each user wore the smartphone for 14.5 hours every day and a total of 926,249 WiFi hotspots were collected in addition to information on 53,038 Bluetooth devices. However, most of the WiFi hotspots and Bluetooth device information were repeated because the system continued to scan and store wireless information even when the user was not moving. To detect activities, the system collected accelerometer data at 5Hz. In a two week period, a total of 3,605,018 3-axis accelerations data readings were collected. As mentioned above, GPS information collected in the study is not classified by time in order to save battery life, but by distance. The system turns off the GPS automatically when it collects WiFi information matching to a specific latitude and longitude as a location source. Therefore, only 26,721 GPS point were collected. As there was no WiFi available in some areas, Base Station information was collected to aid locating users 1,701 Base Station IDs were collected. In addition to these, 59,965 environmental noise level readings, 153 phone call events, 26 SMS message events, 82 phones charging events and 6 music events were also identified.

## 5.6 Event Segmentation using SVM

This study employed the support vector machine learning (SVM) to segment events. The SVM has good generalisation ability as it is based on the principle of the structural risk minimisation in statistical learning theory. The details on why use SVM has been presented in Section 3.4.

Event boundary recognition is a classification problem with limited samples where the SVM classifier finds its usage. These experiments used LibSVM which is an implementation of SVM. LibSVM optimises the best parameters for the RBF kernel on different classifications using five-fold cross-validation (Hsu et al., 2003). We follow the four-steps for using SVM as:

1) choosing the training dataset (the dataset with manual annotation by five participants);
2) extracting the optional attributes of data (three methods as image only, location only and all attributes);
3) training the classification model;
4) evaluating the classification.

The classification results are shown by confusion matrix in Tables 5.2, 5.3 and 5.4. The effects of each of the approaches discussed earlier were investigated experimentally and the results are reported in Figure 5.8. To evaluate the effectiveness of each approach, three different metrics were used: precision, recall, and F1-Measure. Results in this section are reported in terms of F1-Measure as it is hoped to maximise both precision and recall.

- **Precision:** This is a percentage measure of the boundaries proposed by the system that are accurate, compared with the results annotated by participants.
- **Recall:** This is a percentage measure of how many of the ground truth boundaries were identified, compared with the results annotated by participants.
- **F1-Measure:** It's a single measure that incorporates both precision and recall.

$$F_1 = \frac{2*precision*recall}{precision+recall} \qquad \textbf{(Equation 5.2)}$$

## 5.7 Experimental Results on Event Segmentation

In this study, we do not use any raw sensor data, but the attributes described in previous sections. An experiment is set up to investigate three approaches for detecting event boundaries with SVM machine learning using different datasets.

- **Using image attributes:** As shown in Table 5.2, 1,563 (63.3%) event boundaries were detected correctly from 2,470 instances, 907 (36.6%) were false negatives and 982 (0.8%) false positives which are not event boundaries were incorrectly detected as boundaries.
- **Using location attributes:** Our activities related to location and previous research has shown that location is one of the most important clues for people to recall their lives (Naaman et al., 2004). During this experiment it was found that users tend to separate their lives using locations. The results shown in Table 5.3 indicate that 1,254 (50.8%) out of 2,470 event boundaries were correctly detected, 1,216 (49.2%) were false negatives and 1,689 (1.4%) false positive instances which were not event boundaries were incorrectly detected as boundaries.
- **Using multiple attributes:** This not only includes image attributes and location attributes, but also activity, social and environment attributes. Those attributes will provide more choices to generate a classification model. In the results shown in Table 5.4, 1,912 (77.4%) of 2,470 event boundaries were detected correctly. 558 (22.6%) were false negatives and 769 (0.7%) false positive instances which are not event boundaries were wrongly detected as boundaries. This approach resulted in the best accuracy.

**Table 5.2: Confusion matrix of the SVM classifier for colour attributes**

| | Participants Annotation | Event Boundary | Not Event Boundary |
|---|---|---|---|
| Event Boundary | 2,470* | **1,563** (63.3%) | 907 (36.6%) |
| Not Event Boundary | 117,676* | 982 (0.8%) | **116,694** (99.2%) |

Note: * indicates the data is the user's annotation.

**Table 5.3: Confusion matrix of the SVM classifier for location attributes**

| | Participants Annotation | Event Boundary | Not Event Boundary |
|---|---|---|---|
| Event Boundary | 2,470* | **1,254** (50.8%) | 1,216 (49.2%) |
| Not Event Boundary | 117,676* | 1,689 (1.4%) | **115,987** (98.6%) |

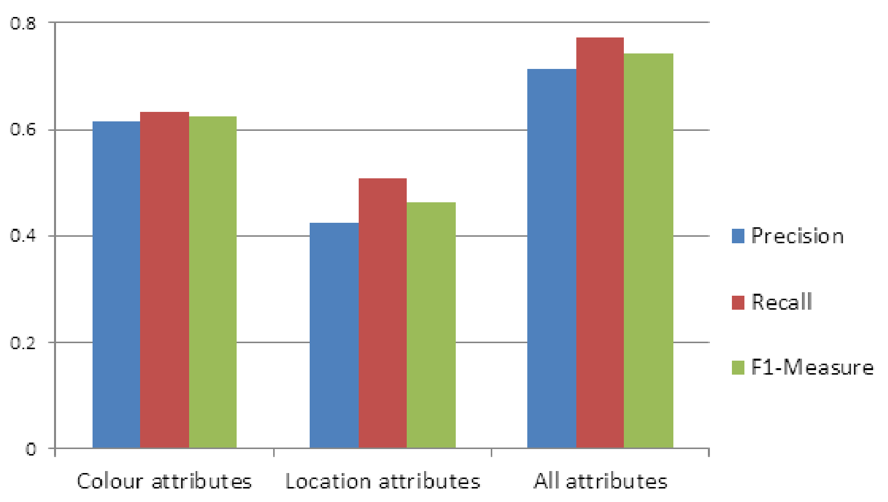
Note: * indicates the data is the user's annotation.

**Table 5.4: Confusion matrix of the SVM classifier for all attributes**

| | Participants Annotation | Event Boundary | Not Event Boundary |
|---|---|---|---|
| Event Boundary | 2,470* | **1,912** (77.4%) | 558 (22.6%) |
| Not Event Boundary | 117,676* | 796 (0.7%) | **116,907** (99.3%) |

Note: * indicates the data is the user's annotation.

**Figure 5.8: Overall classifiers performance**




Note: The optimal performance was obtained using all attributes of the classification model in terms of precision, recall and F1-measure.

Source: The author (2013)

As shown in Figure 5.8, three criteria, i.e. precision, recall and F1-measure, score the highest when all attributes are used. Along with the higher accuracy rate of all attributes method than image only or location only attributes, it can be seen that using multiple attributes to detect event boundaries gives the best results.

## 5.8 Keyframe Selection

So far using the SVM to detect event boundaries has been described. Now the focus shifts towards discussing methods to select a representative image for each event, i.e. keyframe selection.

Keyframe selection is a very common task in the video domain and has recently been introduced in lifelogging research (Doherty et al., 2008). Lifelogs are very different to video data. Video is the same for every audience where most of the faces belong to strangers. In lifelogs, the faces appearing most often are known (Doherty and Smeaton, 2008b). The location does not usually change during an event. Therefore, this research only investigates how much activity, social, and environment contexts affect keyframe selection.

### 5.8.1 Keyframe Selection Experiments

Experiments were set up to judge three approaches to select a keyframe. The three approaches are social, activity and environment context-based.

- **Keyframe selection based on social context:** Socialising interactions are very important activities for people, such as talking to friends. This approach will detect the face from every photograph in order to investigate if faces affect the importance of the photograph.

- **Keyframe selection based on activity context:** Most of people's activities involve sitting and much of that could be facing a computer or TV. In such circumstances photographs can be considered valid keyframe candidate photographs.
- **Keyframe selection based on environment context:** The environment can impact on the importance of photographs, e.g. when users attend an important or very busy event. This experiment will investigate if the level of environmental noise as an indicator of environment context affects the importance of the photograph.

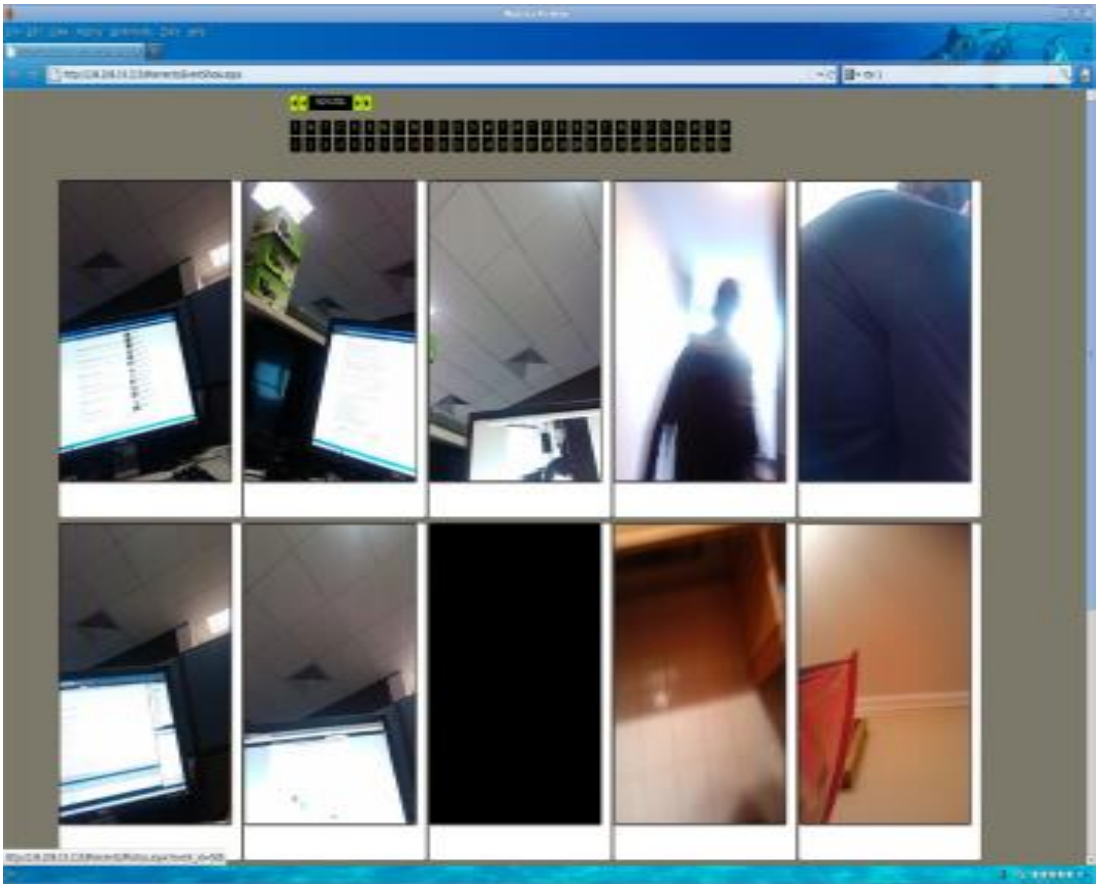
Three users participated in this experiment. They were asked to wear the experimental smartphone for 14 days and to annotate all important photographs that represented important moments during the day. This prevented the users from forgetting what happened. It also helps minimise a user's misjudgement. For example, in one event, where all photographs are similar, if the user is asked to choose one as the keyframe, he may choose one randomly. Such annotation would be useless for the experiment. In this experiment, the users were not requested to select a keyframe from each event but to select all important photographs.

An additional requirement is that the users need to browse every photograph collected that day. The browser allows users to access any photograph by clicking on a relevant hour as shown in Figure 5.9. All photographs captured in that period will be displayed in the Web Browser. As illustrated in Figure 5.10, participants can rate photographs with different scores. The number of stars indicates the importance of the photograph. If a user thinks the photograph is mildly important, he can rate it with one star. A rating of 4 stars means it is the most important photograph for the user. In total, 56,201 photographs were collected in this experiment and 637 (1.1%) of these were annotated as important photographs by users. The statistics of important photographs is shown in Table 5.5.

**Table 5.5: Statistics on keyframe selection survey data**

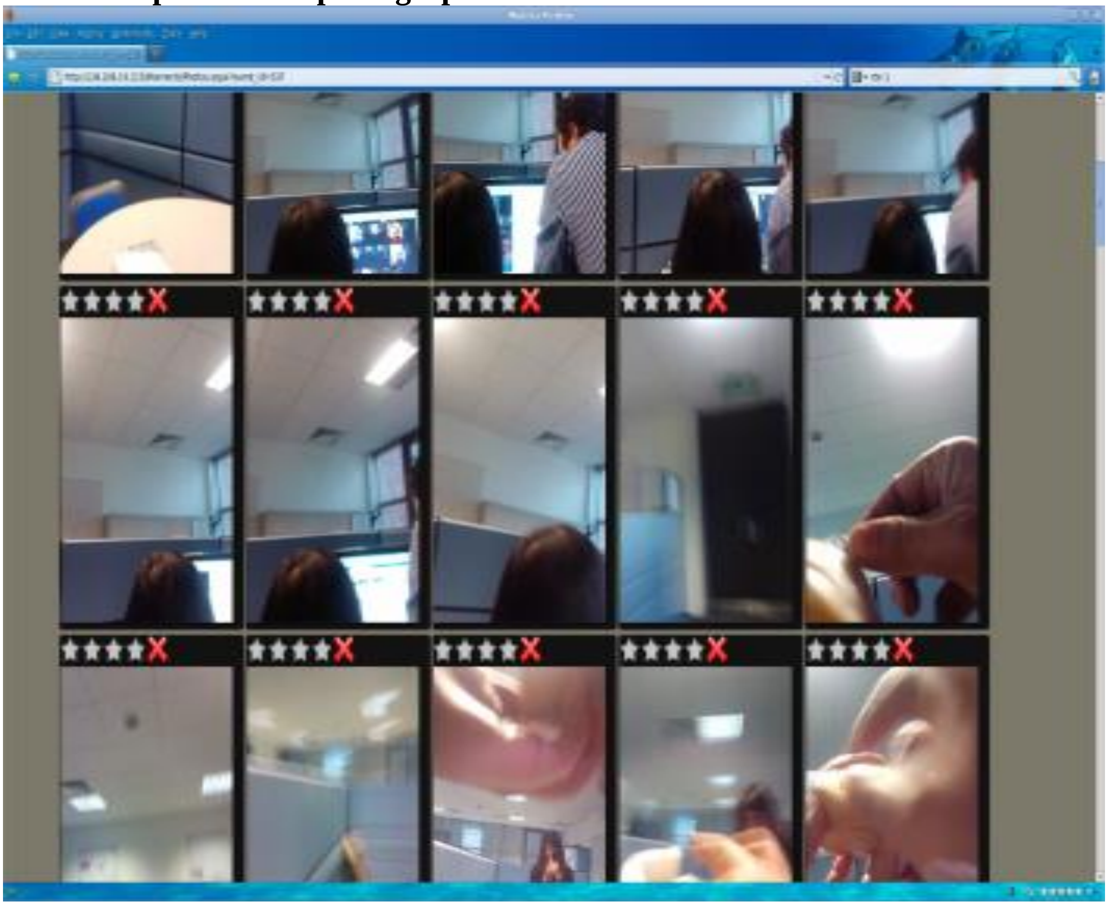
| Rate | Number | Percentage |
|---|---|---|
| 1* | 170 | 26.69% |
| 2** | 224 | 35.16% |
| 3*** | 171 | 26.84% |
| 4**** | 72 | 11.31% |
| Total | 637 | 100% |

**Figure 5.9: The hourly view of the user interface to select the important photographs**



Source: The author (2013)

**Figure 5.10: The photograph view of the user interface which allows them to rate the importance of photographs**



Source: The author (2013)

### 5.8.2 Keyframe Selection Experiments Result Analysis

The previous sections introduced the experiment configuration of keyframe selection. The results on keyframe selection experiments based on different contexts such as social, activity and environment, are now presented.

#### 5.8.2.1 Effect of Social Context on Keyframe Selection

Face-to-face conversation plays an important role in people's daily life. The impact of face in important photograph selection is investigated. This experiment used the "www.face.com" face detection engine which is a technology platform with best-in-class facial recognition software to detect the number of faces in the photographs (Kotsiantis et al., 2007).

As shown in Table 5.6, in 637 important photographs, 75 (11.8%) photographs contain faces. Only 643 (1.1%) photographs containing faces were detected in total 56,201 photographs. It seems that photographs containing faces have a higher chance of being selected as important photographs. An interesting finding is that the proportion of photographs with faces decreases when photograph importance increases. There are 170 one-star important photographs and 31 (18.2%) of them contain faces. While in 72 four-star photographs, only 4 (5.6%) of them contain faces.

**Table 5.6: The importance of photographs with faces**

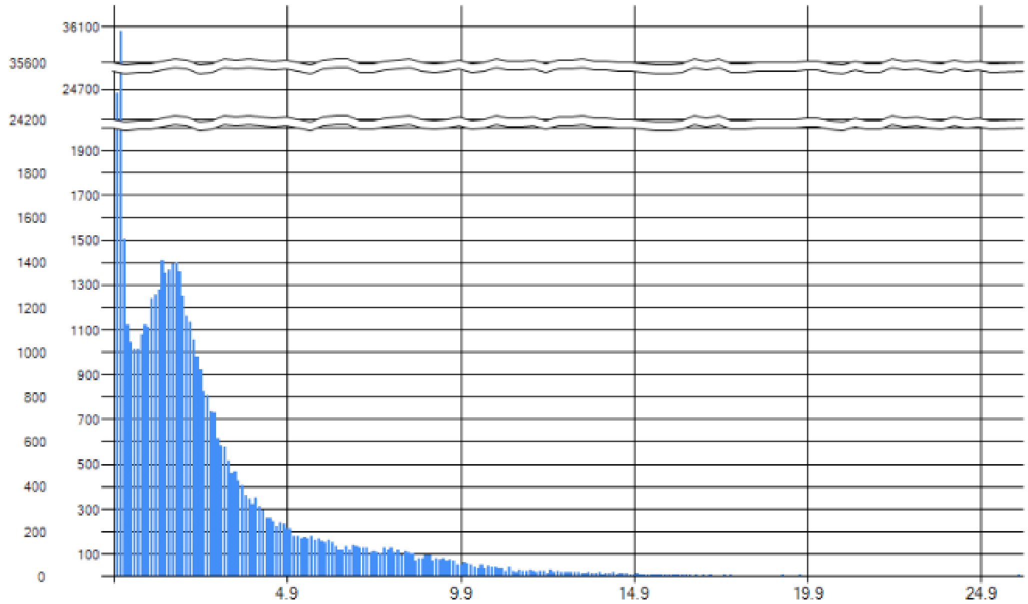
| Importance | Photograph with faces | Photographs |
|---|---|---|
| 1 | 31 (18.2%) | 170 |
| 2 | 24 (10.7%) | 224 |
| 3 | 16 (9.0%) | 177 |
| 4 | 4 (5.6%) | 72 |

#### 5.8.2.2 Effect of Activity Context on Keyframe Selection

The previous chapter (Section 4.1.4) detailed the approaches used to detect four kinds of activities using accelerometer data such as sitting/standing, lying, walking and driving. The quality of photographs is affected by the light level and by the movement of the body. For example, most photographs taken from a wearable camera are blurred when users are running. The standard deviation of the accelerometer's 3-axes is an important attribute to indicate the strength of an activity.

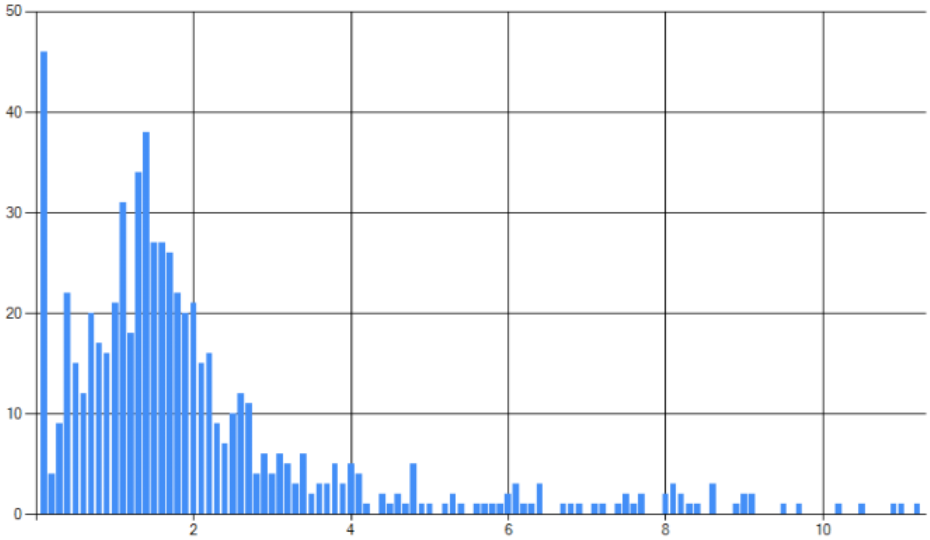
In this experiment, 30 seconds' accelerometer values are taken around each photograph as one unit. The standard deviation for each axis is calculated as one value deviation using the average of 3-axis standard deviation. It is found that the greatest numbers of photographs were taken when the user had no movement or very low strength movement (e.g. sitting/standing or walking) because many of the participants are working with computers for a considerable period of time every day. A very small proportion of important photographs were taken when participants are participating in a low strength activity (e.g. walking) still because of the nature of participants' work which does not require them to participate in low or high strength activities such as walking or running. Therefore, the number of photographs does not depend on activity strength, but indeed, depends on the time length of the activity (e.g. one photograph per 30 seconds).

**Figure 5.11: Distribution of activity strength for all photographs**




Source: The author (2013)

**Figure 5.12: Distribution of activity strength for important photographs**



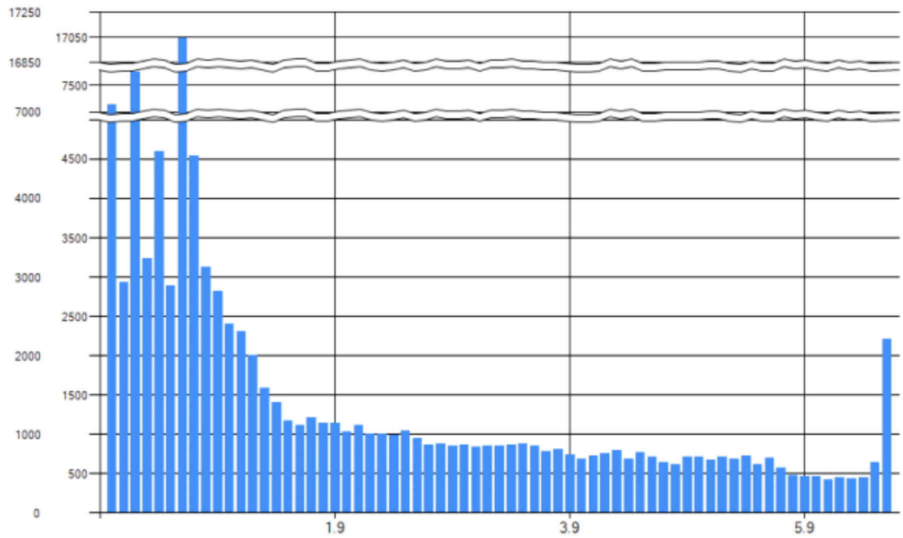

Source: The author (2013)

However, to investigate if the number of important photographs will depend on the activity strength, two figures are presented (Figures 5.11 and 5.12) to visually compare the numbers of important photographs and all of the photographs based on the activity strength. It can be seen that the distribution of important photographs is similar to the distribution of all of the photographs. Both the numbers of important, and all, photographs depend on the duration of activities rather than their strengths. Therefore, no support is found for the effect of activity on keyframe selection.

### 5.8.2.3 Effect of Environment Context on Keyframe Selection

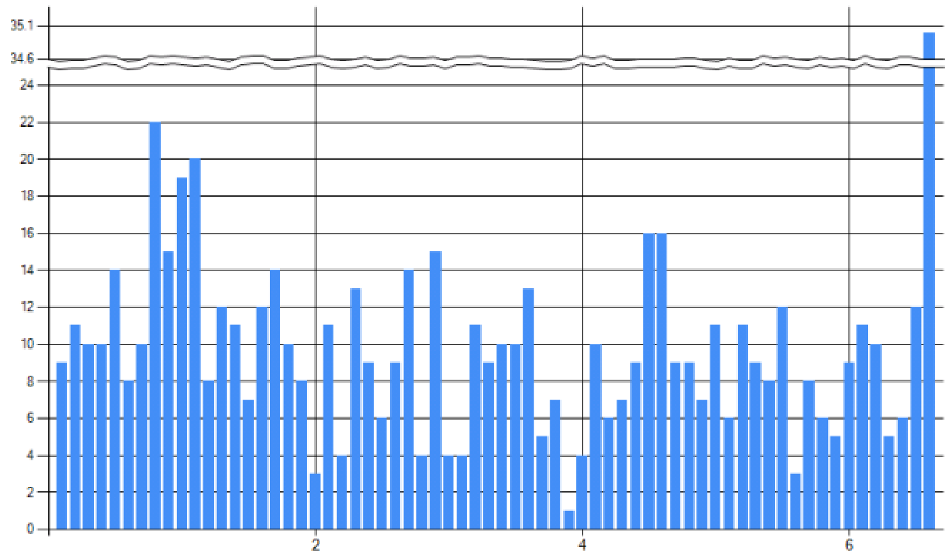
This part reports the results on the effect of environmental noise level on important photograph selection. For example, talking or playing could be identified using environmental noise levels which can be chosen as important. However, there is no direct relationship between important photographs and environmental noise level as shown in Figures 5.13 and 5.14. The environmental noise level can be a better data source to detect conversation when the user is talking to people who are not facing him directly. It may be possible to adjust the frequency of environmental noise reading and to train the corresponding model to detect conversation and then to investigate the impact of conversation on keyframe selecting, but this is not the focus of the work in this thesis.

**Figure 5.13: Distribution of environmental noise level for all photographs**




Source: The author (2013)

**Figure 5.14: Distribution of environmental noise level for important photographs**




Source: The author (2013)

The above results indicate that only social context has an effect on selecting important photographs. Therefore, in this study, an approach to select keyframe considering social context is used. It is outlined in the following steps:

1. Sort all the photographs by image quality.
2. Detect faces from all images in the event. Faces counted in the photographs will not be considered.
3. If some photographs contain faces, choose the best quality ones as the keyframe. If no face is detected in the event, choose the best quality photograph as the keyframe.

## 5.9 Summary

This chapter demonstrated the process of segmenting events which is in the data analysis components in the lifelogging system. Through how to detect the changes in contexts, we standardised the unit and extracted the attributes which is the prior work for segmenting events.

Using SVM, we segmented lifelog data streams into events by detecting the event boundaries. This study also investigated the accuracy of classification using colour, location and multiple attributes and found that the multiple attributes provided the best accuracy for classification.

Keyframe is the representation of a whole event. This study used different contexts such as activity, social, and environment contexts to find the best keyframe. It was found that only the face presence in social context may indicate photograph importance correctly. Based on other research, this study considered face and image quality as key attributes to choosing keyframes.

Overall, in this chapter, we found that context could affect the importance of moments in a person's life. In our lives, there are always some moments being stored in our long term memory. We judge the importance of photographs or moments with context which are collected using our biological sensors such as eyes and ears. The proposed lifelogging system in this study can help to sample such data. It is possible to detect important moments using many varied sources of context. Therefore, we found support for the hypothesis 1. The event segmentation can be performed effectively by detecting changes in sensor data.

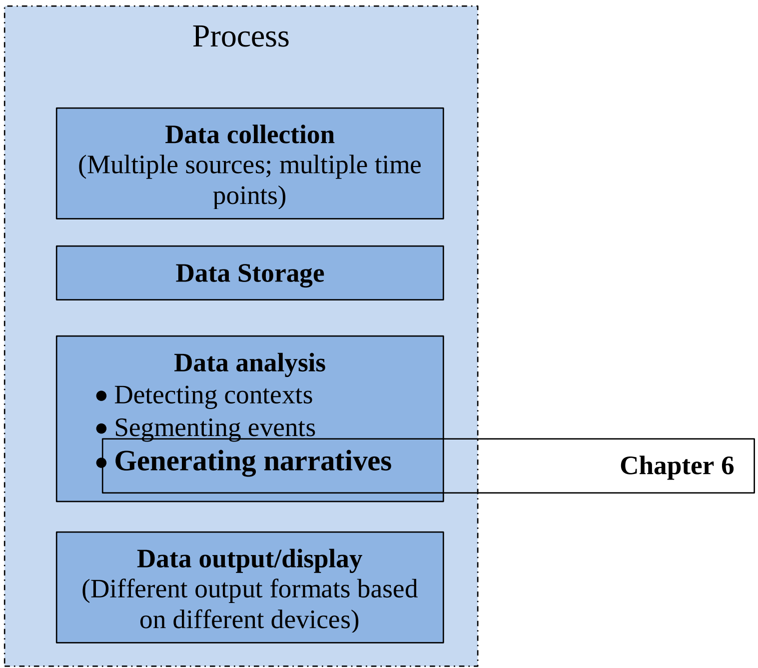
# CHAPTER SIX

# GENERATING NARRATIVES

## 6.1 Introduction

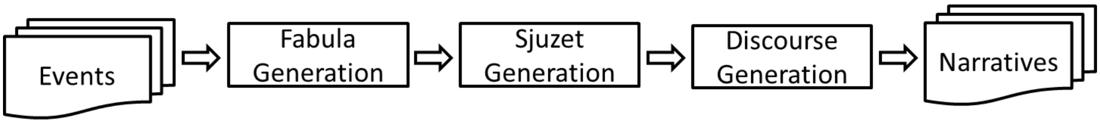
This chapter proposes approaches to generate a narrative for each event in the lifelog. The prior research concepts of fabula, sjuzet and discourse to generate meaningful and effective narrative summaries of events are explored. To give the user a clear narrative, in the process, some important contexts are selected to generate a narrative using virtual sensors. Figure 6.1 shows the position of this chapter's work in the whole model. Figure 6.2 presents the process to generate narratives.

**Figure 6.1: Work in Chapter 6**



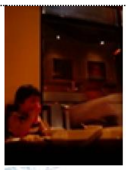

Source: The author (2013)

**Figure 6.2: The process to generate narratives**




Source: The author (2013)

**Table 6.1: An example of generating narratives**

| Stage | Event 1 | Event 2 |
|---|---|---|
| **Fabula** | It was 8.00am on Monday last week.<br>You were at home.<br>You were looking after the children.<br>You were with Tim.<br>It was very noisy. | It was 8.10am on Monday last week.<br>You were walking from home to work place.<br>You were with Anna.<br>It was very noisy. |
| **Sjuzet** | It was 8.00 am on Monday last week. You were looking after the children with Tim at home. It was very noisy. | It was 8.10 am. You were walking with Anna from home to work place. |
| **Discourse** |  It was 8.00am on Monday last week. You were looking after the children with Tim at home. It was very noisy. |  It was 8.10 am. You were walking with Anna from home to work place. |

As shown in Figure 6.2, there are mainly three processes to generate narratives from the detected contexts and segmented events, namely fabula, sjuzet, and discourse generation. In this study, fabula is a series of sentences based on the detected contexts and segmented events; sjuzet is a paragraph of narratives generated from the fabula without the repeated sentences; and discourse is a paragraph of narrative swith an illustrated picture/keyframe taken during the event. To provide a clear picture of these three concepts, we present an example in Table 6.1. It describes two events; at the Fabula generation stage, many simple sentences are generated from the segmented events; a paragraph is generated to describe the event at the

Sjuzet generation stage. Lastly, at the discourse stage, a keyframe, i.e. a representative image for the event, is included with the paragraph description. Then a narrative is generated. An example of one day's narrative generation is shown in Appendix 1.

## 6.2 Generating Fabula

Fabula is the raw material of a narrative, to which we apply selections and transformations in order to generate sjuzet and discourse. To generate a fabula of an event is to answer *"When"*, *"Where"*, *"Who"*, *"What"* and *"How"* questions using simple sentences. In this study, fabula is the story composed by all the contextual knowledge about the user in a single-event time; although it is limited by the available sensors and concepts. In that period of time, some contents may be repeated. To describe the event, some repeated sentences need be removed. In the following subsections, we will discuss the process of selecting contexts and generating the fabula.

### 6.2.1 Selecting Location Context

Location is a very important clue for users to recall their lives, and it has also been shown to be a very efficient way to organise and access personal data (O'Hare et al., 2005*b*). In this study, we use different location sources to calculate a user's location such as GPS, WiFi, Bluetooth and Base Station. However, for people, the addresses are more meaningful. To express location information in user friendly names, we translate all location points to addresses using the Google Geocoding API. Furthermore, important addresses such as home are more meaningful to the user;

hence we use semantic location names instead, which is employed for a small number of semantically meaningful locations.

For many events, there may be a set of locations involved. For example, when a user is driving, many different locations may be collected. However, we do not include all locations in the fabula. For example, when we drive to a place, we usually consider the starting point and the destination as the relevant locations. Therefore, for this type of event, we describe the beginning and end locations if the event involves different locations in the format of *"from"* one place *"to"* another.

### 6.2.2 Selecting Activity Context

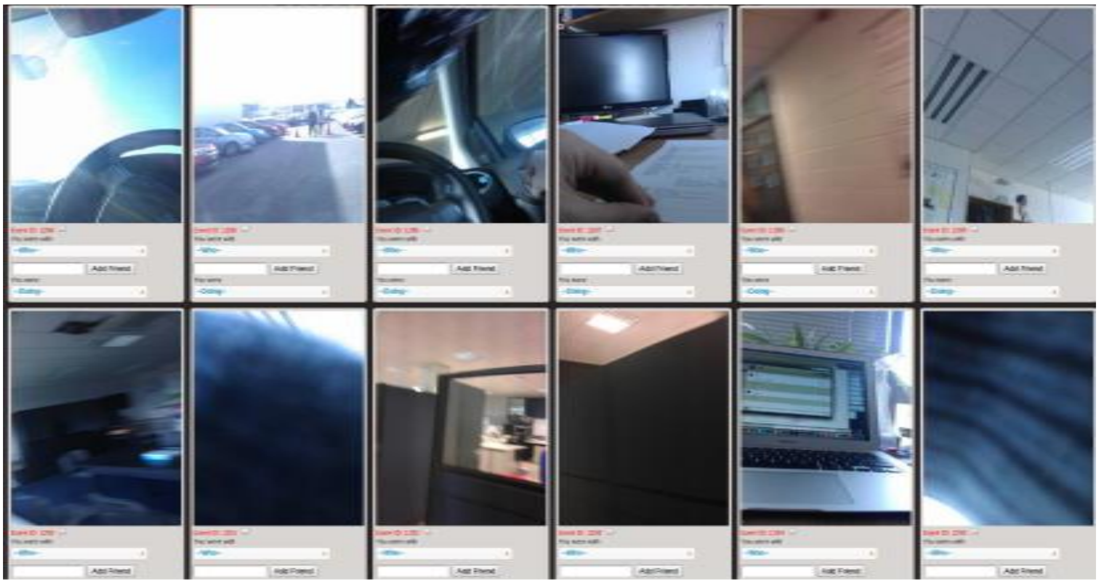
Many activities may happen during one-event period. For example, a user may stand up, get a cup of coffee and sit back down again within one minute. Typically this would not suggest a new event in this study[3], although the example event contains sitting, walking and drinking. This is because if all the changes of activities are described in detail in a narrative summary, a user is likely to lose interest in reading such a large number of long narratives.

In the lifelogging system in this study, an approach of top activity selection is employed. It firstly calculates the frequencies of each activity in one event's period. It then ranks the activities by their frequencies. Finally, the activity with the highest frequency is chosen and used as top activity for this event.

[3] Please note that the addition of one small detail (e.g. taking tablets from pocket before taking a cup of coffee) could make this a very important event. However, as explained above, *if all the changes of activities are described in detail in a narrative summary, a user is likely to lose interest in reading such a large number of long narratives*. Due to this, the current study does not consider these potential important events. It will be a future research direction.

In the system, four main activities can be detected automatically namely sitting, lying, walking and driving. Information on other activities would naturally improve the system performance. Considering this, we designed a web-based interface (see Figure 6.3) for users to manually select more activities. Based on Kahneman et al. (2004), we initially selected 15 most common daily activities into the list for users to choose from as listed in Table 6.2.

**Figure 6.3: Web interface for user to add more information**



Source: The author (2013)

**Table 6.2: The fifteen activities for users to manually choose**

| | | |
|---|---|---|
| Socializing | Relaxing | Pray/worship/ meditate |
| Exercising | Watching TV | Shopping |
| On the phone | Napping | Caring for my children |
| Housework | Working | Commuting |
| Eating | Preparing food | Computer/ e-mail/Internet |

Based on Kahneman et al. (2004)

### 6.2.3 Selecting Social Context

The social context indicates who the user is talking to and engaging with. Based on the face detection technique, the system chooses an image with a face as a keyframe. It also provides labels such as 'friend' when the user is in a social place or 'colleague' when user is in the work place or during working hours.

Ideally, the system will achieve a higher performance if it can tell who the person in the image is. However, the current face recognition technique is not perfect. Adding face recognition into the system may generate more errors which would hinder the focus of this work for supporting multimodal access. To improve the comprehensiveness, we provide an interface (see Figure 6.3) through which the users can input the name of the person whose face is in the photograph. Once a user labels the name of a face, such changes would be detected and new narratives are generated with all relevant indexes updated.

Phone communication is another source from which to select social context. For example, when the user is using their phone to communicate with his/her friends through phone call logs or texts, our system can capture all the phone calls and texts users have sent and received. However, users may feel uncomfortable in seeing their message content in narratives. Therefore, in fabula, the content of a message is not displayed.

To show a user's entire life in a lifelog, the system also provides narrative on the user's behaviour of listening to music. It shows which music the user listened to and when he began to charge his phone in the event. By doing so, the system generates more detailed fabula depending on users' input.

### 6.2.4 Selecting Environment Context

Environments in this study include weather information and noise levels. When selecting environments in generating fabula for an event, this system does not include weather information. The reason for this is that the weather tends not to change dramatically during one event. For the user, they are not interested in reading about weather information in their narratives repeatedly. Therefore, weather information is not added into fabula and narratives.

Noise is an important environmental data source. The environmental noise level can change in a period of an event. In the fabula, the environmental noise level is included if it is higher or lower than normal range. The normal range is the average environmental noise level during an event, across all user events.

### 6.2.5 Building Fabula

Fabula is made up of several sentences. As shown in Table 6.1, one example of fabula for event 1 includes: "It was 8.00am on Monday last week. You were at home. You were looking after the children. You were with Tim. It was very noisy."

Fabula are generated with the following sentences. The detail of pseudocodes for generating fabula is shown in Figure 6.4.

- **Time sentence:** Time sentence is generated using relative time. For example, the events happened in yesterday is described as "Yesterday, at 8:00 am...". If it happened last week, it is described as "On Monday...". In this process, the system considers semantic time first. All semantic time adopted in this study is shown previously in Table 4.1.

- **Location sentence:** Similar to time, location sentence considers the selected semantic location first. For example, narrative uses “You were at home” instead of “You were at Glasnevin, Dublin, Ireland”. During certain events such as driving events where locations change all the time, the system does not generate location sentence, but includes the location sentence in the activity sentence.
- **Activity sentence:** For events happened in different locations, such as driving event, the sentence would be “You drove from home to the work place.” If there is no location changing, e.g. user was sitting, the sentence is “You were sitting.”

**Figure 6.4: Pseudocode for generating fabula**

```
SET time=getTimeContextFromVirtualSensor()
generating time sentence
IF userInputActivity THEN
        SET topActivity=userInputActivity
ELSE
        SET activityList=getListofActivityFromVirtualSensor()
        SET activityCount=length of activityList
        FOR i=1 to activityCount
                correct the wrong detected activity
                // for example, sitting between drivings should be driving, because of
red light
        NEXT
        SET topActivity= the most frequency activity in the event
END IF
SET locationFrom=getStartPointFromVirtualSensor()
SET locationTo=getEndPointFromVirtualSensor()

IF  locationFrom= locationTo THEN
        generating location sentence with one location
ELSE
        generating location sentence using two locations
END IF
IF userInputActivity THEN
        SET socialContext=userInputSocial
ELSE
        SET socialContext=getListofSocialFromVirtualSensor()
        generating social sentence using socialContext
END IF

SET environmentContext=getEnvironmentFromVirtualSensor()
generating environment sentence using  environmentContext
```

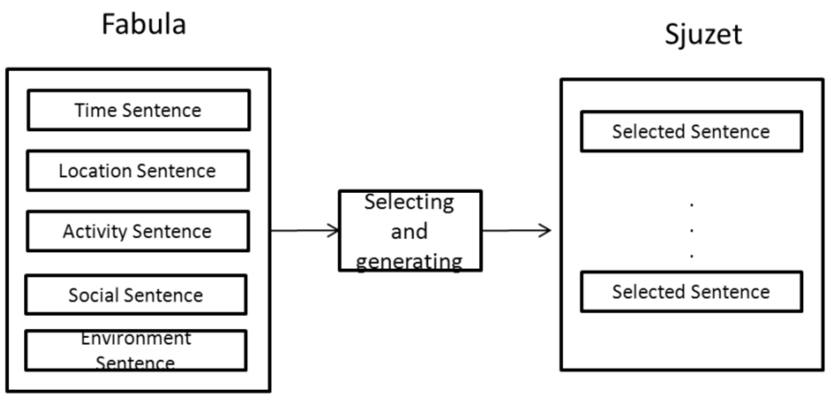
Source: The author (2013)

- **Social sentence:** If the user was speaking to somebody, the activity sentence is generated as “You were sitting and speaking to somebody”. If the location is social place, the sentence would be “You were sitting and speaking to your friend”. If the user answered the phone during the event, the social sentence is “At 8:05 am you received a 3 minutes’ phone call from Anna”.
- **Environment sentence:** As mentioned above, environment is generated when environmental noise level is higher or lower than normal/average level. For example, if it was noisy, the environment sentence would be “It was very noisy”.

## 6.3 Generating Sjuzet

As shown in Figure 6.5, generating sjuzet involves generating a paragraph of narratives based on the fabula (sentences).

**Figure 6.5: Generating sjuzet from fabula**




Source: The author (2013)

As mentioned in previous sections, an event involves at least one activity over a period of time or involves at least one change of context. However, not every context changes from event to event. For example, the user begins to make tea after sitting.

The only changing context is his activity. The location and environment do not change. If everything is described in the narrative, there will be a lot of duplications which would not be the ideal, nor would it appear in anyway.

To avoid duplication, the system generates sjuzet based on some aspects of fabula when it is changed. For example, the location sentence of the previous event and current event are both “You were at home.” This sentence (fabula) is not chosen to generate sjuzet for the current event. This is because the user has already known the location information when he read the previous event narrative. It will be chosen to generate sjuzet when current location is not “at home”.

## 6.4 Generating Discourse

The final step is to generate the discourse, which is the content that is shown to the user. The discourse could be text, photograph and even film (Cheong and Young, 2008). As we mentioned in previous sections, a good quality image would be a very useful addition to a lifelog. In this study, we use the keyframe photograph and the generated sjuzet as the discourse. As shown in the Table 6.1, the discourse for event 1 includes a photograph taken during the event. An example of the process of narrative generation is shown in Appendix 1. The detail of pseudocodes on generating sjuzet is shown in Figure 6.6.

## 6.5 Evaluating Narratives

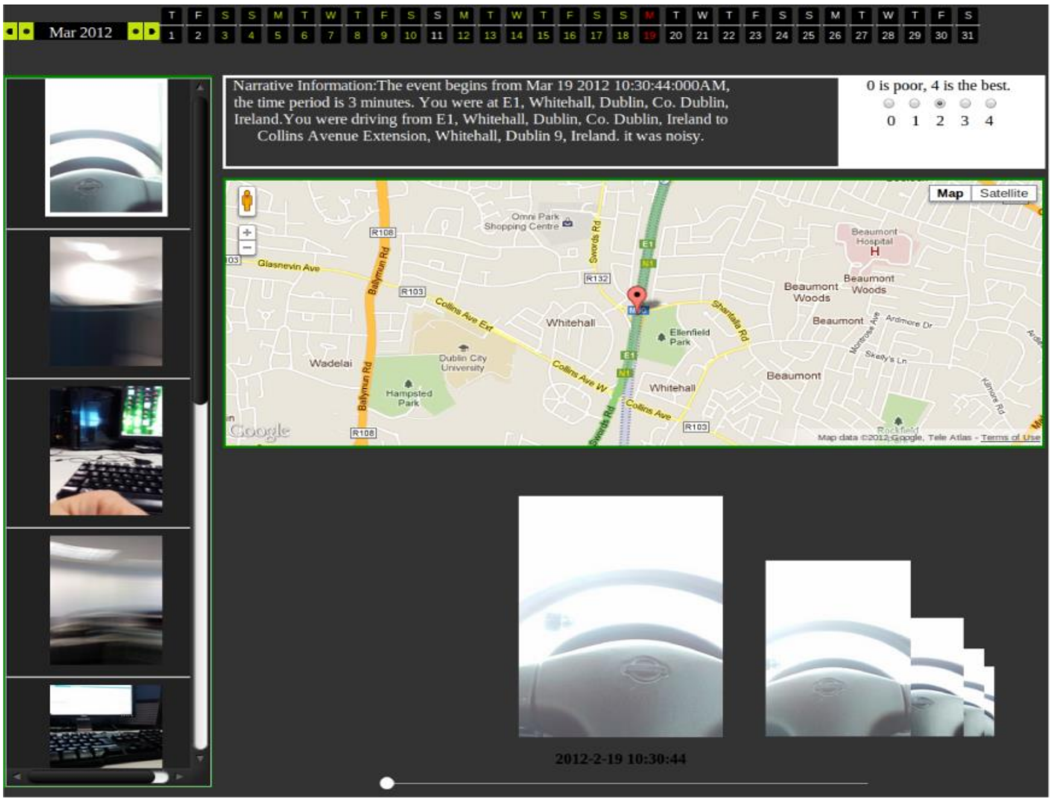
To investigate to what extent the narratives generated from the system can help people recall their past experience, five users were employed to evaluate these narratives. The five users collected data for two weeks. This data was analysed using the proposed lifelogging system. Narratives were generated for them to evaluate through a web-based interface as shown in Figure 6.7.

**Figure 6.6: Pseudocode for generating sjuzet**

```
SET prevousFabula=getPreviousEventFabula()
SET CurrentFabula=getPreviousEventFabula()
SET sjuzet= new Sjuzet()
IF prevousFabula.locationSentence!=currentFabula.locationSentence THEN
   sjuzet.locationSentence=currentFabula.locationSentence
END IF
IF prevousFabula.socialSentence!=currentFabula.socialSentenceTHEN
   sjuzet.socialSentence=currentFabula.socialSentence
END IF
IF prevousFabula.activitySentence!=currentFabula.activitySentenceTHEN
   sjuzet.activitySentence=currentFabula.activitySentence
END IF
IF prevousFabula.environmentSentence!=currentFabula.environmentSentenceTHEN
   sjuzet.environmentSentence=currentFabula.environmentSentence
END IF
sjuzetContent=sjuzet.ToString()
```

Source: The author (2013)

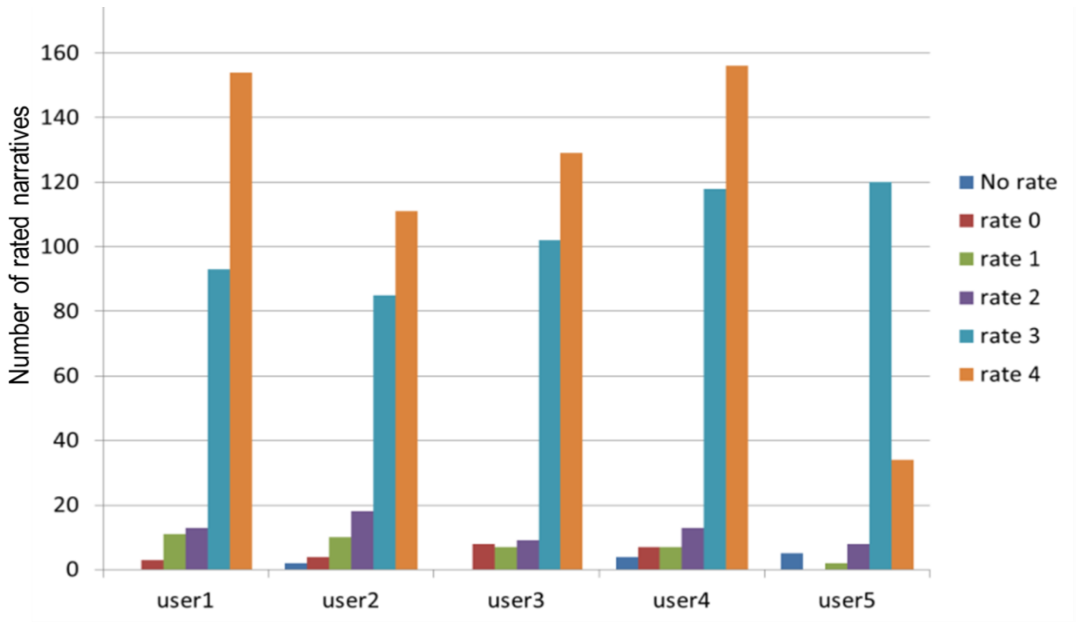
**Figure 6.7: Web interface for participants to evaluate narratives**




Source: The author (2013)

The user interface in Figure 6.7 includes five components. On the top, there is calendar panel with which the user can choose which day they would like to evaluate. On the left hand side, all events that happened in that day are shown through keyframes. When the user clicks a keyframe in the left panel, the narratives generated for the event will be shown in the information panel on the right hand side. A map is shown to provide the location information where the photograph (keyframe) was taken. As well as the narratives on the right hand side, there is a group of radios buttons, which the user can use to rate the accuracy of the event narrative. An accuracy value of zero means that the generated narrative is not useful content at all; and of four means that the generated narrative can give the user the best clue to recall the events. On the bottom right hand side, there is a play button. If the user is not sure of the accuracy of the narrative with the keyframe, they can browse all photographs taken in that event using the play function.

**Figure 6.8: The results of narratives evaluation**




Source: The author (2013)

Overall, the results of the evaluation are positive. All of the five users have provided average scores higher than 3 (user1: 3.4; user2: 3.3; user3: 3.3; user4: 3.6; users5: 3.2). Figure 6.8 presents the results of the number of ratings by 5 users. In the results, most of annotated narratives are rated as 3 and 4. It indicates that the narratives can help the users to recall their past experiences to a great extent.

In the results, user2 (2), user4 (4) and user5 (5) have missed some narratives (the number is shown in the brackets) to evaluate. Compared with the rated ones, the missing ratings are very rare (0.9% for user2; 1.3% for user4; 3.0% for uesr5) and are ignored.

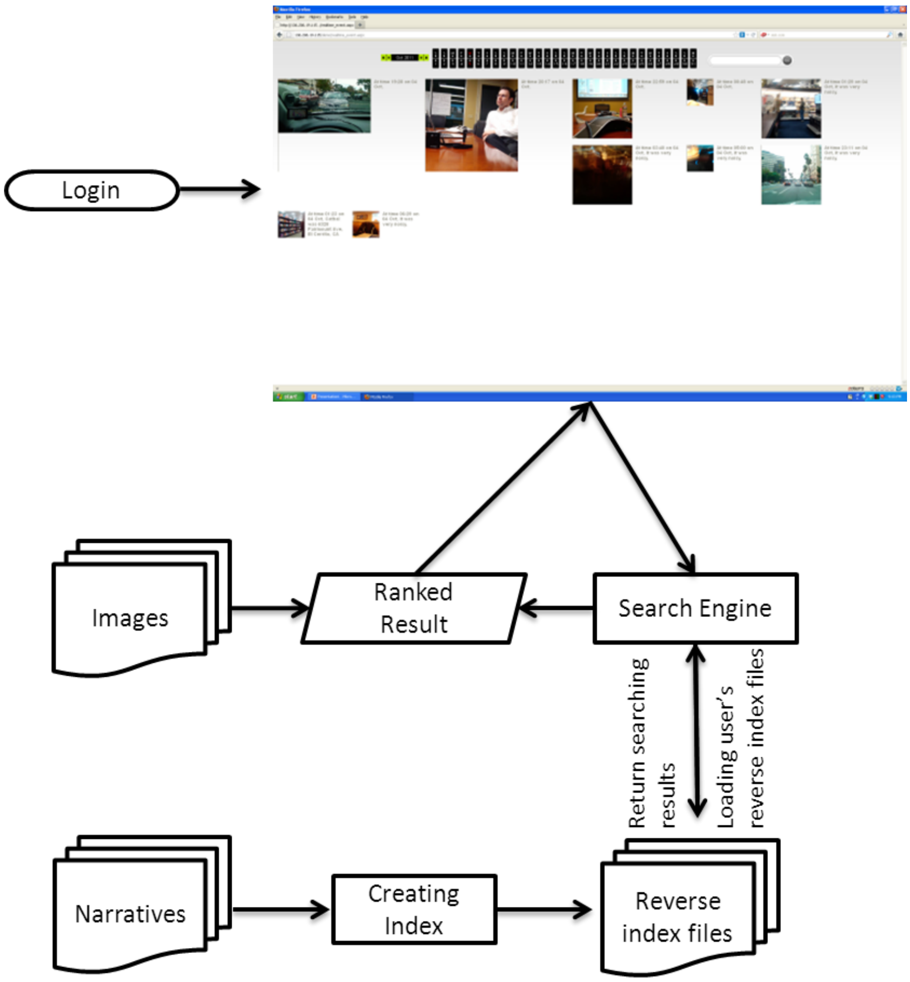
We explore the reason for the low scores to improve system performance in the future. Events we used to evaluate narratives are generated using our event segmentation model. Some (typically short) events may have not enough contextual information, e.g. the example given in Section 6.2.2 (a user stands up, gets a cup of coffee and sits back down again within one minute). The quality of narrative is thus reduced. Since the frequencies of low score are much less than the high score, it is not a serious concern in this study.

The narratives generated from the system are not perfect. However, the experiment results have shown that they are sufficiently enabling users to recall their memories. It has achieved the goal of this study.

## 6.6 Personal Life Experience Search Engine

As an addendum to this work on generating descriptive narratives, we developed a personal life experience search engine. To generate readable narratives, we only chose parts of fabula to generate sjuzet. However, when the user searches his/her personal information, they may expect that all information can be searched. Keeping this in mind, we indexed the fabula using Lucene[4] (Hatcher et al., 2004) which contains more information. As shown in Figure 6.9, a user can search their lifelog data by inputting a textual query string. When users type his/her queries, the system will submit the query to the search engine. After the search engine returns the ranked results, our system will analyse and show them on the web page as a ranked list of events.

**Figure 6.9: The process of searching personal information using search engine**




Source: The author (2013)

[4] Lucene: Apache Lucene is a free/open source information retrieval software library, originally created in Java by Doug Cutting. It is supported by the Apache Software Foundation and is released under the Apache Software License.

## 6.7 Summary

It is our conjecture that users would desire quick automatic diary and simple episode summaries that are easy to understand. In this chapter, we mainly introduced the process of generating narratives using all kinds of concepts, i.e. fabula (sentences), sjuzet (paragraphs) and discourse (paragraphs with keyframes/pictures).

Narratives can help users to recall their past experiences. To test to what extent the narratives can help people recall their past experience, an experiment was set up for users to rate the usefulness of the narratives. The results showed great support for using the narratives to recall users' memories. Therefore, support is found for hypothesis 2. A meaningful textual narrative that accurately represents an event can be generated automatically.

In addition, we developed a personal life experience search engine through which users can search from all of the collected information rather than only the essential contexts, events and narratives generated from the system.

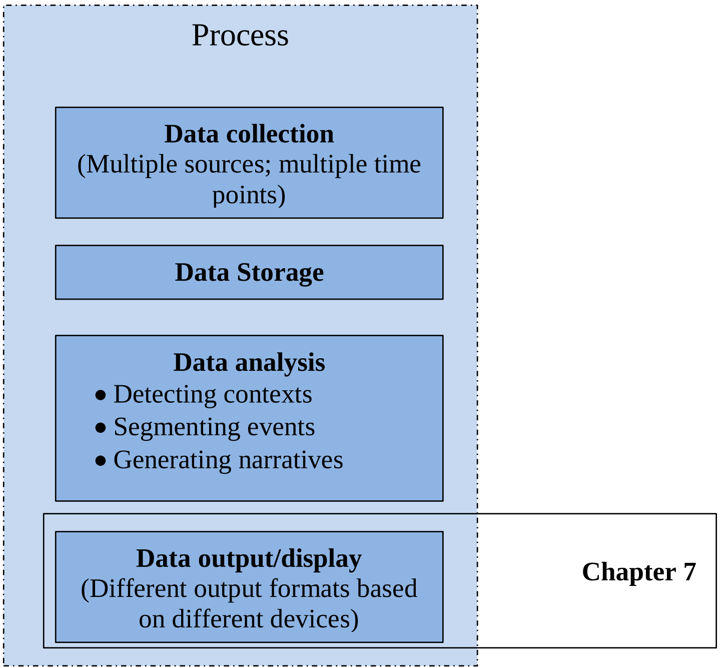
# CHAPTER SEVEN

# SUPPORTING MULTIMODAL ACCESS

## 7.1 Introduction

The previous chapters have presented data collection, storage and analysis including context detection, event segmentation, and narrative generation. In this chapter, we focus on the last process of the lifelogging system - data display. We will explore how to best present lifelog data on multiple accessing devices, i.e. multimodal access to lifelogs. Three devices for representing results are applied as computer, smartphone and E-book reader. Results from the user experience experiments indicate different data displaying performance for different lifelogging devices. Figure 7.1 shows the position of this chapter's work in the whole model.

**Figure 7.1: Work in Chapter 7**




Source: The author (2013)

## 7.2 Overall User Experience Experiments and Survey

In order to evaluate what types of output from lifelogging system are the most suitable for displaying lifelog data analysis results on which of the devices (computer, smartphone and E-book reader), we set up a few user experience experiments.

### 7.2.1 Dataset

We firstly chose a previous user's lifelog data (10-day segment) after receiving his permission to use it. This lifelog data was processed and annotated by the data owner. It includes the four main activities detected from the system as sitting/standing, lying, walking and driving. It also includes the manual selection from the 15 activities as shown in Table 6.1. Example activities include socializing, eating, and shopping etc. From the ten days' data, the event segmentation technique identified 253 events. Keyframes were extracted and narratives are generated. To save the participants' time, 16 events were selected by the data owner based on his own experience and preference as query topics as shown in Table 7.1.

### 7.2.2 Participants

In this study, we adopted a convenient sampling strategy which is very common in the lifelogging domain (e.g. Schmidt et al., 2012). 16 participants were recruited to investigate the performance of different interfaces on different devices. 11 participants are male and 5 are female. They are from different research groups. They are derived from a wide variety of research areas including information retrieval (3), engineering (2), HR (2), chemical (2), data mining (2), video processing (3), biological (1) and cloud computing (1).

**Table 7.1: The sixteen questions for user interface test**

| Queries | Topic |
|---|---|
| Q1 | Find out how many times the user used computer on the $20^{th}$. |
| Q2 | Find out the time when the user left home on the $12^{th}$. |
| Q3 | Find out the time when the user was relaxing at home on the $16^{th}$. |
| Q4 | Find out the time when the user was talking to J on the $11^{th}$. |
| Q5 | Find out all the events of the user was working with A on the $9^{th}$. |
| Q6 | Find out the scene of the user was relaxing at home on the $10^{th}$. |
| Q7 | Find out the scene of the user was having a conversation with an unknown person on the $16^{th}$. |
| Q8 | Find out the time when the user was meeting R on the $10^{th}$. |
| Q9 | Find out scene of the user was working with a whiteboard on the $11^{th}$. |
| Q10 | Find out the colour of the user's jumper on the $16^{th}$. |
| Q11 | Find out how often the user used a computer on the $11^{th}$. |
| Q12 | Find the scene of a bedroom on the $9^{th}$. |
| Q13 | Find out the time when the user was driving to work at on the $18^{th}$. |
| Q14 | Find out how many different people the user interacted with on the $16^{th}$. |
| Q15 | Find out the time when the user was relaxing at home on the $16^{th}$. |
| Q16 | Find the event of the user using the internet on the $10^{th}$. |

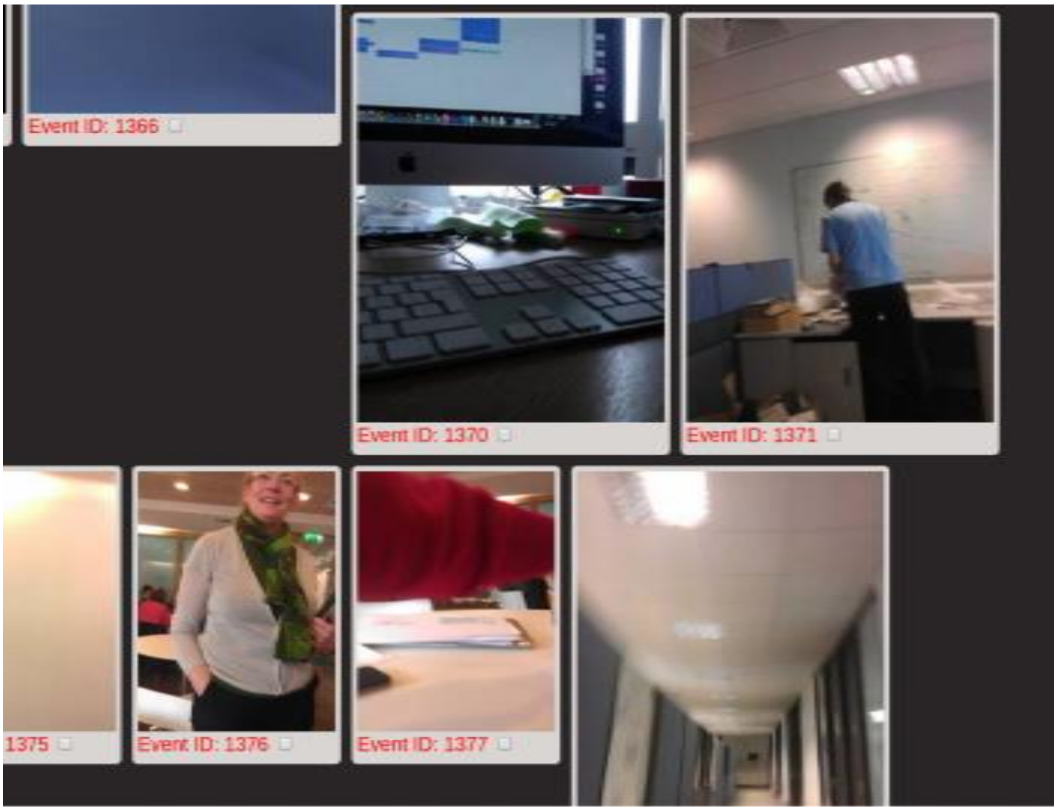
The participants rarely know about the data owner in terms to his life, friends, or places in which he regularly spends time. In this way, the impact (bias) of participants' knowledge and experience about the data owner on the results is avoided. Although the sample may not be truly representative of the population, which may be obtained through random sampling, this method was suitable for this study to provide a general understanding of people's use of technology.

### 7.2.3 Eight User Interfaces for Displaying Lifelogs

Eight different types of user interfaces for displaying lifelogs are developed for this work including images, images and annotations, images and icons, images and narratives, animations, diaries, icons, and narratives. They are described in detail as below.

- **Images:** In this interface, all keyframes of events are selected and shown on the webpage. Users can access them by computer or smartphone. On this interface, all events are loaded and ordered by time. The interface is shown in Figure 7.2. This interface is the typical interface for lifelogging users. It is used as our baseline interface.

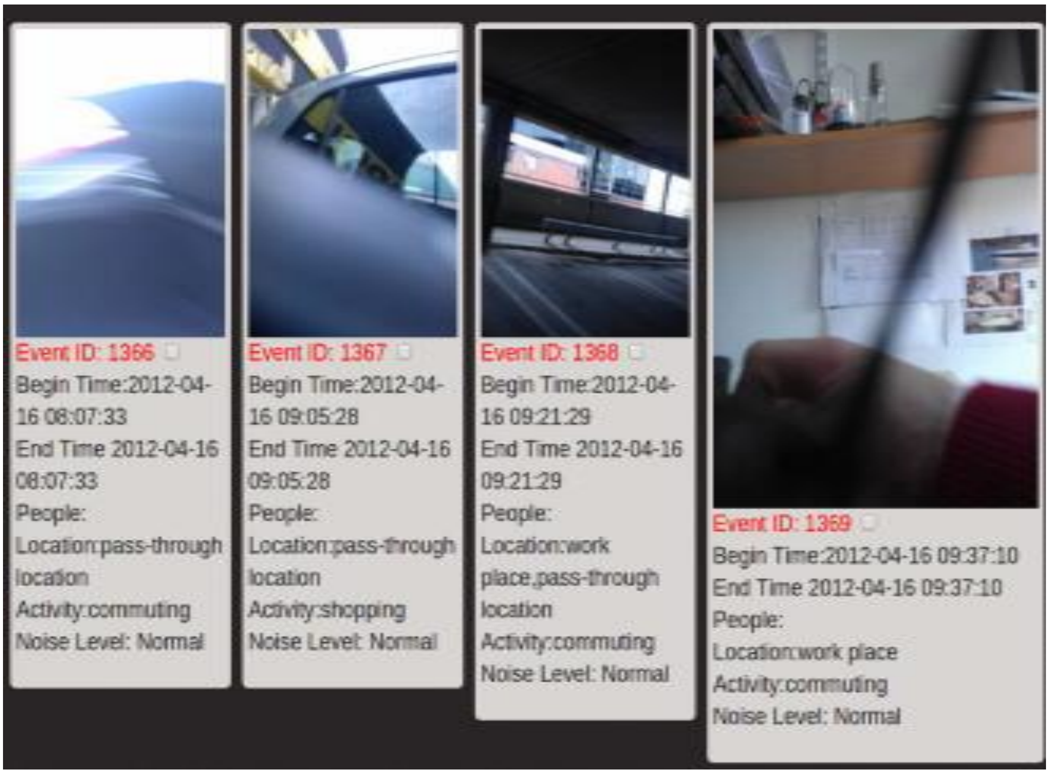
**Figure 7.2: Showing user's everyday life event with images**




Source: The author (2013)

- **Images and annotations:** The interface does not only contain keyframes of all events, but also the additional information, such as the user's activity and event time. The user interface is shown in Figure 7.3. In this case, the raw semantic annotations are accompanying the event and this would represent a minimal addition to the baseline image-only interface.

**Figure 7.3: Showing user's everyday life event with images and annotations**

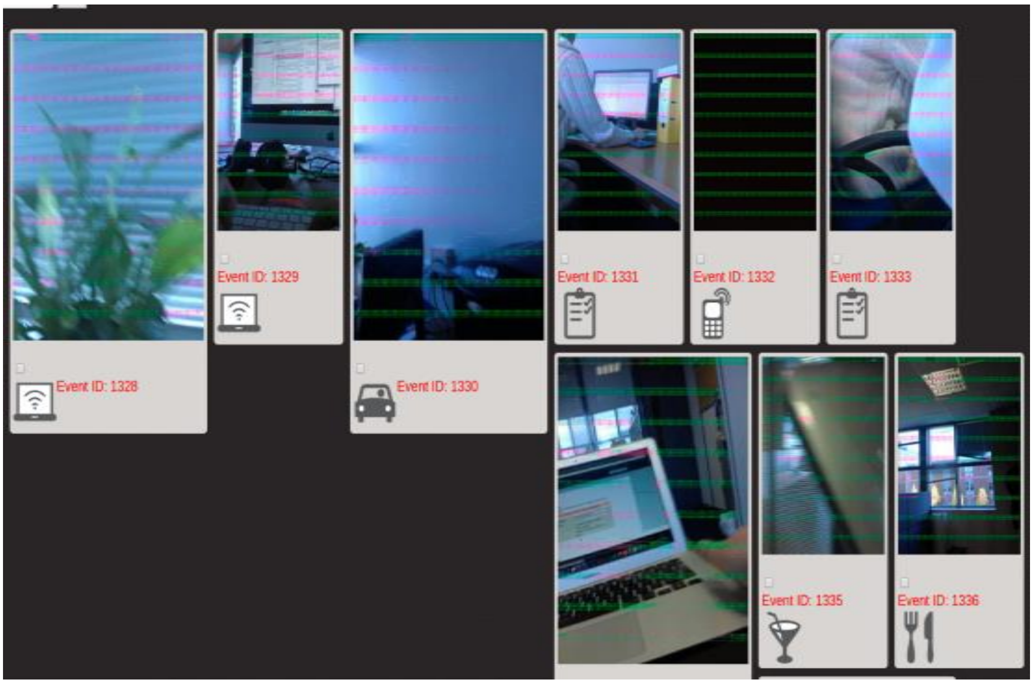



Source: The author (2013)

- **Images and icons:** In this interface, all users' activities are replaced by icons, and they are shown with keyframes in Figure 7.4. The icon provides a quick reference visual cue to represent the 15 important life activities that we mentioned in the previous chapter and were shown previously in Table 6.1.

- **Images and narratives:** The interface contains keyframes and generated narratives as shown in Figure 7.5. All current day's events are shown in one page. In this case, we are interested to know if the narratives are liked by the participants or found to be useful in locating relevant events faster.

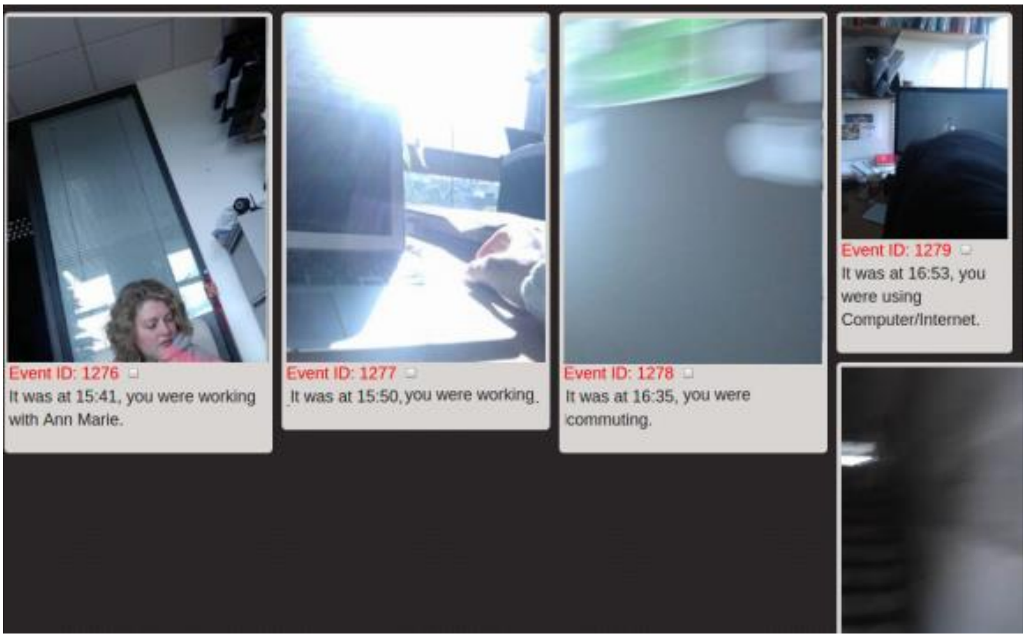
**Figure 7.4: Showing user's everyday life event with images and icons**




Note: The icons are designed for 15 activities.

Source: The author (2013)

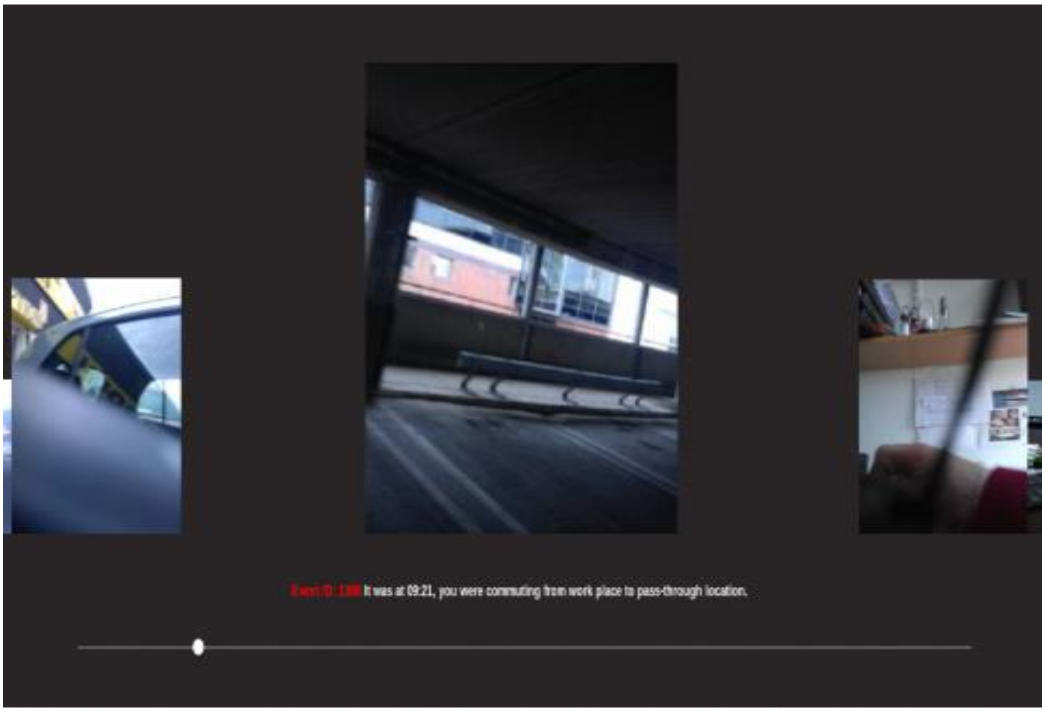
**Figure 7.5: Showing user's everyday life event with images and narratives**




Source: The author (2013)

- **Animations:** This interface shows all the day's events with keyframe and related narratives in animation mode from beginning to the end. The time span between two events is 500 ms. The UI is shown in Figure 7.6 and would be useful in a less-interactive, lean-back environment, such as on a TV or other large screen, relaxation-focused device.

**Figure 7.6: Showing the user's everyday life event with animations**

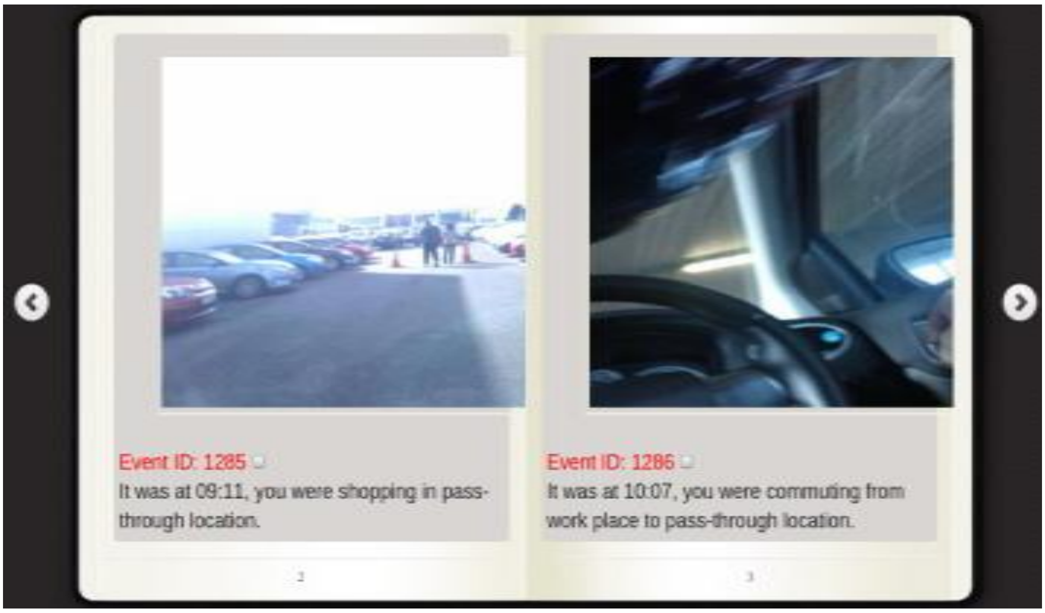



Source: The author (2013)

- **Diaries:** In this UI, keyframes and related narratives are shown in a diary style similar to a story book. User can view the previous or next event by clicking buttons. The interface is shown in Figure 7.7. This interface attempts to recreate the concept of a human diary of a day's events, with the selected keyframe.

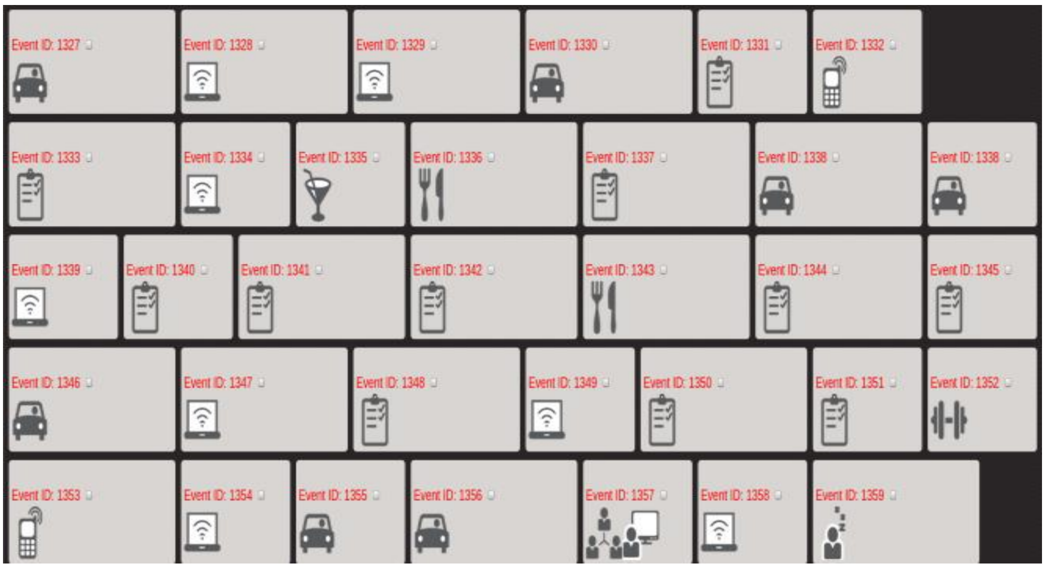
- **Icons:** There are only icons shown in this UI as shown in Figure 7.8. In this interface, we are evaluating if semantic annotations alone can help to locate relevant events fast. The icon list would provide a small visual summary of a day/an event suitable for small screen devices, such as a watch or ipod nano, or even on a next generation wearable device like Google Glass.

**Figure 7.7: Showing the user's everyday life event with diaries**




Source: The author (2013)

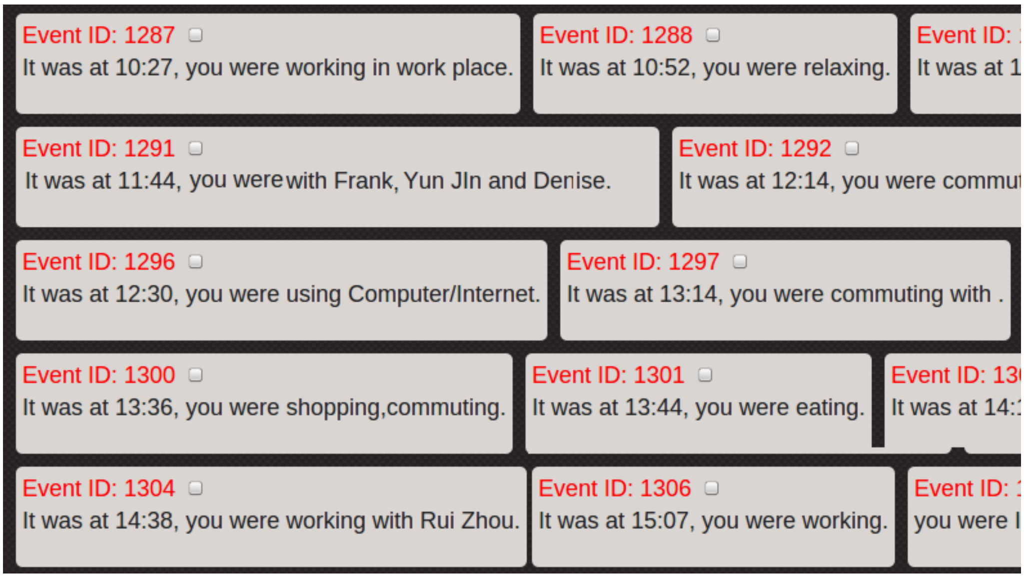
**Figure 7.8: Showing the user's everyday life event with icons**




Source: The author (2013)

- **Narratives:** There are only narratives shown on the web page as shown in Figure 7.9. All current day's events will be shown in one page. This narrative description is replicating the concept of the basic diary entry, per event, with no accompanying keyframe. It is particularly suitable for generating summaries of periods of time, e.g. days or weeks, where a keyframe selection methodology has not yet been developed to be effective over long time periods.

**Figure 7.9: Showing user's everyday life event with narratives**




Source: The author (2013)

## 7.2.4 Evaluating Display Performance

When participants answer the questions using the above eight user interfaces, their accuracy and time length is recorded as one source to evaluate the performance of the eight user interfaces on the tested devices.

In addition, we adopted an online survey method to ask the participants to evaluate each user interface. Four criteria are included, i.e. visual appeal, subjective satisfaction, potential for errors and speed of use. Visual appeal is believed to dominate impression judgments (Lindgaard et al., 2011). The other three criteria (effectiveness, efficiency and satisfaction) are defined by the international standard ISO/IEC 9241-11 and have been used by a lot of researchers, e.g. Shneiderman and Plaisant (2005).

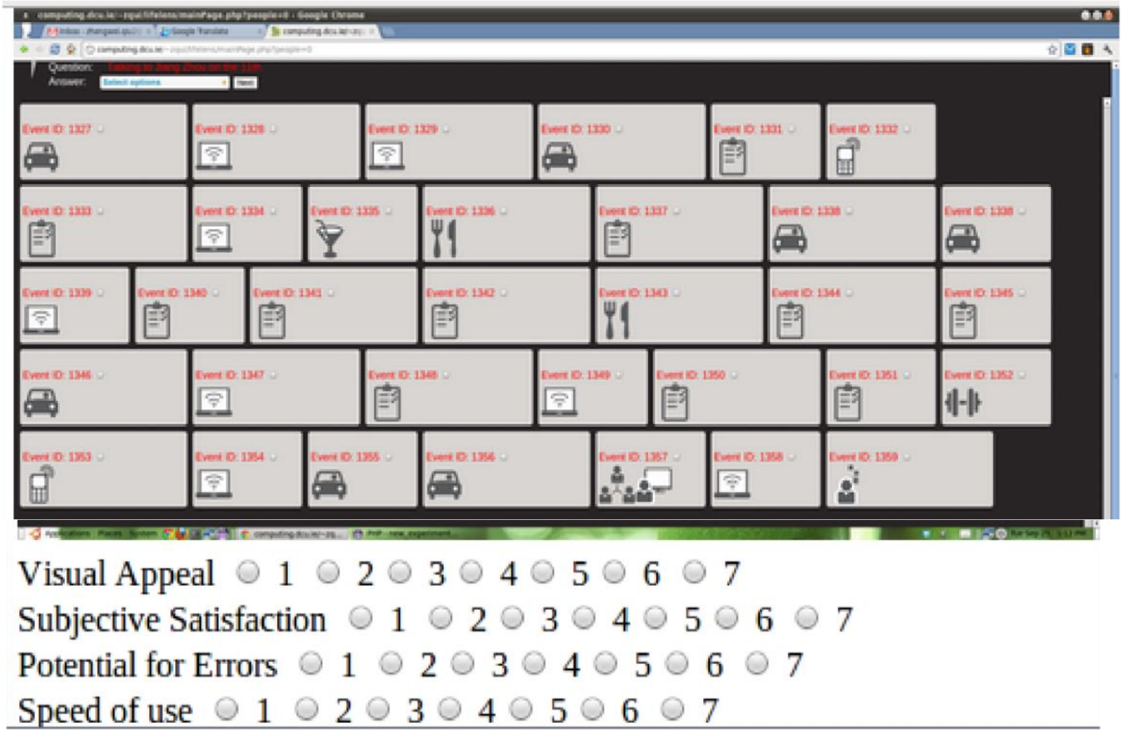
The survey was distributed to participants immediately after the experiment was completed. For each user interface, the participant was presented with the screenshot of the user interface. A seven-point Likert scale was used (1 = the lowest; and 7 = the highest).

The Likert scale was named after Dr. Rensis Likert, a sociologist at the University of Michigan. He developed this technique and published it published in the *Archives of Psychology* in 1932 with a title of "*A Technique for the Measurement of Attitudes*" (p.1-55). Since then, the Likert scales have been used to measure responses' experience and feelings through survey/questionnaire method in many research domains, e.g. psychology (e.g. Gong et al., 2010), business management (e.g. Fu et al., 2013) etc. The advantages of Likert scales include 1) they are quicker and more economical to collect data comparing with other data collection method e.g. interview; 2) they are easily adapted to various measurement situations such as from agreement to satisfaction; 3) the data are very easy to be analysed using statistic software. Meantime, some researchers criticize the Likert scales. For example, Jamieson (2004) outlines some common means to abuse Likert scales in practice. They include that the intervals between Likert scale values are not equal although they have a rank order. Despite the disadvantages of Likert scales, many studies have used them. For example, in human-computer interaction field, Shneiderman et al. (2010) adopted Likert scales in a survey to evaluate the user interface satisfaction.

Similar to Shneiderman et al. (2010), this study applies Likert scales to evaluate the user interface satisfaction.

It is noted that the "potential for errors" criterion is reverse coded which means the larger the value is, the less errors and the better performance the user interface has. As the questionnaire took approximately 5-10 minutes to complete and questions were very simple, all participants completed their questionnaire properly. Figure 7.10 presents the interface example for evaluation display performance.

**Figure 7.10: An example of web interface to evaluating display performance**




Source: The author (2013)

### 7.2.5 Procedures

As presented in Section 7.2.2, 16 participants were recruited to carry out the experiments, i.e. evaluating the potential value of the eight user interfaces on the three devices. The eight user interfaces (UIs) include images, images and annotations, images and icons, images and narratives, animations, diaries, icons, and narratives. Three devices are the computer, smartphone and the E-book reader.

As the three devices have different characteristics, not all of the eight user interfaces are shown for each device. For example, the E-book reader has a low displaying capability in white and black. Only four UIs are used and evaluated by participants for this device: “Images”, “Images and annotations”, “Images and narratives” and “Narratives” while all eight UIs are used in computer and smartphone. Table 7.2 presents what UIs are used for each device.

For the 16 participants/users for the experiments on three devices, eight participants evaluated the display performance of computer; four participants evaluated the smartphone and the remained evaluated the E-book reader. All participants completed the experiment only once. It meant that a participant who evaluated all sixteen questions on the computer will not be faced with the same topics again on the smartphone or the E-book reader.

Considering the interest and time of the participants, it was not necessary to answer all of the 16 questions for each UI. For example, the participant 1 in computer display answered two questions (Q1 and Q9) on images UI, two questions (Q2 and Q10) on image and annotation UI. The same method is applied to other participants for the other two devices. Another reason for applying this method is due to questions themselves and the devices. For the user interfaces on both computer and smartphone, participants can answer topics by clicking on the related event on the interface. For example, to answer the question Q4 (Talking to J on the $11^{th}$), participants only need to click on the related photograph or narrative, and the system will log the answer and the spent time. For some questions such as Q9

(Working with a Whiteboard on the 11th), the participant may not find the answer using “Icons” UI. In this situation, the participant can just skip that topic. Once the topic is answered, the timer will stop and results will be logged.

Performance was measured based on entry speed and accuracy, as well as a survey on four criteria (visual appeal, subjective satisfaction, potential for errors and speed of use). The survey was given to all participants at the end of each session to obtain subjective feedback.

**Table 7.2: The eight interfaces and their deployment in the experiment**

| UI | Computer | Smartphone | E-book Reader |
|---|---|---|---|
| Images | Y | Y | Y |
| Images and Annotations | Y | Y | Y |
| Images and Icons | Y | Y | N |
| Images and Narratives | Y | Y | Y |
| Animation | Y | Y | N |
| Diaries | Y | Y | N |
| Icons | Y | Y | N |
| Narratives | Y | Y | Y |

## 7.3 User Experience with Computer Display

The experiment setup on user experience with computer display is presented along with the results in the following subsections.

### 7.3.1 Experiment Set Up for Computer User

In this experiment, we employed eight participants to find the answers for all sixteen questions shown in Table 6.1 and using eight user interfaces with computer.

As mentioned in the previous section, considering the interest and time of the participants, not all of the 16 questions are requested to be answered for each UI. For example, participant 1 in this experiment answered two questions (Q1 and Q9) on images UI, two questions (Q2 and Q10) on images and annotations UI. More details are shown in Table 7.3.

**Table 7.3: Experiment configuration for computer user interface test.**

| Participant | Images | Images and Annotations | Images and Icons | Images and Narratives | Diaries | Animations | Icons | Narratives |
|---|---|---|---|---|---|---|---|---|
| 1 | Q1<br>Q9 | Q2<br>Q10 | Q6<br>Q14 | Q3<br>Q11 | Q4<br>Q12 | Q5<br>Q13 | Q7<br>Q15 | Q8<br>Q16 |
| 2 | Q2<br>Q10 | Q3<br>Q11 | Q7<br>Q15 | Q4<br>Q12 | Q5<br>Q13 | Q6<br>Q14 | Q8<br>Q16 | Q1<br>Q9 |
| 3 | Q3<br>Q11 | Q4<br>Q12 | Q8<br>Q16 | Q5<br>Q13 | Q6<br>Q14 | Q7<br>Q15 | Q1<br>Q9 | Q2<br>Q10 |
| 4 | Q4<br>Q12 | Q5<br>Q13 | Q1<br>Q9 | Q6<br>Q14 | Q7<br>Q15 | Q8<br>Q16 | Q2<br>Q10 | Q3<br>Q11 |
| 5 | Q5<br>Q13 | Q6<br>Q14 | Q2<br>Q10 | Q7<br>Q15 | Q8<br>Q16 | Q1<br>Q9 | Q3<br>Q11 | Q4<br>Q12 |
| 6 | Q6<br>Q14 | Q7<br>Q15 | Q3<br>Q11 | Q8<br>Q16 | Q1<br>Q9 | Q2<br>Q10 | Q4<br>Q12 | Q5<br>Q13 |
| 7 | Q7<br>Q15 | Q8<br>Q16 | Q<br>Q12 | Q1<br>Q9 | Q2<br>Q10 | Q3<br>Q11 | Q5<br>Q13 | Q6<br>Q14 |
| 8 | Q8<br>Q16 | Q1<br>Q9 | Q5<br>Q13 | Q2<br>Q10 | Q3<br>Q11 | Q4<br>Q12 | Q6<br>Q14 | Q7<br>Q15 |

Note: Participants did not need to find answers using all UIs. For example, Participant 1 used “Images” to find answers for Q1 and Q9, and used “Images and Annotations” to find answers for Q1 and Q9. They only need to answer all questions once.

In this experiment, each interface was encountered twice by each participant; for different questions. To avoid bias, we ensured that the topics were not repeated on the same interfaces and the users encountered the interfaces in different sequences. When the experiment begins, the web page with timer will load appropriate events to the interface according to the question. The timer is stopped and results are stored in the database when user clicks the photograph or other content for the question.

### 7.3.2 Results of Computer User Experiment

Figures 7.11 presents the experiment results generated from the computer-based interfaces on accuracy, time and the four criteria (visual appeal, subjective satisfaction, potential for errors and speed of use) using eight user interfaces (UIs). The eight user interfaces are images, images and annotations, images and icons, images and narratives, animations, diaries, icons, and narratives.

In terms of accuracy, the results generated from the computer-based interfaces indicate that both “images and narratives” and “images and annotations” have the highest scores - 0.94. The “images and icons”, “diaries” and “images” are 0.88, 0.88 and 0.80 as shown in Figure 7.11. Three UI’s accuracy levels are found to be lower than the average score (0.79). They are the “narratives” (0.75), “animate” (0.69) and “icons” (0.44).

In terms of time spent by computer participants on UIs, the results show great difference ranging from 7.4 to 15.8 seconds. The “images” takes the shortest time at 7.44 seconds, which is followed by “images and narratives” (9.75), “images and annotations” (11.5), leaving the other five UIs taking longer than the average time which is 11.97 seconds: “animation” (11.56),“images and icons” (12.3), “icons” (13), “diaries” (14.43), and “narratives” (15.75).

Figure 7.11 also presents the participants’ experience of the eight UIs on four criteria (visual appeal, subjective satisfaction, potential for errors and speed of use). The potential of errors is reversed coded which means the larger the value is, the less errors and the better performance the user interface has.

**Figure 7.11: Evaluation results of displaying performance with computer**


Source: The author (2013)

For "images and icons", "images", "image and narrative" and "images and annotations", participants share similar patterns on the four experience indicators, where the scores on visual appeal and speed of use are higher than the ones with subjective satisfaction and potential for errors. For "animations" and "icons", participants provide the highest scores on visual appeal and similar scores for the rest of three indicators which are all above 3. For "narrative", the highest score is given on potential for errors (>4), leaving the other three indicators at similar level ranging from 2.7 to 3.2. For the last UI "diary", participants provide highest score on visual appeal (6.38), which is followed by subjective satisfaction (5.8), speed of use (4.5), and potential for errors (1.6).

It can be seen that the scores on the visual appeal are the highest (>5) for each UI except for "images" and "narratives". Participants are most subjectively satisfied with "diaries" (>6), and least with "narratives" (<3). Participants think the "narratives" has least (>5) and the "diary" has the most (<3) potential for errors (reversed coding). In terms of participants' experience on speed of use, the scores are generally high (>4) for all UIs expect for "icons" and "narratives".

## 7.4 User Experience with Smartphone Display

The experiment setup on user experience with smartphone display is presented along with the results in the following subsections.

### 7.4.1 Experiment Set Up for Smartphone User

Smartphone touch screen interfaces are more challenging due to the limited screen space (Chittaro, 2006). However, the smartphone is the most convenient tool for users to access their data. Due to its ubiquitous nature, the smartphone is the device that is most likely to be used for many lifelog accesses in the real-world. In the smartphone experiment, we used the same UIs as the computer UI experiment. HTML5 was used to enhance the interface specifically for the smartphone's 4 inch screens, as shown in Figure 7.12.

**Figure 7.12: User interfaces showing on smartphone**

Source: The author (2013)

**Table 7.4: Experiment configuration for smartphone user interface test**

| Participant | Images | Images and Annotations | Images and Icons | Images and Narratives | Diaries | Animations | Icons | Narratives |
|---|---|---|---|---|---|---|---|---|
| 1 | Q1<br>Q9 | Q2<br>Q10 | Q6<br>Q14 | Q3<br>Q11 | Q4<br>Q12 | Q5<br>Q13 | Q7<br>Q15 | Q8<br>Q16 |
| 2 | Q2<br>Q10 | Q3<br>Q11 | Q7<br>Q15 | Q4<br>Q12 | Q5<br>Q13 | Q6<br>Q14 | Q8<br>Q16 | Q1<br>Q9 |
| 3 | Q3<br>Q11 | Q4<br>Q12 | Q8<br>Q16 | Q5<br>Q13 | Q6<br>Q14 | Q7<br>Q15 | Q1<br>Q9 | Q2<br>Q10 |
| 4 | Q4<br>Q12 | Q5<br>Q13 | Q1<br>Q9 | Q6<br>Q14 | Q7<br>Q15 | Q8<br>Q16 | Q2<br>Q10 | Q3<br>Q11 |

Note: Participants did not need find answers using all UIs. For example, Participant 1 used "Images" to find answers of Q1 and Q9, and used "Images and Annotations" to find answers for Q2 and Q10. They only need to answer all questions once.

In this experiment, we only employed four participants as shown in Table 7.4. Each participant answered 16 questions using the eight different interfaces; and the timeout for smartphone participant is 60 seconds. The experiment configuration details are shown in Table 3.2.

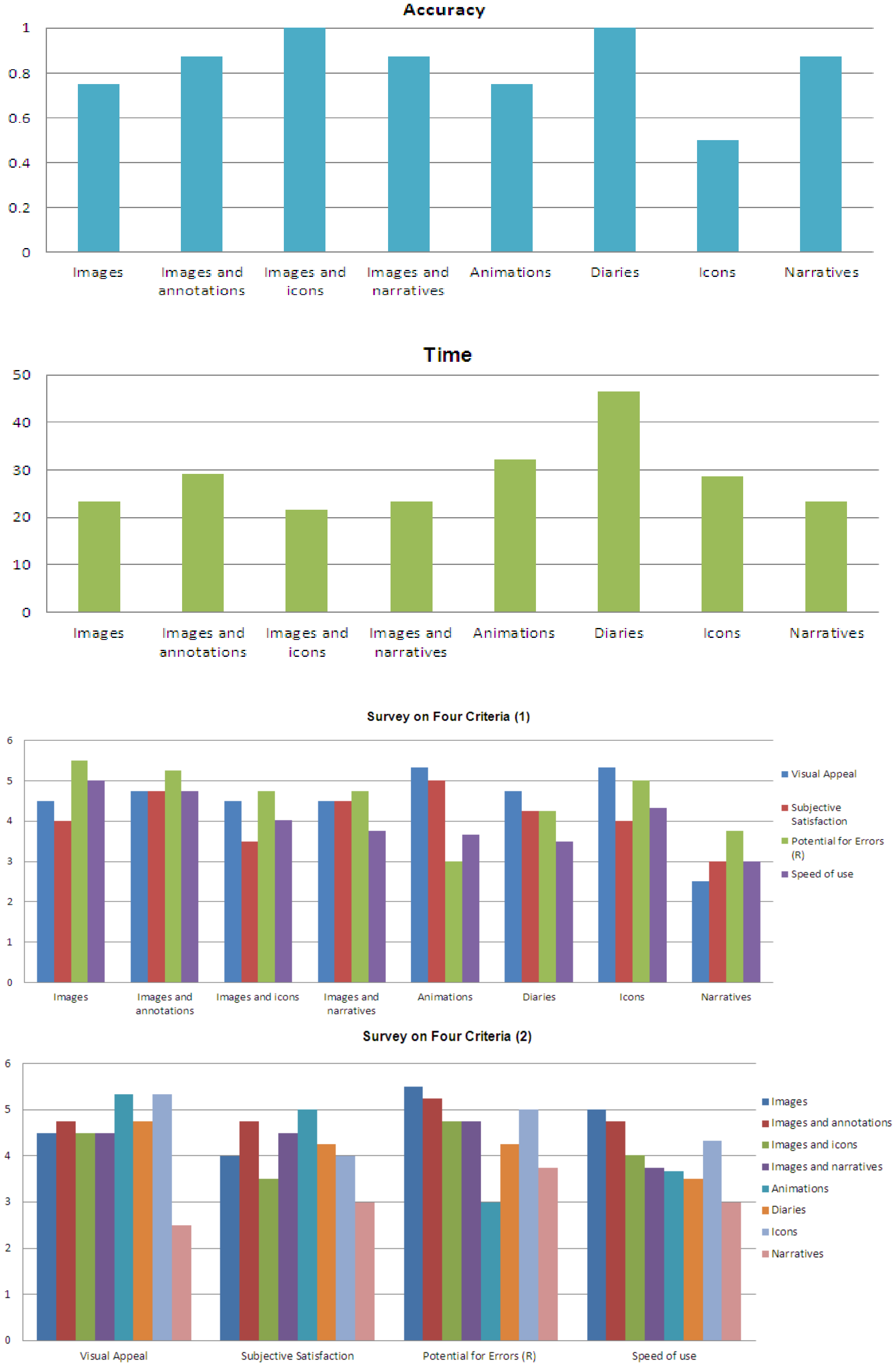
### 7.4.2 Results of Smartphone User Experiment

Figures 7.13 presents the experimental results generated from the smartphone-based interfaces on accuracy, time and the four criteria (visual appeal, subjective satisfaction, potential for errors and speed of use) in relation to the eight user interfaces (UIs).

In terms of accuracy, the results generated from smartphone indicate that both “images and icons” and “diaries” have the highest scores as 0.94, which are followed by “images and narratives”, “narratives” and “images and annotations” with both scores of 0.88, and “images” and “animations” with score of 0.75. The “icons” has the lowest score on accuracy which is 0.50.

In terms of the time spent by computer participants on UIs, the results show huge difference ranging from 21.5 to 46.6 seconds. The “images and icons” takes the shortest time which is 21.5 seconds while the “diaries” takes the longest time which is 46.6 seconds. The “animations” takes the second longest time (32.3 seconds), which is followed by “images and annotations” (29.3 seconds) and “icons” (28.7 seconds). For the rest of the three UIs, i.e. “images”, “images and narratives” and “narratives”, they take similar time at 23.8 seconds. Overall, the time spent by smartphone participants on each UI is longer than the time spent by computer participants.

**Figure 7.13: Evaluation results of displaying performance with smartphone**




Source: The author (2013)

One potential reason is that the smartphone has limited screen space and it can only show one photograph each time. Therefore, it may take the participants more time to find the answers according to our prior study.

Figure 7.13 also presents the participants' experience of the eight UIs on visual appeal, subjective satisfaction, potential for errors and speed of use.

For "animation", "dairies" and "icons", the visual appeal has the highest score. For "images", "images and icons", "images and annotations", and "images and narratives", their potential errors performance are better than the other three indicators. For "images and narratives" and "images and annotations", participants provide similar scores on visual appeal and subjective satisfaction at around 4.5, and highest score on potential for errors (4.9 for "images and narratives" and 5.2 for "images and annotations"). For "narratives", the potential for errors is given the highest score (3.8) with the other three indicators sharing the similar scores (2.5 for visual appeal, and 3 for subjective satisfaction and speed of use). For "diaries", the scores on four indicators stay similar ranging from 3.5 to 4.5 (3.5 for speed of use, 4.2 for potential for errors, 4.2 for subjective satisfaction and 4.5 for visual appeal).

In addition, it can be seen that the scores on visual appeal are the highest (>4) for all of the UIs except for "narratives". Participants are most subjectively satisfied with "animation" (>5), and least satisfied with "narratives" (3). Participants think the "animation" has most potential for errors (3) and the "images" has the least (5.5) potential for errors. In terms of participants' experience on speed of use, the scores are quite diverse ranging from 3 for "narratives" to 5 for "images".

## 7.5 User Experience with E-Book Reader Display

The experiment setup on user experience with the E-book reader display is presented along with the results in the following subsections.

### 7.5.1 Experiment Set Up for E-Book Reader User

The Barnes & Noble Nook (styled “Nook" or “NOOK") is a brand of electronic-book reader developed by American book retailer Barnes & Noble, based on the Android platform (Source from Wikipedia). Nook works like a diary book, and it does not support animation. It can display images, but in white-black mode. Images on E-book reader could not supply as much information as on computer and smartphone. Especially for small images such as an icon, Nook could not display them clearly. Therefore, in the experiment on an E-book reader, we only adopted four UIs: “images”, “images and annotations”, “images and narratives” and “narratives”. We generated different pdf files containing the four UIs.

- **Images:** There is only one keyframe on every pdf page. Every page represents a new event and it is presented to the participants with the correct day at the beginning of the topic.
- **Images and Annotations:** There is one keyframe and related annotation on every pdf page.
- **Images and Narratives:** There is one keyframe and related narrative on every pdf page.
- **Narratives:** There are only narratives of many events.

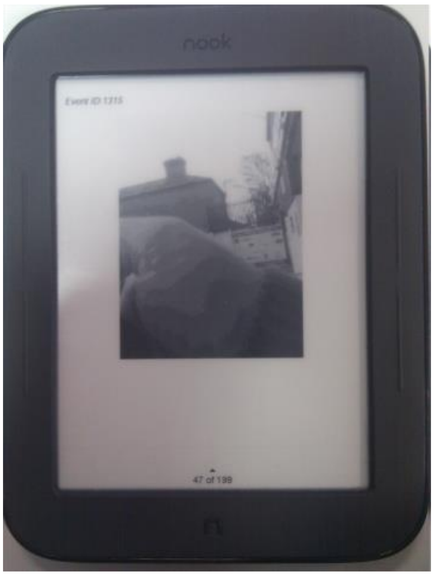
With these files, participants can access lifelogs easily as shown in Figure 7.14. In the experiment, we employed four participants. Two participants answered the first eight questions using four different pdf files (user interfaces), and other two participants found the answers for the other eight questions. We also developed a timer system. It could record the time of the user spent on searching for the answer. The details of the experiment are shown in Table 7.5. Similar to the above experiment with computer and smartphone, users' behaviour will be logged into the database as well.

**Table 7.5: Experiment configuration for E-book reader user interface test**

| Participant | Images | Images and Annotations | Images and Narratives | Narratives |
|---|---|---|---|---|
| 1 | Q1 | Q2 | Q3 | Q4 |
| | Q5 | Q6 | Q7 | Q8 |
| 2 | Q4 | Q3 | Q2 | Q1 |
| | Q8 | Q7 | Q6 | Q5 |
| 3 | Q9 | Q10 | Q11 | Q12 |
| | Q13 | Q14 | Q15 | Q16 |
| 4 | Q12 | Q11 | Q10 | Q9 |
| | Q16 | Q15 | Q14 | Q13 |

Note: Participants did not need find answers using all UIs. For example, Participant 1 used "Images" to find answers for Q1 and Q5, and used "Images and Annotations" to find answers for Q2 and Q6. They only need to answer all questions once.

**Figure 7.14: Showing generated lifelog pdf file with E-book reader**




Source: The author (2013)

### 7.5.2 Results of E-Book Reader User Experiment

Figures 7.15 presents the experiment results generated from the E-book reader-based (Nook) interfaces on accuracy, time and the four criteria (visual appeal, subjective satisfaction, potential for errors and speed of use) using four user interfaces (UIs). These include Images, Images and annotations, Images and narratives, and Narratives.

In terms of accuracy, the results generated from Nook indicate a stable accuracy for four UIs at around 0.82.

In terms of the time spent by Nook participants on UIs, the results show a huge range from 48 to 95 seconds. The “images” takes the longest time which is 95 seconds, which is followed by “images and annotations” with 73 seconds and “images and narratives” with 53 seconds, leaving the “narratives” with the shortest time which is 48 seconds. Overall, the time spent by Nook participants on each UI is longer than the time spent by both computer and smartphone participants. One potential reason is that the Nook needs time to refresh a new page. Therefore, it took participants longer to find the answers.

Figure 7.15 also presents the participants’ experience of the four UIs on visual appeal, subjective satisfaction, potential for errors and speed of use. Participants have very different experience on each UI.

For “images”, “images and annotations” and “narratives”, participants provide highest score on potential for errors (R) (3.5 for “images”, 4.7 for “images annotations” and 6 for “narratives”). For “images and narratives”, participants provide the highest scores to both are performance with low potential errors (5) and subjective satisfaction (5), followed by visual appeal (4.5) and speed of use (3.7).

**Figure 7.15: Evaluation results of displaying performance with the E-book reader**


Source: The author (2013)

For “image”, the lowest score is given to subjective satisfaction (2.2). For “images and annotations”, participants provide similar scores on the three indicators with 3.5 for the speed of use and 3.2 for the visual appeal and subjective satisfaction.

In addition, it can be seen that the scores on the visual appeal are higher for “images and narratives” and “narratives” (both at 4.5) than “images” and “annotations” (both at 3.2). This applies to the subjective satisfaction where the scores for “images and narratives” (5) and “narratives” (5.5) than “images” (2.2) and “annotations” (3.2). Participants think the “images” has the greatest (3.5) and the “narrative” has the least (6) potential for errors, leaving the other two UIs at the similar level (around 5). For the speed of use, participants think the “narratives” is the fastest (5.2) while the “images” is the slowest (2.8), leaving the other two UIs at the similar level (around 3.5). Given the while-black and text-based features of Nook, the results are not surprising.

## 7.6 Overall Findings

In the experiments, the computer has been shown to be the fastest approach to access personal lifelogs. It took the participants 12 seconds to find the answer on average on the computer. On the smartphone and E-book reader they took 29 and 67 seconds respectively. This is because their screen sizes are much smaller than the computer’s screen. Both the smartphone and E-book reader can only show one image at a time. For Nook, it needs an even longer time as it needs to refresh content to avoid flashing. However, compared with the computer, smartphone or E-book reader is more ubiquitous. It can be brought anywhere. For the computer experiment participants, “Images” is the fastest UI, as participants did not need to read other information. The computer screen can show several images at once, so the participants can make their decision quickly. For Nook participants, they could not receive the same quality of information from images. Some information will be lost when images are downsized to black-white mode. However, textual narrative could work well.

Due to its limited display space, the smartphone participants have to scroll down to see more information in the user interfaces containing images. However, the “Animations” UI does not require user to scroll down manually. This is why smartphone got the best feedback on “Visual Appeal” and “Subjective Satisfaction”. The “Animation” UI can satisfy most of participants. For the smartphone and computer, the “Narratives” UI is the slowest and with lowest accuracy. However, it was voted as the best one on E-book reader. In the experiment, the E-book reader only supports black-white. The images were also compressed with quite low quality, causing issues for the participants in locating the desired information quickly. However, E-book readers are designed for text documents; hence it can support “Narratives” well. The narratives use smaller screen space compared with images. These made “Narratives” as the most suitable content for E-book reader user.

In the lifelogging system in this study, we chose different UIs for users to access their data on different devices. We usually have different aims to access lifelog data. For example, sometimes we may view a lifelog to search for an important event, and sometime we just want to browse our daily life. Hence in the lifelogging system in this study, we chose different UIs for users according to different aims. As shown in Table 7.6, we choose the “images” UI for computer, because it can help users to browse their data quickly. The “diaries” UI on computer suits users who want to read their lifelogs. This is because computer display has higher visual appeal and subjective satisfaction, and lowest potential for errors. When users want to do a search on UI, “images and narratives” will suit them more.

**Table 7.6: The suggested user interfaces for different devices**

| | **Optional interface** | **Reasons to use** |
|---|---|---|
| 3*Top 3 interfaces for Computer | Diaries | Highest visual appeal and subjective satisfaction, lowest potential for errors |
| | Images | Fastest UI for computer participants |
| | Images and icons | The second fastest UI, highest accuracy, reasonable visual appeal, subjective satisfaction, potential for errors and speed of use |
| 3*Top 3 interfaces for Smartphone | Images and icons | Highest accuracy, fastest UI |
| | Animations | Highest visual appeal and subjective satisfaction |
| | Images and annotations | The second highest visual appeal and subjective satisfaction, reasonable speed and accuracy |
| 2*Top 2 interfaces for E-book reader | Narratives | The fastest, highest accuracy, visual appeal and subjective satisfaction, lowest potential for errors |
| | Images and narratives | The fastest, highest accuracy, visual appeal and subjective satisfaction, lowest potential for errors UI with images |

For smartphone users, “images and icons” is the fastest UI. Therefore, we choose it for users to browse their lifelog data on a smartphone. Compared with the computer, the smartphone is not easy to scroll. We therefore chose “animations” for users to view their daily lifelog data. Because the smartphone can only show one reasonably sized image on the screen at a time, it is not easy for user to view their data event by event. We suggest that the user uses “images and annotations” if they want to access other information on phones.

“Narratives” has shown great potential in helping users to access their lifelog data using an E-book reader. We therefore suggest the “narratives” UI for E-book reader users. Images on an E-book reader do not contain as much information as on the computer and smartphone. For users who want to access their image data through an E-book reader, we suggest users use “images and narratives” to access their images according to high accuracy, visual appeal and subjective satisfaction.

## 7.7 Summary

In this chapter, we presented the last process of the proposed lifelogging system, the representation of results. To assist users accessing their lifelogs, we investigated the display performance on three devices (computer, smartphone and E-book reader) via eight user interfaces (UIs, as images, images and annotations, images and icons, images and narratives, animations, diaries, icons, and narratives).

The results from these experiments indicate different UIs should be proposed for different devices. For example, “diaries”, “images”, and “images and icons” are the best UIs for computers to display lifelog data; “images and icons”, “animations”, and “images and annotations” are the best UIs for smartphone to display lifelog data; and “narratives” and “images and narratives” are the best UIs for E-book reader to display lifelog data. Based on these findings, we found support for the hypothesis 3. Different access devices benefit from different representations of lifelog data.

# CHAPTER EIGHT
# DISCUSSION

## 8.1 Overview

The aim of this study is to develop a lifelogging system allowing users to automatically capture their daily lives. In this system, multiple sensors' data are firstly collected through smartphones. To save battery life, the data is temporarily stored in the SD card in the smartphone and then transmitted to the server only when the phone is being charged. On the server side, the raw data is in a different format and has different sampling frequencies. Virtual sensors are used to fuse these data. Six contexts are detected from the data: personal, time, location, activity, social and environment contexts. Events are then segmented based on context changes. A keyframe (picture) is selected to represent each event. Narrative is generated based on detected contexts, segmented events and selected keyframes. Finally, different user interfaces (UIs) are adopted to display lifelog results on different devices based on the display performance evaluation results.

From this work, we have shown that software sensors can be employed to detect meaningful semantics from raw wearable sensor data, either single sensors or multiple sensors together. A meaningful textual narrative representing an event can be generated automatically. In addition, different access devices benefit from different representations of lifelog data.

Overall, we have developed a lifelogging system supporting the multimodal access to lifelogs.

## 8.2 Contributions

This study introduces a new lifelogging system which includes multiple sensors' data collection, large volume data storage, data analysis founded on detecting contexts, segmenting events, generating narratives, and the representation of results to users. By doing so, this study contributes to the lifelogging research in the following ways.

Firstly, in this study, a new generation of lifelogging tool has been developed to collect, store, analyse and display lifelogging data automatically. It does not need user's input. Users just carry their smartphone and review the data output afterwards. Therefore, it avoids some users' fear about lacking of professional background to get involved in lifelogs.

Second, the lifelogging tool has been designed to collect a full range of sensor data from a smartphone in a power-efficient manner. For example, the system is context-aware. It can learn the user's situation and decide which sensor needs to be turned on or off in order to maintain all-day data capture.

Third, approaches are found to extract semantic contexts from raw sensors using term weighting and support vector machine learning (SVM) techniques. This helps to bridge the semantic gap between the human and machine. Six contexts are detected: personal, time, location, activity, social and environmental context.

Fourth, a real-time lifelogging system is developed to analyse lifelog data containing face detection and uploading data to a server in real-time functions. With that, users can easily browse and share their status using a web browser.

Fifth, a method is presented to obtain a user's location using a fusion of GPS, WiFi, Bluetooth and Base Station providing more accurate location information. The combination of multiple location sensors also dynamically extended the smartphone battery life, compared with using GPS alone.

Sixth, a new approach is found to segment lifelog stream data into events using SVM. SVM has a good generalisation ability as it is based on the principle of the structural risk minimisation in statistical learning theory. This provides a very good reference method for future researchers working on lifelogs.

Seventh, a tool is implemented to detect users' activities (sitting/standing, lying, walking and driving) based on accelerometer sensor. This helps to reduce the semantic gap in lifelog data analysis.

Eighth, an approach is designed to generate a narrative of event using all the contexts extracted from physical and virtual sensor data. It improves the comprehensiveness of the data usage and accuracy in data analysis.

Lastly, a multi-access model to access lifelogs is established based on different devices. Suggestions are offered on the most suitable representations to enable fast and effective access to lifelog data using different modalities of access.

## 8.3 Applications

This work has shown that software sensors can be employed to detect meaningful semantics from raw sensor data. A meaningful textual narrative representing an event can be generated automatically. In addition, different access devices benefit from different representations of lifelog data. Some implications are generated from this work in the context of lifelogging development.

The lifelogging approach has many promising applications. Examples include transmitting professional knowledge (Bush, 1945), supporting the data owner's memory also called memory aid (Wood et al., 2012; Sellen et al., 2007), health monitoring (Doherty et al., 2013; Lane et al., 2011), mental health tool (Rennert and Karapanos, 2013; Son et al., 2013), social network analysis tool (Sueda et al., 2012), and as an even urban design tool (Ihara et al., 2011). Lifelogs can help objectively supply data and reveal potential errors inherent in self-reporting (Doherty et al., 2013). Many other potential yet undiscovered areas exist where lifelogging may be exploited by users in future generations.

Our work provides a good basis for the above applications. Researchers from other backgrounds can easily and efficiently use the lifelogging system developed in this study. Our system is based on the Android platform which is an open source. It means any person who is using the same platform can simply download the system and use it straight away.

We acknowledge that there are challenges to be overcome, such as privacy concerns, data storage, security of data and the development of a new generation of search and organisation tools, but we believe that these will be overcome and that we are on the cusp of a positive turning point for society. We anticipate the era of quantified individuals who know more about themselves than ever before, have more knowledge to improve the quality of their own life and can share life events and experiences in rich detail with friends and contacts.

## 8.4 Limitations

This study is concerned with improving the design of the lifelogging system in order to provide users with more approaches to access their lifelog data. Along with its contributions and implications, some limitations exist.

In this study, all of the experiments were conducted by a small number of participants, all of whom were working in the university. There are two main reasons for using a small sample size and a convenient sample strategy. Firstly, we had limited access to smartphones for this experimentation, due primarily to budgetary limitations. Secondly, owing to the nature of lifelogging, which requires the user to bring the lifelogging tool everywhere all the time, it was impossible to employ a large population to collect data. Lifelogging, as a subject has a history of small-scale experimentation. For example, some lifelogging research is only based on the data collected by one user (Hamm et al., 2013; Smith et al., 2011). Liao et al. (2007) employed four participants in their study. Six participants were employed in Harper et al. (2007)'s research and related their stories which were found by browsing SenseCam footage. Even for the MyLifeBits project, all data comes from Gordon Bell himself (Gemmell et al., 2002). In the experiments in this thesis, the participants employed were from candidate's university, but they all have different backgrounds. None of them had prior knowledge about lifelogging for the experiments being undertaken. However, it is important to point out that where possible, large-scale data was employed. For example, the important location-detecting data collection utilised three years of data; data that was gathered before this experiment was conceived, planned or conducted before the experiment was conducted.

As indicated in Section 6.2.2, this study did not consider all of the activity context change in segmenting events. For example, a user stands up, gets a cup of coffee and sits back down again within one minute. It would be treated as having no new event in this study. However, if there is one addition of one small detail (e.g. taking tablets from pocket before taking a cup of coffee), it could make this a very important event. The reason for this study not considering it is that if all the changes of activities are described in detail in narratives, a user may lose interest in reading such a large number of long narratives. It is a limitation of this study and is also a future research direction.

This study applied the support vector machine learning (SVM) technique to detect activity contexts and to segment events. Four reasons for choosing an SVM have been presented in Chapter 3. They are:

1) SVM has relatively low sensitivity to the number of training samples;
2) The same algorithm solves a variety of problems with little tuning;
3) SVM provides good out-of-sample generalisation;
4) The classification complexity in SVM does not depend on the feature space;
5) SVM is very easy-to-use.

On the easy-to-use function of SVM, a lot of applications and libraries in different programming languages are available online. We acknowledge that there are some limitations using SVM such as slow speed in training the dataset and the fact that it only supports a binary classification feature (all results are true or false) (Kotsiantis et al., 2007). However, these limitations are not serious concerns in this study due to the fact we do not have an extremely large volume of dataset and the dataset only needs training once. While there might be better machine learning techniques, which machine learning technique is best is not the focus of this study. In addition, according to Kotsiantis et al. (2007), there is no algorithm that can be the best in all cases. Comparison of system performance using the numerous machine learning techniques is a future research direction.

## 8.5 Future Work

Our work and results pose many new research questions. There are naturally a number of different directions of future research that could be undertaken.

- **About the lifelogging framework:** The lifelogging system in this study is a new generation of lifelogging tool. It has several advantages over previous work. Firstly, it does not need the user to upload data manually. Secondly, it does not have a storage limit; all data is uploaded to the server and automatically removed from the smartphone. Thirdly, a smartphone is a two-way communications medium; it can also give feedback to the user. Many apps will be available to give the user a summary of their data in the near future. Furthermore, it is easy to upgrade the software and the user does not need to buy additional hardware.

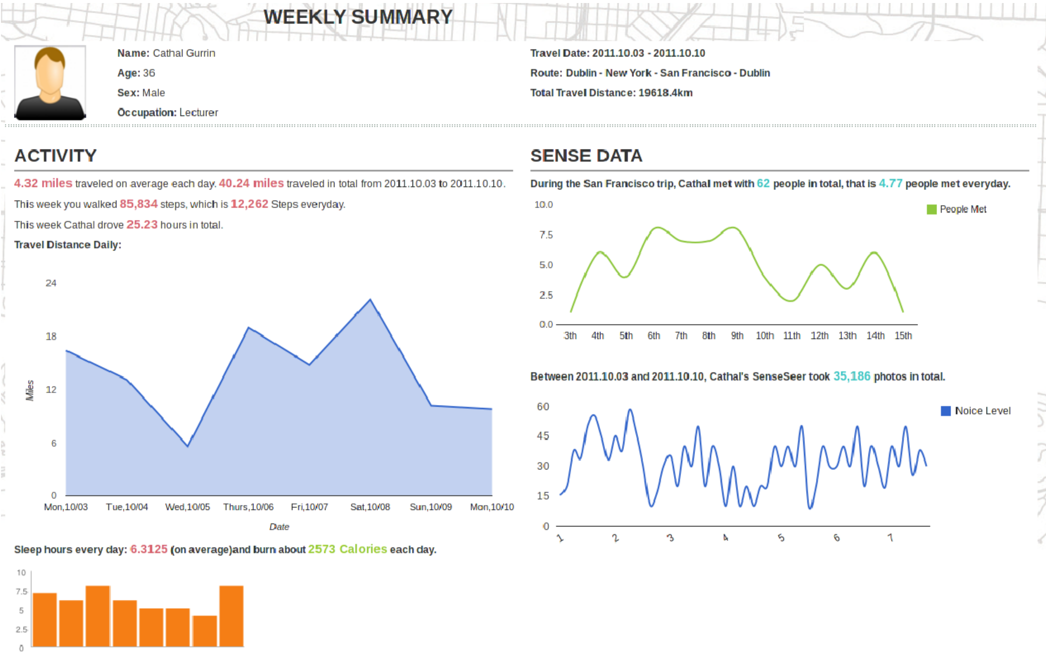
  However, it is still far from ideal. It does not have a fish-eye lens, so it only has limited field of view and it does not (as yet) implement accelerometer triggered photograph capture. For future work, we suggest that the system continues to collect data unless the user stops it manually. However, for various reasons, the user does not want to collect data in some situations. In future, we will investigate ways to control data collection based on users' context. For example, a user can decide to turn on or off the tool using location. If the user does not want to collect any information when he is at home, it can stop working automatically, and it can begin to work again, when the user leaves home.

  Because the software was initially developed as a lifelogging tool, it collects a wide range of data from all available sensors. There is no option for users to choose which data they want to collect or not. In future, we will make it more flexible. Users can customise what data they want to collect. Hence, that will also open more domains, such as home security and health care, with the inclusion of alerts and triggers. Based on a user's life pattern, it could also give the user some useful suggestion in advance. For example, it can request weather forecast from the web and the user's destination from user's location history. If it will rain in the user's destination, it will remind the user to bring an umbrella using text to speech technique before he/she goes outside. Hence real-time triggers and interventions are a good potential target for future work to increase the capability, or real-world impact, of lifelogs.

- **About the context:** In this thesis, we have explored how to detect six different contexts from different sources of sensor data. However, there are countless contexts in our lives. In future research, we will investigate additional contexts and also identify the most important contexts.
- **About the event segmentation:** A user may have many event boundary options if the event's boundary is not clear. A more flexible event segmentation algorithm would be necessary, one that does not impose a strict event boundary. It is known that events decrease in importance over time, so this should be taken into account. For our own future work, we will explore options for user-focused event segmentation approaches; we believe it will improve segmentation accuracy dynamically. In addition, we suggest the use of additional attributes of different contexts using other techniques or data mining algorithms for user to test and train new event segmentation models.

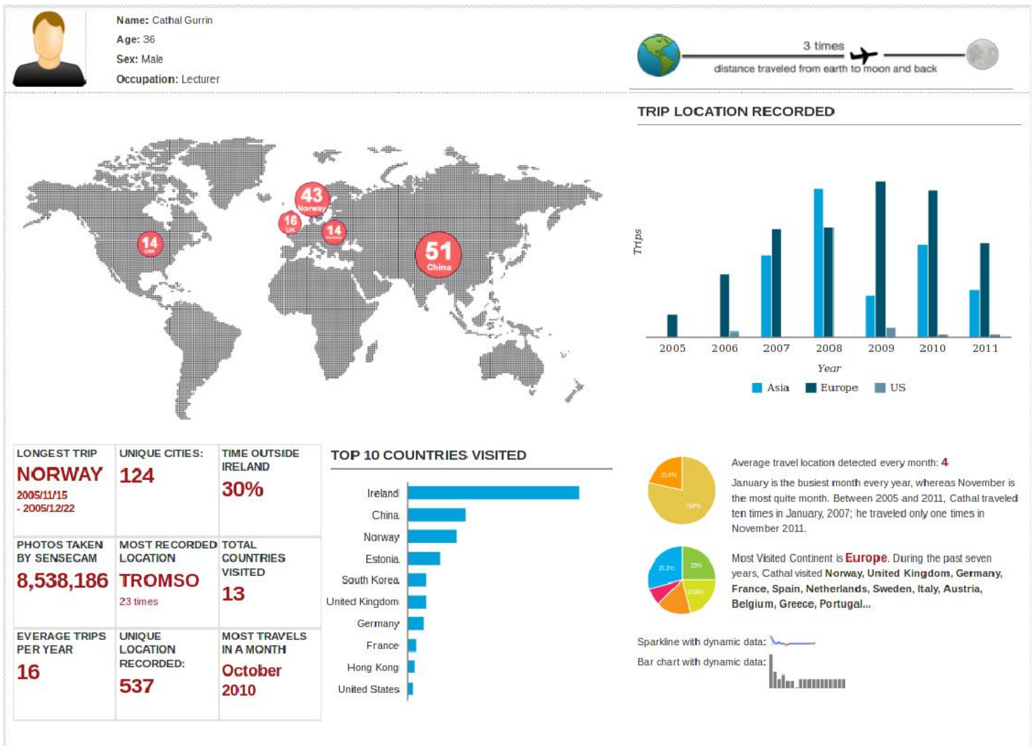
- **About the narrative generation:** In this thesis, we generated narratives based only on a limited number of contexts. However, there are a range of contexts which were not used in this work such as indoor, meeting, etc. from the photograph visual analysis of photographs. In future work, we will investigate narratives generation using more contexts from the virtual sensors. In addition, it is proposed to explore the summarisation of large amounts of event narratives to provide daily, weekly, etc. narrative summaries. This opens opportunities for the integration of novelty detection algorithms and the ability to place emphasis on narrative generation techniques.
- **About the personal information retrieval based on narratives:** In this study, lifelog data searching is currently based on indexed narratives. When a user searches for events using our system, the search engine would analyse his query and return all the related results. All the results are ranked by information retrieval algorithms using conventional text weighting approaches. These can be tailored for users and also can be context sensitive with the inclusion of novelty detection algorithms. The weight of different contexts for different users has been identified in previous work (Naaman et al., 2004). In the future, we suggest the development of new ranking algorithms for personal lifelog information retrieval.
- **About the lifelog representation:** In Chapter 7, we have shown some examples of lifelog data representation on different devices. However there are numerous ways to display lifelog data and this is just a starting point. More detailed and new approaches should be explored to generate personal info graphics. Examples of interfaces are presented in Figures 8.1 and 8.2.

**Figure 8.1: Web interface with lifelog data summary**




Source: The author (2013)

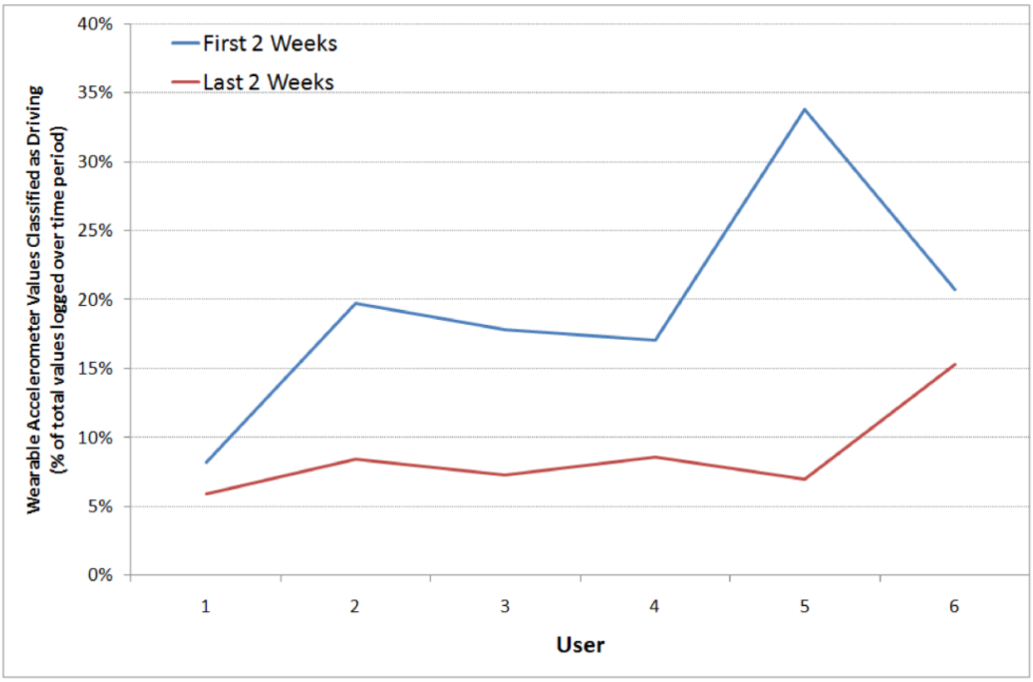
**Figure 8.2: Web interface with personal travel report**




Source: The author (2013)

- **About the personal information browsing and sharing:** The original idea of lifelogging was to collect all the information about one person. However, the success of social networking such as Facebook has shown people's enthusiasm for sharing their own information. This system collects all kinds of information. Users have more options to browse and share their data. For example, a user can choose to browse and share all photographs taken in some locations. He/she can also share some photographs which contain faces. Furthermore, users can combine all kinds of conditions to browse and share his/her information. For example, he/she can get all his/her photographs using environmental noise level, time, locations, and activities. In future, we will investigate and implement such approaches to browse and share lifelog data.

**Figure 8.3: Indication of driving activity detected over 6-week trial period**




Source: Doherty et.al (2010)

- **About the lifelogging affecting change:** One issue that we touched on in this work, but have not explored in depth, is that of affecting change in the user, as a result of lifelogs. Below (Figure 8.3) is an example of a user's driving behaviour changing during 6 weeks' experiment period (Doherty et al., 2010). In this small 'exploratory' experiment on developing a $CO_2$ estimator based on the accelerometer output, my role was to capture and manage the accelerometer output from the sensors. We found that the user reduced the driving frequency or length after he/she had involved in the lifelogging, thus indicating a possible change in user behaviour.

## 8.6 Final Thoughts

Lifelogging is, we believe, going to become a well-accepted technology in the coming years. We can see this trend in the Quantified Self movement whose members log (mostly manually) aspects of their lives. However manual logging is not possible for most people and not feasible in a longer term; manual logging needs to be replaced with automatic life logging. As O'Hara et al. (2008b) suggested "every piece of information may be valuable"; it is worthwhile capturing as much information as possible. It is worthwhile logging for future use if needed. It is within this framework that we have focused our research. The development and evaluation of end-to-end lifelogging solutions, that can be effective in real-time, for real-world data with minimal user input. There is still a long way to go. We have suggested the best representation for lifelog data on multiple devices, but much more work needs to be done in this area. At the moment, our lifelog data collection and organisation is operational but the search engine to actually locate content is not working on real-time data. It only supports very simple functionality. For example, it only indexes users' narrative documents once a day when users' data has been uploaded to the server. However, we believe this work provides a good starting point which provides valuable clues and guidance for future researchers in this area.

## REFERENCES

Ab Rahim, M.F. 2004. Steganografi: Penyembunyian Teks Dalam Video. PhD Thesis. Universiti Teknologi Malaysia.

Abe, M., Morinishi, Y., Maeda, A., Aoki, M., Inagaki, H., 2009. A life log collector integrated with a remote-controller for enabling user centric services. Consumer Electronics, IEEE Transactions on 55, 295–302.

Abowd, G.D., Atkeson, C.G., Hong, J., Long, S., Kooper, R., Pinkerton, M., 1997. Cyberguide: A mobile context-aware tour guide. Wireless networks 3, 421–433.

Agrawal, R., Faloutsos, C., Swami, A., 1993. Efficient similarity search in sequence databases. Foundations of Data Organization and Algorithms 69–84.

Aizawa, K., 2005. Digitizing personal experiences: Capture and retrieval of life log, in: Multimedia Modelling Conference, 2005. MMM 2005. Proceedings of the 11th International. pp. 10–15.

Aizawa, K., Hori, T., Kawasaki, S., Ishikawa, T., 2004a. Capture and efficient retrieval of life log, in: Proceedings of the Pervasive 2004 Workshop on Memory and Sharing of Experiences, Linz/Vienna, Austria, Tuesday.

Aizawa, K., Tancharoen, D., Kawasaki, S., Yamasaki, T., 2004b. Efficient retrieval of life log based on context and content, in: Proceedings of the the 1st ACM Workshop on Continuous Archival and Retrieval of Personal Experiences. pp. 22–31.

Albertos, P., Goodwin, G.C., 2002. Virtual sensors for control applications. Annual Reviews in Control 26, 101–112.

Allen, A.L., 2008. Dredging up the past: Lifelogging, memory, and surveillance. The

University of Chicago Law Review 47–74.

Appan, P., Sundaram, H., Birchfield, D., 2004. Communicating everyday experiences, in: International Multimedia Conference: Proceedings of the 1 St ACM Workshop on Story Representation, Mechanism and Context. pp. 17–24.

Ashbrook, D. & Starner, T., 2003. Using GPS to learn significant locations and predictmovement across multiple users, Personal and Ubiquitous Computing 7(5), 275–286.

Auria, L., Moro, R.A., 2008. Support vector machines (SVM) as a technique for solvency analysis. Deutsches Institut für Wirtschaftsforschung.

Azizyan, M., Constandache, I., Roy Choudhury, R., 2009. Surround Sense: mobile phone localization via ambience fingerprinting, in: Proceedings of the 15th Annual International Conference on Mobile Computing and Networking, MobiCom '09. ACM, New York, NY, USA, pp. 261–272.

Baeza-Yates, R., Ribeiro-Neto, B., others, 1999. Modern information retrieval. Addison-Wesley New York.

Bargh, M.S., Groote, R. de, 2008. Indoor localization based on response rate of bluetooth inquiries, in: Proceedings of the First ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments. pp. 49–54.

Bao, L. & Intille, S., 2004. Activity recognition from user-annotated acceleration data, Pervasive Computing 3001, 1–17. 10.1007/978-3-540-24646-6-1.

Belimpasakis, P., Roimela, K., You, Y., 2009. Experience explorer: a life-logging platform based on mobile context collection, in: Next Generation Mobile Applications, Services and Technologies, 2009. NGMAST'09. Third

International Conference On. pp. 77–82.

Bell, C.G., Gemmell, J., 2009. Total recall: How the e-memory revolution will change everything. Dutton New York.

Bell, G., Gemmell, J., 2007. A digital life. Scientific American Magazine 296, 58–65.

Berchtold, M., Budde, M., Gordon, D., Schmidtke, H. R. & Beigl, M., 2010. Actiserv: Activity recognition service for mobile phones, in Wearable Computers (ISWC), 2010 International Symposium on, IEEE, pp. 1–8.

Bradley, P.S., Mangasarian, O.L., 1998. Feature selection via concave minimisation and support vector machines, Machine Learning Proceedings of the Fifteenth International Conference (ICML'98). pp. 82–90.

Brank, J., Grobelnik, M., Milic-Frayling, N. & Mladenic, D., 2003. Training text classi-fiers with svm on very few positive examples, Microsoft Research .

Brooks, K.M., 1997. Do story agents use rocking chairs? The theory and implementation of one model for computational narrative, in: Proceedings of the Fourth ACM International Conference on Multimedia. pp. 317–328.

Brown, N.R., Schopflocher, D., 1998. Event clusters: An organization of personal events in autobiographical memory. Psychological Science 9, 470.

Bruner, J., 2004. Life as narrative. Social Research: An International Quarterly 71, 691–710.

Bush, V., 1945. As we may think. Atlantic Monthly 176, 101–108.

Byrne, D., Kelliher, A., Jones, G.J., 2011. Life editing: third-party perspectives on lifelog content, in: Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems. May 2011, Vancouver, Canada. pp. 1501–1510.

Byrne, D., Kelly, L., Jones, G.J., 2012. Multiple multimodal mobile devices: Lessons learned from engineering lifelog solutions, in: Handbook of Research on Mobile Software Engineering: Design, Implementation and Emergent Applications.

Carr, D., 1986. Time, narrative, and history. Cambridge Univ Press.

Chang, C.C., Lin, C.J., 2011. LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST) 2, 27.

Chen, G., Kotz, D., 2000. A survey of context-aware mobile computing research. Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College.

Chen, Y., 2009. Exploring memory cues to aid information retrieval from personal lifeLog archives, in: 23rd BCS Conference on Human-Computer Interaction, 1-5 September 2009, Cambridge, U.K.

Chen, Y., Jones, G.J., Ganguly, D., 2011. Segmenting and summarizing general events in a long-term lifelog, in: The 2nd Workshop Information Access for Personal Media Archives (IAPMA) at ECIR 2011, 18-21 April 2011, Dublin, Ireland.

Cheng, P.-W., Chennuru, S., Buthpitiya, S., Zhang, Y., 2010. A language-based approach to indexing heterogeneous multimedia lifelog, in: International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction. p. 26.

Chennuru, S., Chen, P.-W., Zhu, J., Zhang, J.Y., 2012. Mobile Lifelogger–Recording, Indexing, and Understanding a Mobile User's Life, in: Mobile Computing, Applications, and Services. Springer, pp. 263–281.

Cheong, Y.G., Young, R., 2008. Narrative generation for suspense: Modeling and evaluation. Interactive Storytelling 144–155.

Cheverst, K., Davies, N., Mitchell, K., Friday, A., Efstratiou, C., 2000. Developing a context-aware electronic tourist guide: some issues and experiences, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '00. ACM, New York, NY, USA, pp. 17–24.

Chittaro, Luca. 2006. Visualizing Information on Mobile Devices. Computer, 39(3): 40–45.

Conway, M.A., Loveday, C., 2011. SenseCam: The Future of Everyday Memory Research. Memory 19, 685–807.

Costantini, R., Susstrunk, S., 2004. Virtual sensor design, in: Electronic Imaging 2004. pp. 408–419.

Cusano, C., Ciocca, G., Schettini, R., 2004. Image annotation using SVM, in: Proceedings of Internet Imaging IV, Vol. SPIE. pp. 330–338.

De Jager, D., Wood, A.L., Merrett, G.V., Al-Hashimi, B.M., O'Hara, K., Shadbolt, N.R., Hall, W., 2011. A low-power, distributed, pervasive healthcare system for supporting memory, in: Proceedings of the First ACM MobiHoc Workshop on Pervasive Wireless Healthcare, MobileHealth '11. ACM, New York, NY, USA, pp. 5:1–5:7.

Dey, A.K., 2001. Understanding and Using Context. Personal Ubiquitous Comput. 5, 4–7.

Dey, A.K., Abowd, G.D., Salber, D., 2001. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Human–Computer Interaction 16, 97–166.

Doherty, Aiden R. 2008. Providing effective memory retrieval cues through automatic structuring and augmentation of a lifelog of images. PhD thesis, Dublin City University.

Doherty, A., Kelly, P., Foster, C., 2013. Wearable Cameras: Identifying Healthy Transportation Choices. Pervasive Computing, IEEE 12, 44–47.

Doherty, A.R., Byrne, D., Smeaton, A.F., Jones, G.J.F., Hughes, M., 2008. Investigating keyframe selection methods in the novel domain of passively captured visual lifelogs, in: Proceedings of the 2008 International Conference on Content-based Image and Video Retrieval. pp. 259–268.

Doherty, A.R., Caprani, N., Conaire, C.O., Kalnikaite, V., Gurrin, C., O'Connor, N.E., Smeaton, A.F., 2011. Passively recognising human activities through lifelogging. Computers in Human Behavior 27, 1948–1958.

Doherty, Aiden R., Chris JA Moulin, & Alan F. Smeaton. 2011. Automatically assisting human memory: A SenseCam browser. Memory 19(7): 785-795.

Doherty, A.R., Moulin, C.J.A., Smeaton, A.F., 2011. Automatically assisting human memory: A SenseCam browser. Memory 7, 785–795.

Doherty, A.R., Pauly-Takacs, K., Caprani, N., Gurrin, C., Moulin, C.J.A., O'Connor, N.E., Smeaton, A.F., 2012. Experiences of Aiding Autobiographical Memory Using the SenseCam. Human-Computer Interaction 27, 151–174.

Doherty, A.R., Qiu, Z., Foley, C., Lee, H., Gurrin, C., Smeaton, A.F., 2010. Green multimedia: informing people of their carbon footprint through two simple sensors, in: Proceedings of the International Conference on Multimedia. pp. 441–450.

Doherty, A.R., Smeaton, A.F., 2008a. Automatically segmenting lifelog data into events, in: Ninth International Workshop on Image Analysis for Multimedia Interactive Services. pp. 20–23.

Doherty, A.R., Smeaton, A.F., 2008b. Combining face detection and novelty to identify important events in a visual lifelog, in: IEEE 8th International Conference on Computer and Information Technology Workshops. pp. 348–353.

Doherty, A.R., Smeaton, A.F., 2010. Automatically augmenting lifelog events using pervasively generated content from millions of people. Sensors 10, 1423–1446.

Doswell, J.T., 2006. Context-aware mobile augmented reality architecture for lifelong learning, in: Advanced Learning Technologies, 2006. Sixth International Conference On. pp. 372–374.

Dourish, P., 2004. What we talk about when we talk about context. Personal Ubiquitous Comput. 8, 19–30.

Dowden, B., 2011. Time [WWW Document]. Internet Encyclopedia of Philosophy. URL http://www.iep.utm.edu/time/

Duan, K.-B., Rajapakse, J.C., Wang, H., Azuaje, F., 2005. Multiple SVM-RFE for gene selection in cancer classification with expression data. NanoBioscience, IEEE Transactions on 4, 228–234.

Fatah gen Schieck, A., Mottram, C. & Penn, A., 2003. Generating narrative spaces from events history.  6th International Conference Generative Art. : Milan, Italy.

Fitzgibbon, A., Reiter, E., 2003. Memories for life: Managing information over a human lifetime. UK Computing Research Committee Grand Challenge proposal 22, 13–16.

Fleury, A., Pedersen, J.S., Bo Larsen, L., 2012. Evaluating user preferences for video transfer methods from a mobile device to a TV screen. Pervasive and Mobile Computing.

Fu, N., Flood, P.C., Bosak, J., Morris, T. & O'Regan, P. 2013. Exploring the Performance Effect of High Performance Work System on Service Supply Chain in Professional Service Firms. Supply Chain Management: An International Journal. 18(3): 292-307

Gellersen, H.W., Schmidt, A., Beigl, M., 2002. Multi-sensor context-awareness in mobile devices and smart artifacts. Mob. Netw. Appl. 7, 341–351.

Gemmell, J., Aris, A., Lueder, R., 2005. Telling stories with MyLifeBits, in: Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on Multimedia and Expo. July 6–9 , Amsterdam , The Netherlands, pp. 1536–1539.

Gemmell, J., Bell, G., Lueder, R., Drucker, S., Wong, C., 2002. MyLifeBits: fulfilling the Memex vision, in: In Multimedia '02: Proceedings of the Tenth ACM International Conference on Multimedia. New York, NY : ACM Press, pp. 235–238.

Gong, Y., Chang, S., & Cheung, S. Y. 2010. High performance work system and collective OCB: a collective social exchange perspective. Human Resource Management Journal, 20(2): 119–137.

Grauman, Kristen & Darrell, Trevor. 2005. The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features, 1458–1465, in: Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on. IEEE.

Gu, Y., Lo, A., Niemegeers, I., 2009. A survey of indoor positioning systems for wireless personal networks. Communications Surveys & Tutorials, IEEE 11, 13–32.

Gurrin, C., Byrne, D., O'Connor, N., Jones, G.J., Smeaton, A.F., 2008. Architecture and challenges of maintaining a large-scale, context-aware human digital memory. Presented at the VIE 2008 - The 5th IET Visual Information Engineering 2008 Conference. 29 July - 1 August 2008, Xi An, China.

Gurrin, C., Qiu, Z., Hughes, M., Caprani, N., Doherty, A.R., Hodges, S.E., Smeaton, A.F., 2013. The SmartPhone as a platform for wearable cameras in health research. American journal of preventive medicine 44, 308–313.

Hamm, J., Stone, B., Belkin, M., Dennis, S., 2013. Automatic Annotation of Daily Activity from Smartphone-Based Multisensory Streams, in: Mobile Computing, Applications, and Services. Springer, pp. 328–342.

Hammerl, S., Hermann, T., Ritter, H., 2012. Towards a semi-automatic personal digital diary: detecting daily activities from smartphone sensors, in: Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments. p. 24.

Hansen, R., Wind, R., Jensen, C.S., Thomsen, B., 2009. Seamless indoor/outdoor positioning handover for location-based services in Streamspin, in: Mobile Data Management: Systems, Services and Middleware, 2009. MDM'09. Tenth International Conference On. pp. 267–272.

Hard, B.M., Tversky, B., Lang, D.S., 2006. Making sense of abstract events: Building event schemas. Memory & cognition 34, 1221–1235.

Hardy, B., 1968. Towards a Poetics of Fiction: 3) An Approach through Narrative, in: Novel: A Forum on Fiction. pp. 5–14.

Harper, R., Randall, D., Smyth, N., Evans, C., Heledd, L., Moore, R., 2007. Thanks for the Memory, in: HCI 2007 - Proceedings of the 21st BCS HCI Group Conference. Lancaster, U.K.

Hatcher, E., Gospodnetic, O., McCandless, M., 2004. Lucene in action. Manning Publications.

He, Z.-Y. & Jin, L.-W., 2008. Activity recognition from acceleration data using ar model representation and svm, in 'Machine Learning and Cybernetics, 2008 International Conference on', Vol. 4, IEEE, pp. 2245–2250.

Hess, J., Ley, B., Ogonowski, C., Wan, L., & Wulf, V, 2011. Jumping between devices and services: towards an integrated concept for social tv. In Proceddings of the 9th international interactive conference on Interactive television (pp. 11-20). ACM.

Hirakawa, M., 2007. Going beyond completeness in information retrieval. Databases in Networked Information Systems 70–80.

Hodges, S., Berry, E., Wood, K., 2011. SenseCam: A wearable camera that stimulates and rehabilitates autobiographical memory. Memory 7, 685–696.

Hodges, S., Williams, L., Berry, E., Izadi, S., Srinivasan, J., Butler, A., Smyth, G., Kapur, N., Wood, K., 2006. SenseCam: A retrospective memory aid. UbiComp 2006: Ubiquitous Computing 177–193.

Hoven, E. van den, Sas, C., Whittaker, S., 2012. Designing for Personal Memorie: Past, Present, and Future. Human-Computer Interaction 27, 1–12.

Hsu, C.W., Chang, C.C., Lin, C.J., others, 2003. A practical guide to support vector classification.

Huang, J., Xiao, Y., Liang, Y., 2009. A Novel Secure Access Method for Remote Databases Based on Mobile Agents, in: Natural Computation, 2009. ICNC'09. Fifth International Conference On. pp. 519–522.

Ihara, M., Kobayashi, M., Yoshioka, T., 2011. Human affordance as life-log for environmental simulations, in: Human Centered Design. Springer, pp. 235–242.

Jaimes, A., Omura, K., Nagamine, T., Hirata, K., 2004. Memory cues for meeting video retrieval, in: Proceedings of the the 1st ACM Workshop on Continuous Archival and Retrieval of Personal Experiences. pp. 74–85.

Jain, R., 2008. Eventweb: Developing a human-centered computing system. Computer 41, 42–50.

Jain, R., Sinha, P., 2010. Content without context is meaningless, in: Proceedings of the International Conference on Multimedia, MM  '10. ACM, New York, NY, USA, pp. 1259–1268.

Jamieson, Susan. 2004. Likert Scales: How to (ab) Use Them. Medical education, 38(12): 1217–1218.

Janardhan, S., Kumar, S.U., 2012. Fabrication of Metal and Non-Metal Separating Machine ( No. P5400). SRM University.

Jatob´ , L. C., Grossmann, U., Kunze, C., Ottenbacher, J. & Stork, W. (2008), Context-aware mobile health monitoring: Evaluation of different pattern recognition methods for classification of physical activity, in 'Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE', IEEE, pp. 5250–5253.

Joachims, T., 1999. Making large scale SVM learning practical. Universität Dortmund

Kabadayi, S., Pridgen, A., Julien, C., 2006. Virtual sensors: Abstracting data from physical sensors, in: Proceedings of the 2006 International Symposium on World of Wireless, Mobile and Multimedia Networks. pp. 587–592.

Kahneman, D., Krueger, A.B., Schkade, D.A., Schwarz, N., Stone, A.A., 2004. A

Survey Method for Characterizing Daily Life Experience: The Day Reconstruction Method. Science 306, 1776–1780.

Kaleja, M.M., Herb, A.J., Rasshofer, R.H., Friedsam, G., Biebl, E.M., 1999. Imaging RFID system at 24 GHz for object localization, in: Microwave Symposium Digest, 1999 IEEE MTT-S International. pp. 1497–1500.

Kang, J.H., Welbourne, W., Stewart, B., Borriello, G., 2004. Extracting places from traces of locations, in: Proceedings of the 2nd ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots. pp. 110–118.

Kao, T.-P., Lin, C.-W. & Wang, J.-S. (2009), Development of a portable activity detector for daily activity recognition, in 'Industrial Electronics, 2009. ISIE 2009. IEEE International Symposium on', IEEE, pp. 115–120.

Kasutani, Eiji & Yamada, Akio. 2001. The MPEG-7 Color Layout Descriptor: a Compact Image Feature Description for High-speed Image/video Segment Retrieval, 674–677, in: Image Processing, 2001. Proceedings. 2001 International Conference on. IEEE.

Kelly, P., Doherty, A. R., Berry, E., Hodges, S., Batterham, A. M. & charlie Foster (2011), 'Can we use digital life-log images to investigate active and sedentary travel behaviour? results from a pilot study', Intnational Journal Behavioural Nutrition Physical Activity 8, 44.

Keogh, Eamonn, Chakrabarti, Kaushik, Pazzani, Michael & Mehrotra, Sharad. 2001. Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases. Knowledge and information Systems, 3(3): 263–286.

Kern, N., Schiele, B., Schmidt, A., 2003. Multi-sensor activity context detection for wearable computing, in: Ambient Intelligence. Springer, pp. 220–232.

Kim, D. H., Hightower, J., Govindan, R. & Estrin, D. (2009), Discovering semantically meaningful places from pervasive rf-beacons, in 'Proceedings of the 11th international conference on Ubiquitous computing', ACM, pp. 21–30.

Kim, P. H. (2011), Web-based research collaboration service: Crowd lifelog research case study, in 'Next Generation Web Services Practices (NWeSP), 2011 7th International Conference on', IEEE, pp. 188–193.

Klepeis, N.E., Nelson, W.C., Ott, W.R., Robinson, J.P., Tsang, A.M., Switzer, P., Behar, J.V., Hern, S.C., Engelmann, W.H., others, 2001. The National Human Activity Pattern Survey (NHAPS): a resource for assessing exposure to environmental pollutants. Journal of exposure analysis and environmental epidemiology 11, 231–252.

Kotsiantis, S., Zaharakis, I., Pintelas, P., 2007. Supervised machine learning: A review of classification techniques. Frontiers in Artificial Intelligence and Applications 31, 249–268.

Kubat, M., Matwin, S., 1997. Addressing the curse of imbalanced training sets: one-sided selection, in: Machine Learning International Workshop Then Conference,. pp. 179–186.

Kurby, C.A., Zacks, J.M., 2008. Segmentation in the perception and memory of events. Trends in cognitive sciences 12, 72–79.

Lane, N.D., Mohammod, M., Lin, M., Yang, X., Lu, H., Ali, S., Doryab, A., Berke, E., Choudhury, T., Campbell, A., 2011. Bewell: A smartphone application to monitor, model and promote wellbeing, in: 5th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth2011).

Lara, Ó.D., Pérez, A.J., Labrador, M.A., Posada, J.D., 2011. Centinela: A human activity recognition system based on acceleration and vital sign data. Pervasive and Mobile Computing.

Lavelle, B., Byrne, D., Gurrin, C., Smeaton, A.F., Jones, G.J.F., 2007. Bluetooth familiarity: Methods of calculation, applications and limitations, in: MIRW 2007 - Mobile Interaction with the Real World, Workshop, 9 September 2007, Singapore.

Lee, M.-W., Khan, A. M. & Kim, T.-S. (2011), 'A single tri-axial accelerometer-based real-time personal life log system capable of human activity recognition and exercise information generation', Personal and Ubiquitous Computing 15(8), 887–898.192

Lee, S., Park, H., Hong, S., Lee, K. & Kim, Y. (2003), A study on the activity classification using a triaxial accelerometer, in 'Engineering in Medicine and Biology Society,2003. Proceedings of the 25th annual international conference of the IEEE', Vol. 3, IEEE,pp. 2941–2943.

Lee, Y., Cho, S.B., 2007. Extracting meaningful contexts from mobile life log. Intelligent Data Engineering and Automated Learning-IDEAL 2007 750–759.

Liao, L., 2006. Location-based activity recognition. University of Washington.

Liao, L., Fox, D., Kautz, H., 2007. Extracting places and activities from gps traces using hierarchical conditional random fields. The International Journal of Robotics Research 26, 119–134.

Likert, Rensis. 1932. A Technique for the Measurement of Attitudes. Archives of psychology, 40: 1–55.

Lin, W.-H., Hauptmann, A., 2006. Structuring Continuous Video Recordings of Everyday Life Using Time-Constrained Clustering, in: Multimedia Content

Analysis, Management, and Retieval - SPIE-IST Electronic Imaging. San Jose, California, USA, pp. 111–119.

Lindgaard, G., Dudek, C., Sen, D., Sumegi, L., Noonan, P., 2011. An exploration of relations between visual appeal, trustworthiness and perceived usability of homepages. ACM Transactions on Computer-Human Interaction (TOCHI) 18, 1.

Liu, F., Janssens, D., Wets, G., Cools, M., 2013. Annotating mobile phone location data with activity purposes using machine learning algorithms. Expert Systems with Applications 40, 3299–3311.

Liu, H., Darabi, H., Banerjee, P., Liu, J., 2007. Survey of wireless indoor positioning techniques and systems. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 37, 1067–1080.

Loui, A.C., Savakis, A.E., 2000. Automatic image event segmentation and quality screening for albuming applications, in: 2000 IEEE International Conference on Multimedia and Expo, 2000. ICME 2000. IEEE.

Lu, H., Yang, J., Liu, Z., Lane, N.D., Choudhury, T., Campbell, A.T., 2010. The Jigsaw continuous sensing engine for mobile phone applications, in: Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems. pp. 71–84.

Luhn, H.P., 1958. The automatic creation of literature abstracts. IBM Journal of research and development 2, 159–165.

Ma, L., Smith, D., Milner, B., 2003a. Environmental noise classification for context-aware applications, in: Database and Expert Systems Applications. pp. 360–370.

Ma, L., Smith, D.J., Milner, B.P., 2003b. Context awareness using environmental noise classification, in: Eighth European Conference on Speech Communication and Technology.

Majumder, P., Ray, P.P., 2012. Hatch-Sens: a Theoretical Bio-Inspired Model to Monitor the Hatching of Plankton Culture in the Vicinity of Wireless Sensor Network. International Journal of Computer Science and Information Technologies 3, 4764–4769.

Manjunath, B.S., Ohm, J.-R., Vasudevan, V.V., Yamada, A., 2001. Colour and texture descriptors. IEEE Transactions on Circuits and Systems for Video Technology 11, 703–715.

Mann, S., 1997. Wearable computing: a first step toward personal imaging. Computer 30, 25–32.

Mark, D., Nutting, J., LaMarche, J., 2011. Beginning iOS 5 Development: Exploring the iOS SDK. Apress.

Martincic, F., Schwiebert, L., 2005. Introduction to wireless sensor networking. Handbook of Sensor Networks: Algorithms and Architectures, I. Stojmenovic, Ed. John Wiley & Sons 1–41.

Mateas, M., Sengers, P., 1999. Narrative intelligence, in: Proceedings AAAI Fall Symposium on Narrative Intelligence. pp. 1–10.

Mathur, A., Majumder, A., Datta, S., Menon, S., Malhotra, S., Dahiya, A., 2012. LifeView: a lifelog visualization tool for supporting sentimental recall and sharing, in: Proceedings of the 24th Australian Computer-Human Interaction Conference. pp. 371–380.

Maurer, U., Smailagic, A., Siewiorek, D. P. & Deisher, M., 2006. Activity recognition and monitoring using multiple sensors on different body positions, in ‘Wearable and Implantable Body Sensor Networks, 2006. BSN 2006. International Workshop on’, IEEE, pp. 4–pp.

Mazhelis, O., Žliobaitė, I., Pechenizkiy, M., 2011. Context-aware personal route recognition, in: Discovery Science. Springer, pp. 221–235.

McVeigh, Kathryn Margaret. 2008. Mosaic Narrative a Poetics of Cinematic New Media Narrative. PhD Thesis. University of Southern Queensland.

Meijer, Gerwin AL, Westerterp, Klaas R., Verhoeven, Francois MH, Koper, Hans BM & ten Hoor, Foppe. 1991. Methods to Assess Physical Activity with Special Reference to Motion Sensors and Accelerometers. Biomedical Engineering, IEEE Transactions on, 38(3): 221–229.

Melgani, F., Bruzzone, L., 2004. Classification of hyperspectral remote sensing images with support vector machines. Geoscience and Remote Sensing, IEEE Transactions on 42, 1778–1790.

Mitchell, T., 1997. Machine Learning, McGraw Hill.

Mitzenmacher, Michael. 2004. A Brief History of Generative Models for Power Law and Lognormal Distributions. Internet mathematics, 1(2): 226–251.

Mizuno, H., Sasaki, K., Hosaka, H., 2007. Indoor-outdoor positioning and lifelog experiment with mobile phones, in: Proceedings of the 2007 Workshop on Multimodal Interfaces in Semantic Interaction. pp. 55–57.

Naaman, M., Harada, S., Wang, Q., Garcia-Molina, H., Paepcke, A., 2004. Context data in geo-referenced digital photo collections, in: MULTIMEDIA ’04: Proceedings of the 12th Annual ACM International Conference on Multimedia. ACM, New York, NY, USA, pp. 196–203.

Nakamura, Y., Itou, T., Tezuka, H., Ishihara, T., Abe, M., 2010. Personalized tv-program recommendations based on life log, in: Consumer Electronics (ICCE), 2010 Digest of Technical Papers International Conference On. pp. 143–144.

Newton, Isaac. 1802. Mathematical Principles of Natural Philosophy. London: A. Strahan.

Newtson, D.  1973 Attribution and the unit of perception of ongoing behavior. J. Pers. Soc. Psychol. 28, 28–38

Newtson, D., Engquist, G., 1976. The perceptual organization of ongoing behavior. Journal of Experimental Social Psychology 12, 436–450.

Newtson, D., Engquist, G.A., Bois, J., 1977. The objective basis of behavior units. Journal of Personality and social psychology 35, 847.

Nguyen, D.H., Marcu, G., Hayes, G.R., Truong, K.N., Scott, J., Langheinrich, M., Roduner, C., 2009. Encountering SenseCam: personal recording technologies in everyday life, in: Proceedings of the 11th International Conference on Ubiquitous Computing. ACM, pp. 165–174.

Nicolai, T., Behrens, N., Yoneki, E., 2006. Wireless rope: Experiment in social proximity sensing with bluetooth. IEEE Pervasive Computing and Communications (PerCom)-Interactive Experiment.

Niehaus, J., Young, R.M., 2009. A computational model of inferencing in narrative, in: AAAI Spring Symposium. pp. 2–3.

O'Connor, N., Cooke, E., Le Borgne, H., Blighe, M. & Adamek, T., 2005. The acetoolbox: Low-level audiovisual feature extraction for retrieval and classification, in 'Integration of Knowledge, Semantics and Digital Media Technology, 2005. EWIMT 2005. The 2nd European Workshop on the (Ref. No. 2005/11099)', pp. 55–60.

O'Hara, K., Tuffield, M., Shadbolt, N., 2008a. Lifelogging: Issues of Identity and Privacy with Memories for Life. Presented at the Identity and the Information Society, Arona, Italy.

O'Hara, K., Tuffield, M.M., Shadbolt, N., 2008b. Lifelogging: Privacy and empowerment with memories for life. Identity in the Information Society, 1(1), 155-172.

O'Hare, N., Gurrin, C., Jones, G.J.F., Smeaton, A.F., 2005. Combination of Content Analysis and Context Features for Digital Photo Retrieval, in: 2nd IEE European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies. IEEE Computer Society, Washington, DC, USA, pp. 323–328.

O'Hare, N., Gurrin, C., Lee, H., Murphy, N., Smeaton, A.F., Jones, G.J.F., 2005. My digital photos: where and when?, in: Proceedings of the 13th Annual ACM International Conference on Multimedia. pp. 261–262.

Obeid, N., Rao, R.B.K.N., 2010. On integrating event definition and event detection. Knowledge and information systems 22, 129–158.

Osada, I., Yoshino, T., 2012. Proposal and Evaluation of User's Actions Distribution Method Using Life Streaming Service on Lifelog System, in: Advanced Information Networking and Applications Workshops (WAINA), 2012 26th International Conference On. pp. 99–104.

Pahlavan, K., Li, X., Ylianttila, M., Chana, R., Latva-aho, M., 2000. An overview of wireless indoor geolocation techniques and systems. Mobile and Wireless Communications Networks 1–13.

Qiu, Z., Doherty, A.R., Gurrin, C., Smeaton, A.F., 2011. Mining user activity as a context source for search and retrieval, in: Semantic Technology and

Information Retrieval (STAIR), 2011 International Conference On. pp. 162–166.

Qiu, Z., Gurrin, C., Doherty, A.R., Smeaton, A.F., 2010. Term Weighting Approaches for Mining Significant Locations from Personal Location Logs, in: CIT. pp. 20–25.

Qiu, Z., Gurrin, C., Doherty, A.R., Smeaton, A.F., 2011. Automatically detecting important moments from everyday life using a mobile device, in: CHI 2011 Workshop on Video Interaction – Making Broadcasting a Successful Social Media, 2011-05-07, Vancouver, Canada.

Qiu, Z., Gurrin, C., Doherty, A.R., Smeaton, A.F., 2012. A real-time life experience logging tool. Advances in Multimedia Modeling 636–638.

Quackenbush, John. 2001. Computational Analysis of Microarray Data. Nature Reviews Genetics, 2(6): 418–427.

Quercia, D., Ellis, J., Capra, L., 2010. Nurturing Social Networks Using Mobile Phones. IEEE Pervasive Computing.

Rakotomamonjy, A., 2003. Variable selection using SVM based criteria. The Journal of Machine Learning Research 3, 1357–1370.

Ravi, N., Dandekar, N., Mysore, P., Littman, M.L., 2005. Activity recognition from accelerometer data, in: Proceedings of the National Conference on Artificial Intelligence. pp. 1541–1546.

Rawassizadeh, R., Tomitsch, M., Wac, K., Tjoa, A.M., 2012. UbiqLog: a generic mobile phone-based life-log framework. Personal and Ubiquitous Computing 17, 621–637.

Rekimoto, J., Miyaki, T., Ishizawa, T., 2007. LifeTag: WiFi-based continuous location logging for life pattern analysis. Lecture Notes in Computer Science 4718, 35.

Rennert, K., Karapanos, E., 2013. Faceit: supporting reflection upon social anxiety events with lifelogging, in: CHI'13 Extended Abstracts on Human Factors in Computing Systems. pp. 457–462.

Riedl, M.O., Young, R.M., 2010. Narrative planning: Balancing plot and character. Journal of Artificial Intelligence Research 39, 217–268.

Robertson, S., Wharton, C., Ashworth, C., Franzke, M., 1996. Dual device user interface design: PDAs and interactive television, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 79–86.

Robertson, S.E., Jones, K.S., Laboratory, U. of C.C., 1997. Simple, proven approaches to text retrieval. Computer Laboratory, University of Cambridge.

Schank, R.C., 2000. Tell me a story: Narrative and intelligence. Northwestern University Press.

Schieck, A.F. gen, Mottram, C., Penn, A., 2003. Generating Narrative Spaces from Events History. 6th International Conference Generative Art. : Milan, Italy.

Schilit, B., Adams, N., Want, R., 1994. Context-Aware Computing Applications, in: Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop On. pp. 85–90.

Schmidt, D., Seifert, J., Rukzio, E., Gellersen, H., 2012. A cross-device interaction style for mobiles and surfaces, in: Proceedings of the Designing Interactive Systems Conference. ACM, pp. 318–327.

Sellen, A.J., Fogg, A., Aitken, M., Hodges, S., Rother, C., Wood, K., 2007. Do life-logging technologies support memory for the past?: an experimental study using sensecam, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 81–90.

Sellen, A.J., Whittaker, S., 2010. Beyond total capture: a constructive critique of lifelogging. Communications of the ACM 53, 70–77.

Shi, W., Yang, J., Jiang, Y., Yang, F., Xiong, Y., 2011. Senguard: Passive user identification on smartphones using multiple sensors, in: Wireless and Mobile Computing, Networking and Communications (WiMob), 2011 IEEE 7th International Conference on, 10-12 Oct. 2011. IEEE, Wuhan, China, pp. 141–148.

Shneiderman, B., Plaisant, C., 2005. Designing the user interface: strategies for effective human-computer-interaction. Addison-Wesley.

Shneiderman, Ben, Plaisant, Catherine, Catherine, Maxine S. & Jacobs, Steven M. 2010. Designing the User Interface: Strategies for Effective Human-Computer Interaction. Reading, MA: Addison-Wesley Publ. Co.

Siewiorek, D., Smailagic, A., Furukawa, J., Krause, A., Moraveji, N., Reiger, K., Shaffer, J., Wong, F.L., 2003. Sensay: A context-aware mobile phone, in: Proceedings of the 7th IEEE International Symposium on Wearable Computers.

Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., Jain, R., 2000. Content-based image retrieval at the end of the early years. Pattern Analysis and Machine Intelligence, IEEE Transactions on 22, 1349–1380.

Smith, A., O'Hara, K., Lewis, P., 2011. Visualising the past: Annotating a life with linked open data, in: Web Science Conference 2011, Koblenz , Germany.

Son, Y.-S., Pulkkinen, T., Park, J.-H., 2013. Active monitoring for lifestyle disease patient using data mining of home sensors, in: Consumer Electronics (ICCE), 2013 IEEE International Conference On. pp. 276–277.

Sueda, K., Miyaki, T., Rekimoto, J., 2012. Social Geoscape: Visualizing an Image of the City for Mobile UI Using User Generated Geo-Tagged Objects, in: Mobile and Ubiquitous Systems: Computing, Networking, and Services. Springer, pp. 1–12.

Takata, Katsuhiro, Ma, Jianhua, Apduhan, Bernady O., Huang, Runhe & Jin, Qun. 2008. Modeling and Analyzing Individual's Daily Activities Using Lifelog, 503–510, in: Embedded Software and Systems, 2008. ICESS'08. International Conference on. IEEE.

Tambe, M., 2012. TV Human Interface: Different Paradigm from that of PC and Mobile. IEEE Code of Ethics.

Tancharoen, D., Yamasaki, T., Aizawa, K., 2006. Practical Life Log Video Indexing Based on Content and Context, in: Multimedia Content Analysis, Management, and Retieval In Proceedings of SPIE-IST Electronic Imaging. San Jose, California, USA.

Tao, D., Jin, L., Liu, W., Li, X., 2013. Hessian Regularized Support Vector Machines for Mobile Image Annotation on the Cloud. Multimedia, IEEE Transactions on.

Tapia, E. M., Intille, S. S., Haskell, W., Larson, K., Wright, J., King, A. & Friedman, R., 2007. Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart monitor, in In: Proc. Int. Symp. on Wearable Comp, Citeseer.

Trevisani, E., Vitaletti, A., 2004. Cell-ID location technique, limits and benefits: an experimental study, in: Mobile Computing Systems and Applications, 2004.

WMCSA 2004. Sixth IEEE Workshop on, 2-3 Dec. 2004. Windermere, Cumbria, UK, pp. 51–60.

Tuffield, M.M., Millard, D.E., Shadbolt, N.R., 2005. Narrative as a Form of Knowledge Transfer, Narrative Theory and Semantic: Present Challenges-Future Possibilities (9 month Report).

van den Hoven, E., Sas, C. & Whittaker, S., 2012. Designing for personal memorie: Past,present, and future, Human-Computer Interaction 27(1-2), 1–12.

Vapnik, V., 1999. The nature of statistical learning theory. springer.

Vemuri, S., Schmandt, C., Bender, W., 2006. iRemember: a personal, long-term memory prosthesis, in: Proceedings of the 3rd ACM Workshop on Continuous Archival and Retrival of Personal Experences. pp. 65–74.

Wang, Guanhua. 2011. Improving Data Transmission in Web Applications via the Translation Between XML and JSON, 182–185, in: Communications and Mobile Computing (CMC), 2011 Third International Conference on. IEEE.

Wang, L., Wang, C., Xie, X., Forman, J., Lu, Y., Ma, W.-Y., Li, Y., 2005. Detecting dominant locations from search queries, in: SIGIR ’05: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, New York, NY, USA, pp. 424–431.

Wang, L., Zhang, Y., Feng, J., 2005. On the Euclidean distance of images. Pattern Analysis and Machine Intelligence, IEEE Transactions on 27, 1334–1339.

Wang, Z., Hoffman, M.D., Cook, P.R., Li, K., 2006. VFerret: content-based similarity search tool for continuous archived video, in: Proceedings of the 3rd ACM Workshop on Continuous Archival and Retrival of Personal Experences. pp. 19–26.

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., Vapnik, V., 2001. Feature selection for SVMs. Advances in neural information processing systems 668–674.

Weston, J., Watkins, C., 1999. Support vector machines for multi-class pattern recognition, in: Proceedings of the Seventh European Symposium on Artificial Neural Networks. pp. 219–224.

Wikipedia, Barnes & Noble Nook, http://en.wikipedia.org/wiki/Barnes_%26_Noble_Nook (as of Sept. 16, 2013, 09:30 GMT).

Wikipedia, Bluetooth File Exchange, http://en.wikipedia.org/wiki/Bluetooth_File_Exchange (as of Sept. 16, 2013, 09:25 GMT).

Wolf, J., Guensler, R., Bachman, W., 2001. Elimination of the travel diary: Experiment to derive trip purpose from global positioning system travel data. Transportation Research Record: Journal of the Transportation Research Board 1768, 125–134.

Won, Chee Sun, Park, Dong Kwon & Park, Soo-Jun. 2002. Efficient Use of MPEG-7 Edge Histogram Descriptor. Etri Journal, 24(1): 23–30.

Wood, A.L., Merrett, G.V., de Jager, D., Al-Hashimi, B.M., O'Hara, K., Shadbolt, N.R., Hall, W., 2012. DejaView: Help with memory, when you need it. Presented at the SenseCam 2012: Third Annual Symposium, Oxford, UK.

Xu, J., Kemeny, S., Park, G., Frattali, C., Braun, A., 2005. Language in context: emergent features of word, sentence, and narrative comprehension. Neuroimage 25, 1002–1015.

Yan, Z., Subbaraju, V., Chakraborty, D., Misra, A. & Aberer, K., 2012, Energy-efficient continuous activity recognition on mobile phones: An activity-adaptive approach, in : Wearable Computers (ISWC), 2012 16th International Symposium on, IEEE, pp. 17–24.

Ye, Y., Zheng, Y., Chen, Y., Feng, J., Xie, X., 2009. Mining individual life pattern based on location history, in: Mobile Data Management: Systems, Services and Middleware, 2009. MDM'09. Tenth International Conference On. pp. 1–10.

Zacks, J. M., & Swallow, K. M. 2007. Event segmentation. Current Directions in Psychological Science, 16(2), 80-84.

Zacks, J.M., Tversky, B., 2001. Event structure in perception and conception. Psychological bulletin 127, 3–21.

Zhang, Rui & Wang, Wenjian. 2011. Facilitating the Applications of Support Vector Machine by Using a New Kernel. Expert Systems with Applications, 38(11): 14225–14230.

Zhao, M., Ge, F., Zhang, T. & Yuan, Z., 2011. Antimaldroid: An efficient svm-based malware detection framework for android, in: Information Computing and Applications, Springer, pp. 158–166.

Zhu, C. & Sheng, W., 2009. Human daily activity recognition in robot-assisted living using multi-sensor fusion, in 'Robotics and Automation, 2009. ICRA'09. IEEE International Conference on', IEEE, pp. 2154–2159.

# APPENDIX 1 AN EXAMPLE OF ONE DAY'S LIFELOG DATA ANALYSIS

## 1. Data Collection

The data is collected by one user from 9:12am to 6:13pm on 9th April 2012 (the home part is removed). Raw data profile is presented in Table A1.1.

**Table A1.1 Raw sensor data profile**

| Item | number | Item | number |
|---|---|---|---|
| Image | 1,076 | Noise level | 1078 |
| Accelerometer reading | 161,369 | SMS | 3 |
| Bluetooth | 834 | Screen status | 8 |
| WiFi | 800 | Base station | 14 |

## 2. Context Detection

a) Time is translated to "Day, Month, Year" format. It will be translated to "yesterday", if the user accesses the narrative today.

b) Location is clustered and translated to semantic place, such as "home".

c) Activity of each unit is identified.

d) Bluetooth detecting. The relationship of the user and Bluetooth owner is checked (from the smartphone)

e) Face detection. The numbers of faces in each photograph are checked. In the new version we do face detection on two sides, phone and server, because sometime face detection accuracy is not high. Only when two sides have consistent results, we label the photograph with face.

f) Noisy level is translated to "Noisy", "Quiet", or "Normal" based on its value.

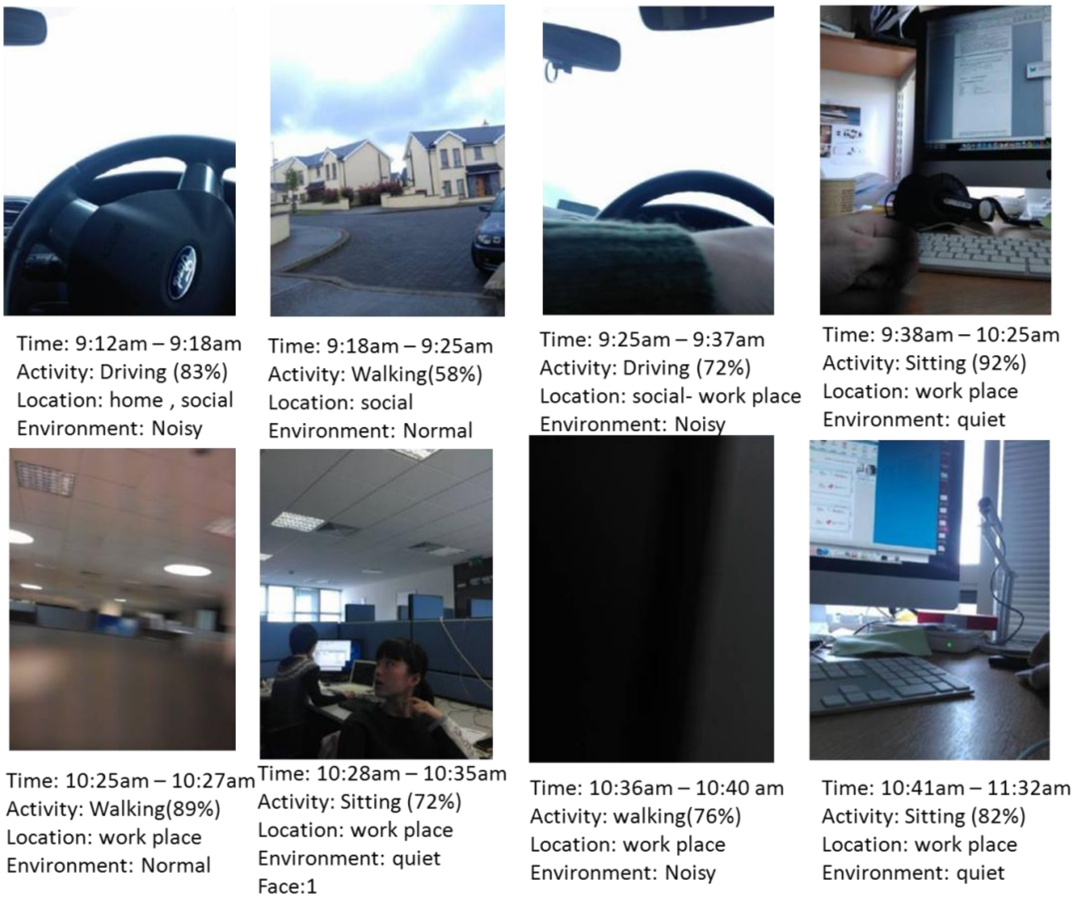
## 3. Events Segmentation

In the event segmentation process, one day's data is segmented into 18 events using our event segmentation model (Trained by SVM).

## 4. Keyframe Selection

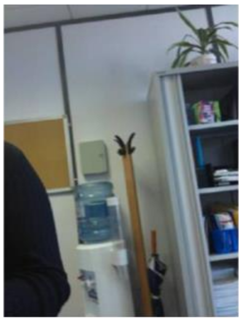
The steps to choose a keyframe include: 1) ordering photographs which contains face in the event by their quality (sharpness); 2) ordering photographs in the event by their quality (sharpness) if there is no photograph which contain face; and 3) selecting the first one as keyframe. The results of keyframe are shown in Figures A1.1 and A1.2.

**Figure A1.1 Keyframe selection results interface**

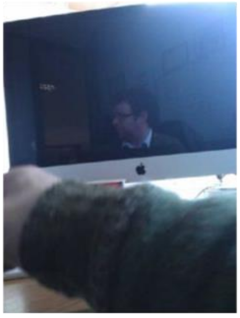



Source: The author (2013)

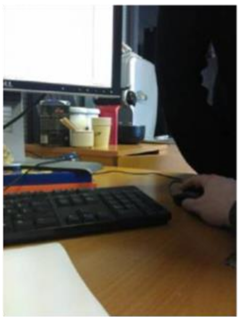
**Figure A1.2 Keyframe selection results interface (continued)**



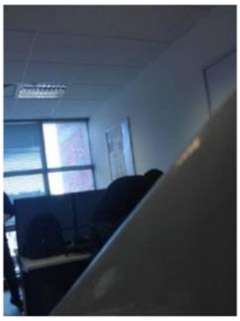
Time: 11:32am – 11:43am
Activity: Walking(83%)
Location: work place
Environment: Noisy



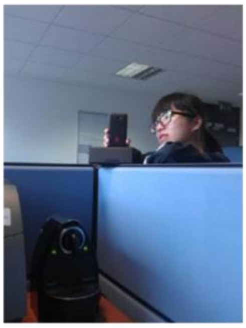
Time: 11:43am – 12:16pm
Activity: Sitting (52%)
Location: work place
Environment: quiet



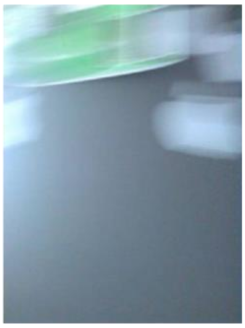
Time: 12:17pm – 2:36pm
Activity: Sitting (83%)
Location: work place
Environment: quiet



Time: 2:37pm – 2:43pm
Activity: Walking(72%)
Location: work place
Environment: Normal



Time: 2:44pm – 3:18pm
Activity: Sitting (62%)
Location: work place
Environment: quiet
Face:1



Time: 3:19pm – 3:35pm
Activity: Walking (82%)
Location: work place
Environment: Noisy

Time: 3:36pm – 5:15pm
Activity: Sitting (92%)
Location: work place
Environment: Normal

Time: 5:16pm – 5:19pm
Activity: Walking(87%)
Location: work place
Environment: Noisy

Time: 5:20pm – 5:38pm
Activity: Sitting (78%)
Location: work place
Environment: Normal

Time: 5:38pm – 6:13pm
Activity: driving(65%)
Location: work place-home
Environment: Noisy

Source: The author (2013)

## 5. Narratives Generation

The sentences are generated using contexts. When the user walks, only the starting point and destination are included. Only dominant activity is included. The fabulas and sjuzets are shown in Table A1.2. As shown in Figures A1.3 and A1.4, narratives were generated without manual inputs.

**Table A1.2 Fabula and sjuzet outputs**

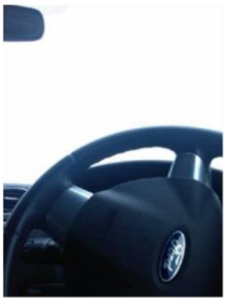
| Fabula | Sjuzet |
|---|---|
| It was at 9:12am on 9th April 2012.<br>You were driving from home to a social place.<br>It was noisy. | It was at 9:12am on 9th April 2012, You were driving from home to social place. It was noisy. |
| It was at 9:18am on 9th April 2012.<br>You were walking in a social place.<br>It was normal. | It was at 9:18am. You were walking in a social place. |
| It was at 9:25am on 9th April 2012.<br>You were driving from social place to the work place.<br>It was noisy. | It was at 9:25am. You were driving from social place to the work place. It was noisy. |
| It was at 9:38am on 9th April 2012.<br>You were sitting in the work place.<br>It was quiet. | It was at 9:38am. You were sitting in the work place. It was quiet. |
| It was at 10:25am on 9th April 2012.<br>You were walking in the work place.<br>It was normal. | It was at 10:25am. You were walking. |
| It was at 10:28am on 9th April 2012.<br>You were sitting in the work place.<br>You were with your colleague.<br>It was quiet. | It was at 10:28am. You were sitting with your colleague. It was quiet. |
| It was at 10:36am on 9th April 2012.<br>You were walking in the work place.<br>It was noisy. | It was at 10:36am. You were walking. It was noisy. |
| It was at 10:41am on 9th April 2012.<br>You were sitting in the work place.<br>It was quiet. | It was at 10:41am. You were sitting. It was quiet. |
| It was at 11:32am on 9th April 2012.<br>You were walking in the work place.<br>It was noisy. | It was at 11:32am. You were walking. It was noisy. |
| It was at 11:43am on 9th April 2012.<br>You were sitting in the work place.<br>It was quiet. | It was at 11:43am. You were sitting. It was quiet. |

**Table A1.3 Fabula and sjuzet (continued)**

| Fabula | Sjuzet |
|---|---|
| It was at 12:17pm on 9th April 2012.<br>You were sitting in the work place.<br>It was quiet. | It was at 12:17pm. It was quiet. |
| It was at 2:37pm on 9th April 2012.<br>You were walking in the work place.<br>It was normal. | It was at 2:37pm. You were walking. |
| It was at 2:44pm on 9th April 2012.<br>You were sitting in the work place.<br>You were with your colleague.<br>It was quiet. | It was at 2:44pm. You were sitting with your colleague. It was quiet. |
| It was at 3:19pm on 9th April 2012.<br>You were walking in the work place.<br>It was noisy. | It was at 3:19pm. You were walking. It was noisy. |
| It was at 3:36pm on 9th April 2012.<br>You were sitting in the work place.<br>It was normal. | It was at 3:36pm. You were sitting. |
| It was at 5:16pm on 9th April 2012.<br>You were walking in the work place.<br>It was noisy. | It was at 5:16pm. You were walking. It was noisy. |
| It was at 5:20pm on 9th April 2012.<br>You were sitting in the work place.<br>It was normal. | It was at 5:20pm. You were sitting. |
| It was at 5:38pm on 9th April 2012.<br>You were driving from work place to home.<br>It was noisy. | It was at 5:38pm. You were driving from work place to home. It was noisy. |

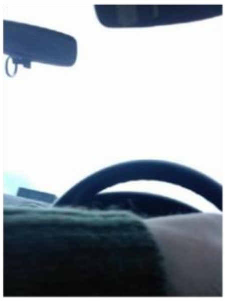
**Information augmentation:** We provide a user interface for user inputting, if they have more detailed information about their activities and the people name they met. If they don't input these data, the basic activities will be kept. The people will be labelled as friends if a face is detected when user is in social place and colleague if in the work place. Information augmentation interface is presented in Figure A1.5. Figures A1.6 and A1.7 show the narratives generated with manual inputs.

**Figure A1.3 Narratives without manual inputs**

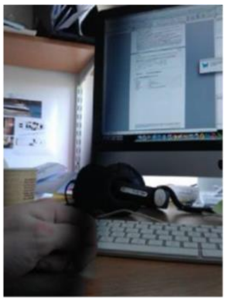


It was at 9:12am in 9th April 2012, You were driving from home to a social place. It was noisy.

It was at 9:18am. You were walking in a social place.

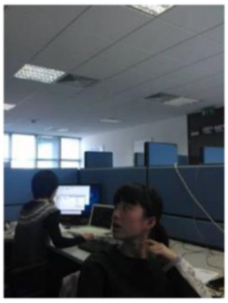


It was at 9:25am. You were driving from a social place to work place. It was noisy.



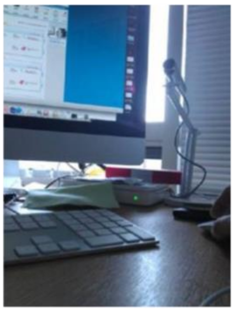
It was at 9:38am. You were sitting in the work place. It was quiet.

It was at 10:25am. You were walking.

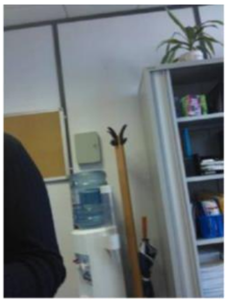


It was at 10:28am. You were sitting with your colleague. It was quiet.

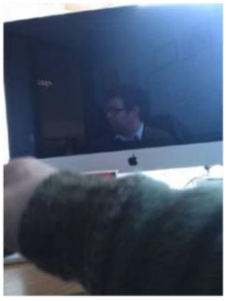
It was at 10:36am. You were walking. It was noisy.



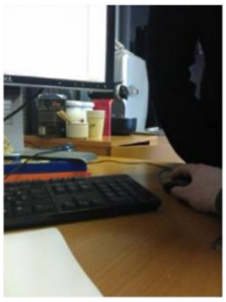
It was at 10:41am. You were sitting. It was quiet.



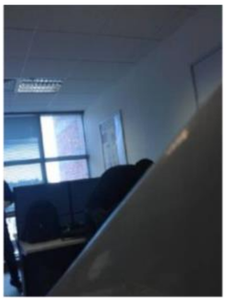
It was at 11:32am. You were walking. It was noisy.



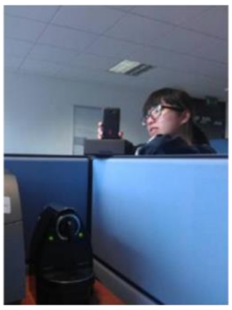
It was at 11:43am. You were sitting. It was quiet.



It was at 12:17pm. it was quiet.



It was at 2:37pm. You were walking.



It was at 2:44pm. You were sitting with your colleague. It was quiet.

It was at 3:19pm. You were walking. It was noisy.

It was at 3:36pm.You were sitting.

It was at 5:16pm. You were walking. It was noisy.

Source: The author (2013)

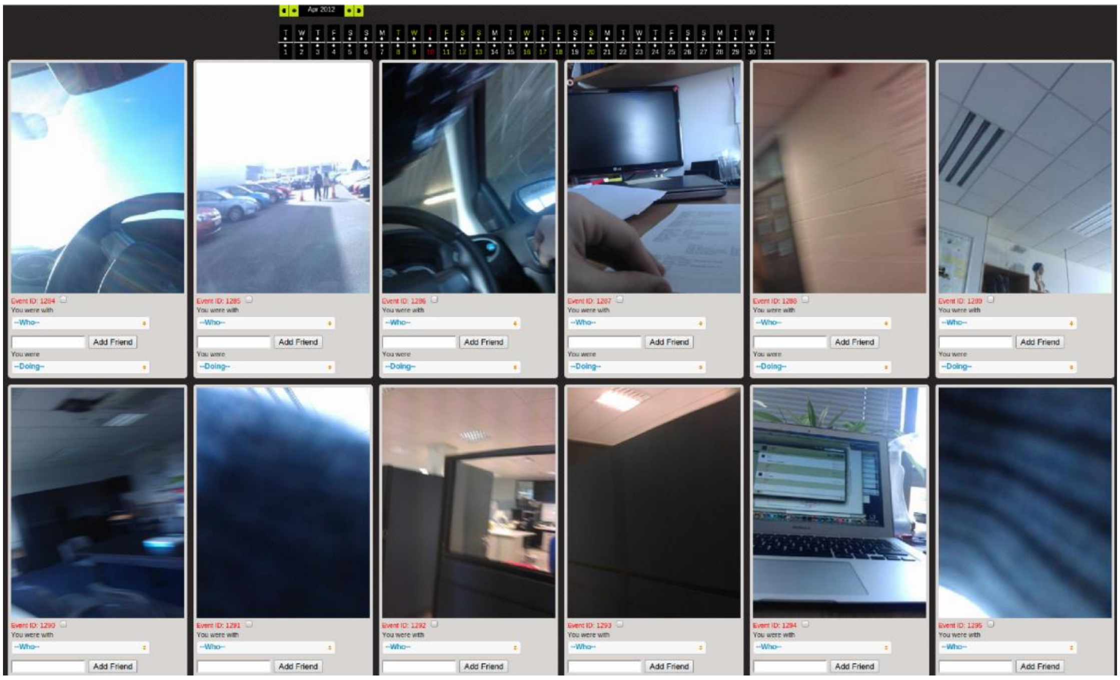
## Figure A1.4 Narratives without manual inputs (continued)

It was at 5:20pm. You were sitting.

It was at 5:38pm. You were driving from work place to home. It was noisy.

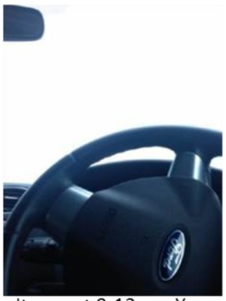
Source: The author (2013)

## Figure A1.5 Information augmentation interface

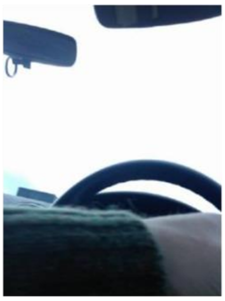


Source: The author (2013)

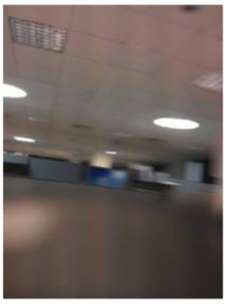
**Figure A1.6 Narratives with manual inputs**

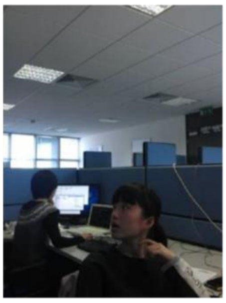

It was at 9:12am. You were driving from home to a social place. It was noisy.


It was at 9:18am. You were shopping in a social place.

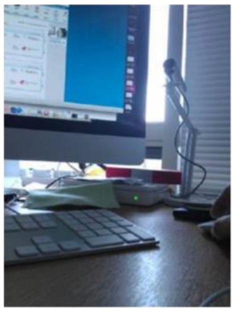

It was at 9:25am. You were driving from a social place to walk place. It was noisy.

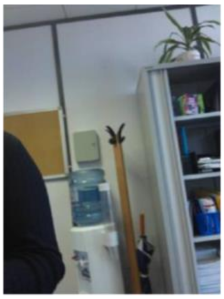

It was at 9:38am. You were working in the work place. It was quiet.

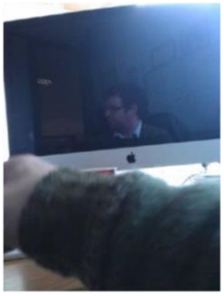

It was at 10:25am. You were walking.

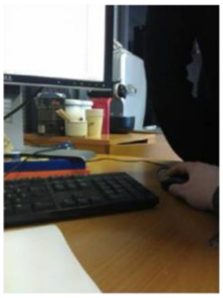

It was at 10:28am. You were working with your Z. It was quiet.

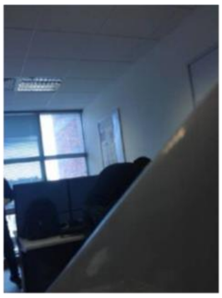

It was at 10:36am. You were walking. It was noisy.

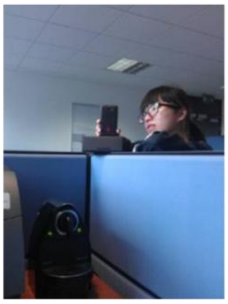

It was at 10:41am. You were working. It was quiet.

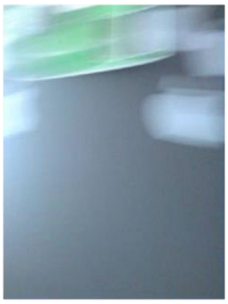

It was at 11:32am. You were walking. It was noisy.


It was at 11:43am. You were eating. It was quiet.


It was at 12:17pm. You were working.


It was at 2:37pm. You were walking.


It was at 2:44pm. You were sitting with Z. It was quiet.

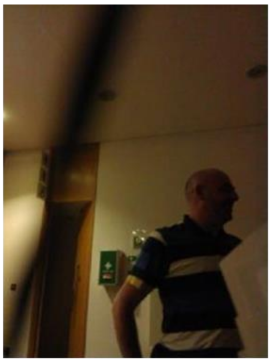

It was at 3:19pm. You were walking. It was noisy.


It was at 3:36pm. You were working with A.


It was at 5:16pm. You were walking. It was noisy.

Source: The author (2013)

**Figure A1.7 Narratives with manual inputs (continued)**



It was at 5:20pm. You were socializing with J.

It was at 5:38pm. You were driving from work place to home. It was noisy.