Human Activity Recognition from Multimodal Data

by Carlos Agell Gimeno
Human Activity Recognition from Multimodal Data

Carlos Agell Gimeno

Submitted to Telecom BCN, Polytechnical University of Barcelona (UPC), on June 14, 2007, in partial fulfillment of requirements for the degree of Telecommunications Engineer

Abstract

Context-aware pervasive systems refer to systems that can be aware of their physical (and virtual) environment or situation, and respond intelligently based on such awareness. Accurate and comprehensive recognition of a user’s context is an important step towards that goal. In this thesis a set of wearable sensors including sensors for location, speech, video, software usage and body sensors are used to recognize daily activities. After a feature extraction step a new semi-supervised clustering algorithm based on error function minimization is introduced to cluster recorded information into 9 different activities. The grouping obtained is bettered using a statistical post-clustering step improving overall recognition accuracy. The context information won is used to record user’s daily activities for further processing or recording purposes.

Key words:

Thesis Advisor: Fernando de la Torre
Thesis Director: Albert Oliveras

Barcelona June 14, 2007
Big Brother is watching you. [George Orwell]

A record if it is to be useful to science, must be continuously extended, it must be stored, and above all it must be consulted. [Vannevar Bush]

Education is the best provision for old age. [Aristotle]
Context-aware pervasive systems refer to systems that can be aware of their physical (and virtual) environment or situation, and respond intelligently based on such awareness. Accurate and comprehensive recognition of a user’s context is an important step towards that goal. In this Project a set of wearable sensors including sensors for location, speech, video, software usage and body sensors are used to recognize daily activities. After a feature extraction step a new semi-supervised clustering algorithm based on error function minimization is introduced to cluster recorded information into 9 different activities. The grouping obtained is bettered using a statistical post-clustering step improving overall recognition accuracy. The context information won is used to record user’s daily activities for further processing or recording purposes.
Resum†

Els sistemes Conscients del contexte són aquell sistemes que poden adonar-se de la seva situació tant física com virtual, i responen intellegantment basats en aquesta consciència. El reconeixement comprensiu i acurat del context de l’usuari és un pas important cap a aquest objectiu. En aquest Projecte Final de Carrera s’empren un conjunt de sensors lleugers i de fàcil ús, entre els que es troben sensors de localització, parla, video, utilització del software en el computador i sensors de variables corporals per a reconèixer les activitats diàries. Posteriorment a un pas d’extracció de les característiques (features) de les dades, s’introduceix un nou algoritme de clusterització (clustering) semi-supervisat basat en la minimització d’una funció d’error. L’esmentat algorisme permet agrupar la informació en 9 activitats diferents. Els grups obtinguts s’alteren lleugerament per mitjà d’un pas de post-processat estadístic per tal de millorar el comportament global del sistema. La informació obtinguda sobre el context en el que es troba l’usuari s’empra per a enregistrar les activitats diàries o per a posteriors passos de processat o fins estadístics.

†[Following the recommendations from Telecom Barcelona (ETSETB) this section and the Introduction of this Thesis will be written both in Catalan and English] [Seguint les recomenacions de Telecom Barcelona (ETSETB) tant aquesta secció (Resum) com la Introducció del present Projecte Final de Carrera seran escrites en Anglès i en Català]
El reconeixement de les activitats que porten a terme les persones durant un dia, lluny d’una entelèquia futurista, és una aplicació d’alt nivell de la tecnologia que avui en dia tenim al nostre abast. Tot i que a primera vista sembli un problema plantejat de manera innovadora, una petita cerca dins els principals motors de busca a nivell universitari i professional ens portarà a veure que és una problemàtica àmpliament estudiada dins la comunitat científica recent. De fet algunes universitats ja compten amb departaments dedicats a la interacció entre homes i màquines (Human Computer Interaction) on seccions senceres estudien el que anomenen Pervasive Computing.

Sembla innecessari destacar la importància que, a dia d’avui i de cara a futur, té la interacció de persones i màquines. En un món on cada vegada tenim més interacció amb aparells governats per l’electrònica (des del telèfon mòbil, els cotxes, els electrodomèstics passant pels ascensors, ordinadors i agendes electròniques) resulta interessant intentar integrar la vida diària al món electrònic, ja que és evident que el pas contrari està més que donat.

L’objectiu d’aquest Projecte Final de Carrera és arribar a classificar les activitats que porta a terme una persona durant un dia. Aquesta tasca és sens dubte un primer pas per a crear la següent generació de productes electrònics. Les aplicacions que aquesta tecnologia pot reportar s’expandeixen en multitud de camps, des del camp mèdic, sociològic, estadístic, a sectors com el de l’economia de mercat a gran escala basat en comportaments socials, passant per sistemes de control i seguiment de la gent gran.

Imaginin telèfons mòbils que no passen trucades quan estem reunits o en un moment de
màxima productivitat, frigorífics que preparen el nostre refresc preferit perquè estigui a la millor temperatura basant-se en l’activitat que hem portat a terme en aquell dia, cotxes que no ens deixen conduir si hem begut o hospitals que ens recomanen alterar el nostre horari perquè estadísticament la gent que segueix un determinat horari té menys riscos d’atac de cor. Aquest projecte proposa una serie de prestacions que ens apropen aquests serveis de somni.

En aquest document la problemàtica s’enfoca des del punt de vista del postprocessat de les dades * per tal de realitzar l’anàlisi detallat i emprant un llenguatge de programació d’alt nivell. Treballs futurs exigiran implementació a baix nivell que permetrà el funcionament del sistema en temps real a més d’un disseny hardware i de comunicacions entre els dispositius emprats.

Aquest Projecte Final de Carrera proposa emprar quatre sensors de baix i mig nivell per a l’anàlisi de les activitats d’una persona dins d’un entorn d’oficina. En concret per a informació referent a mesures fisiològiques s’empra un producte de BodyMedia®Corp capaç de mesurar variables vitals del cos humà (mesures representatives de quantitat de suor, energia, flux de calor i acceleració del bracc en dues dimensions) i mesures d’alt nivell (nombre de passes, si estem ajaguts o dempeus, dormint o no). Per a obtenir la posició de la persona sota estudi en entorns exteriors s’empra un receptor GPS de reduïdes dimensions. Finalment per a l’anàlisi de l’activitat a l’oficina s’empra una càmera web que captura àudio i video, a més d’un software capacitat per enregistrar en qualsevol moment els programes que l’usuari executa i les tecles que oprimeix.

Els sensors comercials emprats, que treballen cadascun a la seva freqüència de mostreig i seguint els seus propis criteris temporals, requereixen un preprocessat. Es fa necessari un pas de sincronització de dades ja que l’anàlisi s’ha de fer sempre prenent les mostres de tots els sensors en un determinat moment*. El Projecte Final de Carrera també inclou uns programes (interfícies d’usuari Matlab®) que permeten visualitzar les dades posteriorment al pas de sincronització.

---

*L’usuari recull les dades i al final del dia les volca a un ordinador on són processades
*La precisió escollida és d’un segon per a tot l’anàlisi tot i que s’ha la realitzat codificació necessària per poder treballar a freqüències menors i majors
Posteriorment a l’etapa de sincronització és imprescindible una etapa d’extracció de la informació d’entre les dades (*Feature Extraction*), per tal d’obtenir les dades que sotmetrem al procés de *clustering*. Resulta evident que no podem processar tota la informació (que en ocasions supera el Gigabyte per jornada) i hem de fer ús d’aquesta etapa.

Un cop les dades estan preparades emprem un sistema d’agrupació o *clustering* basat en la minimització d’una funció objectiu† corresponent a un algorisme de contrastada eficàcia. Donades les característiques de la informació a classificar s’afeiran dos termes addicionals a la funció d’error, tal i com es proposa en algunes publicacions recents sobre el tema. En concret s’empraran termes de coherència temporal i d’aprenentatge supervisat. De la combinació dels dos termes i l’algorisme inicial resultarà l’algorisme que ens servirà per detectar les activitats de l’usuari sota estudi.

Finalment un pas de postprocessat basat en mètodes estadístics com l’Algorisme de Viterbi‡, ens permetrà obtenir la classificació final.

Per tal de d’avaluar el comportament del sistema es varen enregistrar 7 dies complets per a dues persones diferents. Els usuaris sota estudi eren també responsables d’etiquetar les activitats realitzades el mateix dia. Els resultats obtinguts per l’algorisme que es proposa són contrastats amb l’etiquetat fet prèviament pel propi usuari i comparats amb d’altres sistemes de classificació que son a l’avantguarda del *clustering*.

És encoratjador veure que, tot i restringir el problema considerablement, s’assoleixen nivells de deteCCIó al voltant del 90%, fet que permet afirmar que es tracta d’un sistema que es pot emprar amb gran eficàcia.

La població dels països desenvolupats està envellint en mitja, les piràmides de població abandonen la forma que els dóna nom. De fet algunes prediccions apunten a que al 2015 un terç de la població europea estarà per sobre dels cinquanta anys. Això vol dir que hem d’estar preparats per tenir cura de la gent gran. Aquest projecte proposa un sistema que, entre moltes altres aplicacions, pot ser utilitzat per fer front a la perdua de memòria i la malaltia d’Alzheimer. En definitiva la línia d’investigació d’aquest Projecte Final de Carrera, encaminada a fer màquines conscients de l’estat de les persones, serà una eina de

---

† Funció objectiu, d’energia o d’error.
‡ Convenientment adaptat a les necessitats
gran utilitat en un futur no molt llunyà.
Acknowledgements

After five years, studying two different disciplines in two countries, I have learned one thing: I could never have done any of this without the support and encouragement of a lot of people.

My advisor, Fernando de la Torre, while wading through his ocean of responsibilities, always found time to guide me and reassure I was going through the right path. He gave me the opportunity to come to Carnegie Mellon University and have a wonderful time in Pittsburgh. The constance on his work and the patience he had to make me understand all that sound obscure to me have this Project as a result. I am thankful to him.

I thank Albert Oliveras, director of this project, for his invaluable feedback on this document and the final preparation of this work.

The first day I assisted to University was 10/11/2001, first day of a completely new page of both personal and contemporary history that will always be in my mind. Today, almost finishing my work as a Telecommunications Student, it is time to take a break in this path, look back to see what is already walked in order to face successfully any upcoming events.

These years I’ve spent at ETSETB in Barcelona have reported me lots of great moments, friends and knowledge but, over all these, I gained experience to face life and shaped myself as a person.

For a graduate student, his fellow graduate students are the key influence on his academic development. Now that I am probably closing a stage of my life, I would like to thank many of the people who made possible all this work and the building of my own personality. I would like to thank Daniel for its continuous work, I owe him my constance and academic constance.
results. He was an incredible mate both at graduate school, at work in Abertis Telecom and a great colleague, someone I could always count on. I am also really thankful to Tony, the endless hours at the bunker and the countless practical assignments we made together, sharing the croissants from the supermarket. I really enjoyed doing the double degree program in Telecommunications and Electronics together. I would not like to forget Bea, Ricard, Ricardo, Sandro and all the people I am missing from ETSETB, they really shape the way I am. Thank you all, I could never have done any of this without your support and encouragement.

I do not want to miss the invaluable moments I had at Abertis Telecom, where I learned a lot and had a daily close relationship to lots of great people. Charles, Matamala, Falcon, Balletbo, Jaume and Carme are only some of them. I thank you all the moments you wasted because of one of my doubts, the breakfast together and the huge amount of knowledge you transmitted to me without even noticing.

I am also really thankful to the family I found in Pittsburgh. The godfathers Ari and Juan, the uncle Jose and the aunt Esther, all the mates I shared moments with (Alvaro, Joan, Jose Mari, Juanjo and Titi, Carlos and Eli, and all those I am forgetting), they all made me find another home in the US. I will never find a way to thank them for all the moments they stuck in my mind in these 10 months of internship.

This work would have not certainly been possible without the support of my best friend, Oriol Vinyals. I thank him all the moments I had during 5 years of University and the patience and perseverance to convince me to go to Pittsburgh together. I will always keep in my mind the nights we spent studying for the exams at ETSETB, having dinner at the Rovell de l’ou, making promotions together and going to talk with our professors. You have been my friend, my colleague, my coworker, my flatmate and a never-ending source anecdotes. I also have to thank him for the final reviews on this text.

I would also like to express my thanks to the two women who stood next to me during all this time. First to Patricia, who shared with me 4 wonderful years. Thanks for keeping next to me all this time, I really wish the best to you. Secondly to Maria, who is an incredible person and scientist I had the privilege to share the life with. He made me state
that whenever you are happy the rest will follow.

Finally, I would like to dedicate this work to my family. Thanks to my parents for the high standards they raised me with. Thank you for your guidance and the invaluable inheritance you give me, my formation. As my father says, this is only the first step in a life where learning is a never-ending task. If I ever have children I only hope that I can be half the parent that you two have been to me.

Special mention to my sister Helena, who always had the right piece of advice in the right time. You have always opened the path I have followed. Thank you for your unending support over long distance or short.§

§[Translation to Catalan of the last two paragraphs]

Finally m’agradaria dedicar aquesta memòria a la meva família. Gràcies als meus pares pels valors en els que m’han educat. Gràcies per les recomenacions i l’herència que tinc el privilegi de gaudir, la meva formació. Com diu el meu pare, aquest és només el primer pas en una vida on mai pares d’aprendre. Només desitjo que si mai tinc fills, pugui ser la meitat de bon pare del que vosaltres heu estat amb mi.

Voldria fer una especial menció a l’Helena, la meva germana, qui sempre ha tingut el consell adequat en el moment adeuat. Tu has obert sempre el camí que jo he seguit. Moltíssimes gràcies pel teu inesgotable suport tant proper com a la distància.
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.3 Classifier</td>
<td>24</td>
</tr>
<tr>
<td>2.3.1 Synchronization</td>
<td>24</td>
</tr>
<tr>
<td>2.3.2 Clustering</td>
<td>25</td>
</tr>
<tr>
<td>2.3.3 Distance Measure and Decision Taking</td>
<td>27</td>
</tr>
<tr>
<td>2.4 Post-Processing: Viterbi’s Algorithm</td>
<td>28</td>
</tr>
<tr>
<td>3 Implementation</td>
<td>31</td>
</tr>
<tr>
<td>3.1 Data Retrieval</td>
<td>32</td>
</tr>
<tr>
<td>3.1.1 Camera Capturer</td>
<td>33</td>
</tr>
<tr>
<td>3.2 Feature Extraction</td>
<td>34</td>
</tr>
<tr>
<td>3.2.1 VirtualDub Scripting</td>
<td>36</td>
</tr>
<tr>
<td>3.2.2 Matlab interface for Intel OpenCV</td>
<td>37</td>
</tr>
<tr>
<td>3.3 Synchronization</td>
<td>38</td>
</tr>
<tr>
<td>3.3.1 Sampling Rates</td>
<td>38</td>
</tr>
<tr>
<td>3.3.2 Time Reference</td>
<td>39</td>
</tr>
<tr>
<td>3.3.3 Criteria for Data Selection</td>
<td>39</td>
</tr>
<tr>
<td>3.4 Our Clustering Approach</td>
<td>39</td>
</tr>
<tr>
<td>3.4.1 Notation</td>
<td>40</td>
</tr>
<tr>
<td>3.4.2 Objective Function</td>
<td>40</td>
</tr>
<tr>
<td>3.4.3 Spatio-temporal clustering</td>
<td>41</td>
</tr>
<tr>
<td>3.4.4 Adding semi-supervised information</td>
<td>44</td>
</tr>
<tr>
<td>3.4.5 Initialization</td>
<td>45</td>
</tr>
<tr>
<td>3.4.6 Synthetic Example: Temporal Term</td>
<td>46</td>
</tr>
<tr>
<td>3.4.7 Synthetic Example: Semi-Supervised information</td>
<td>48</td>
</tr>
<tr>
<td>3.5 Similarity Measure</td>
<td>49</td>
</tr>
<tr>
<td>3.6 Viterbi Algorithm Based Decision</td>
<td>50</td>
</tr>
<tr>
<td>4 User Interface</td>
<td>53</td>
</tr>
<tr>
<td>4.1 Data Inspector</td>
<td>54</td>
</tr>
<tr>
<td>4.1.1 Data Visualization</td>
<td>55</td>
</tr>
<tr>
<td>4.1.2 Labeling Tool</td>
<td>58</td>
</tr>
<tr>
<td>4.1.3 Results Visualization</td>
<td>59</td>
</tr>
<tr>
<td>4.2 Web Help</td>
<td>60</td>
</tr>
<tr>
<td>5 Data Collection and Analysis</td>
<td>63</td>
</tr>
<tr>
<td>5.1 Practical Issues in Data Collection</td>
<td>63</td>
</tr>
<tr>
<td>5.2 Results</td>
<td>65</td>
</tr>
</tbody>
</table>
## TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.2.1</td>
<td>Running the Algorithms</td>
<td>66</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Accuracy Measure</td>
<td>66</td>
</tr>
<tr>
<td>5.2.3</td>
<td>Output Analysis</td>
<td>67</td>
</tr>
<tr>
<td>6</td>
<td>Summary, Conclusions and Future Work</td>
<td>71</td>
</tr>
<tr>
<td>7</td>
<td>Bibliography</td>
<td>75</td>
</tr>
<tr>
<td>A</td>
<td>Main Clustering Algorithms</td>
<td>85</td>
</tr>
<tr>
<td>A.1</td>
<td>Hierarchical Clustering</td>
<td>85</td>
</tr>
<tr>
<td>A.2</td>
<td>K-means</td>
<td>87</td>
</tr>
<tr>
<td>A.3</td>
<td>C-means</td>
<td>90</td>
</tr>
<tr>
<td>A.4</td>
<td>Other Clustering Techniques</td>
<td>91</td>
</tr>
<tr>
<td>B</td>
<td>Clustering Application: Recognizing People in a Meeting</td>
<td>93</td>
</tr>
<tr>
<td>C</td>
<td>Paper Submitted</td>
<td>97</td>
</tr>
</tbody>
</table>
List of Figures

1.1 Memex device designed by Vannevar Bush. .......................... 5
1.2 Examples of devices used in prior work. From left to right Cameras used in MyLifeBits Project, SenSay cellular phone layout and sensor jacket developed at CMU. ............................................................ 6

2.1 Canonical Scheme for Pattern Recognition from [16]. .................. 14
2.2 Detail of Classifier Block in the Canonical Scheme for Pattern Recognition. 15
2.3 Sensing Platform, sensing devices and examples of data acquired with them 16
2.4 Body sensor fixed on the arm (left) and position of the Sensors at the backside of SenseWear (right). ................................. 17
2.5 Delimitation of the interior area versus the whole image. Values in percentage of the frame dimensions. .............................. 20
2.6 Classifying structure for sound Data ................................. 21
2.7 Screenshot of Activity Monitor from Softactivity .................... 22
2.8 Wearable GPS Personal Navigator .................................. 23
2.9 Example of GPS Data exported and plotted on GoogleEarth© .... 24

3.1 Block diagram of the system architecture. .......................... 32
3.2 CaptureTest main window and detail of some of the menus. ........ 34
3.3 Feature Extraction layer. Position in the global architecture scheme and interaction with other layers. ............................ 35
3.4 Example of face detection using Intel OpenCV ...................... 37
3.5 Multiresolution scheme ............................................. 46
3.6 Toy Data and 2-cluster classification for K-means, DCA and DCA with temporal coherence. .............................................. 47
3.7 Bidimensional Data. Left: natural clustering without supervision, right: clustering imposing two subgroups are in the same cluster. ........ 48
4.1 Role of the User Interface in the general Project Scheme and interaction with other modules. .................................................. 54
4.2 Overview of the Data Inspector GUI .................................................. 55
4.3 Labeling Panel: position in the GUI, detail of the panel and legend ........ 58
4.4 Results Panel: position in the User Interface. ..................................... 59

5.1 Top: Webcam view of two users that collected data for this Thesis’ system. The one in the right is adjusting the body sensor’s strap. Bottom: adjustments in the sensor’s position and wearability (camera and GPS) ... 64
5.2 Labeled Data (Top) and Output of the best recognition algorithm (Bottom) 66
5.3 Accuracy versus number of labeled samples in the supervised term (number of supervised samples over total number of samples in permil (%)) .... 68

A.1 Example of data tree or dendogram .................................................. 86

B.1 Block diagram for people recognition in a meeting using CAMEO. Picture of the CAMEO device at the top right hand corner. ...................... 94
B.2 People Recognition Input (above) and output (below) ......................... 95
List of Tables

1.1 List of defined activities or states. ................................. 9

2.1 Typical parameters for audiovisual recording. ..................... 18
2.2 Features extracted from video data. ............................... 19
2.3 Different Classifications for clustering algorithms. .............. 26

3.1 Software Required to Retrieve Data from each Sensor. ........... 32
3.2 Software Required for Feature Extraction .......................... 35

4.1 Panels and Utilities from figure 4.2 ............................ 55

5.1 Accuracy for the different clustering methods .................... 67
5.2 Accuracy for the different states defined ......................... 69
Chapter 1

Introduction

1.1 Summary

Human activity recognition, far from a futurist idea, is a feasible application with the current technology. Although it might be seen as an innovative problem, a brief search in the main academical and professional search engines will lead us to discover that activity recognition is a highly studied field over the scientific community. In fact some universities have Human Computer Interaction workgroups where Pervasive Computing is their main field of research.

The relevance of human-machine interaction has been continuously growing for the last few years and will surely become a crucial field in the near future. In a world where everyday we have more electronics surrounding us (not only cellular phones, computers, washing machines and the microwave ovens but also many other electronically operated devices like lifts and cars) it is interesting to try to integrate the world of electronics to the daily life, because it seems clear that the other way (the adaptation of our lives to the world of electronics) has already happened.

Nevertheless entering the world of Activity Recognition means getting to combine high level knowledge from fields such as Machine Learning, Signal Processing or Sensors, which is not usually trivial. Real systems, compared to theoretical studies, almost always present unexpected difficulties that challenge the engineer’s ability to solve problems. In this sense
the goal of this Project is not to revolutionize the Activity Recognition field because neither the dedication nor the formation of the author are in position to do so. On the contrary this work proposes a different approach to the community with a tiny step, the author’s bit, towards the overall goal.

The importance of Activity Recognition drove the author of this Project to focus on this area, with the only objective of adding a little contribution in the development and research in it.

The main goal of this Project is classifying the activities held by a person during a day. This task is the first step towards the next generation of electronic products. The applications of this technology affect areas such as medicine, social sciences, statistics, social-based economy and elderly caring.

Let’s imagine for a second that our telephone does not ring when we are in a meeting or in a productive moment, that our fridge adapts the temperature of our favorite soda based on the activity we did in a particular day, that our car does not let us drive if we have been drinking and that our hospital recommends us to follow a different timetable because is less probable for us to have a heart attack. This Project follows a line that will lead to this services we can just dream about.

This Project reports the use of four low and mid-level sensors for the activity analysis of a person in an office environment briefly described in the following lines. We use a device from Bodymedia© Corp as a physical awareness device which gives information about activity related vital variables (such as galvanic skin response, heat flux, near and far temperature or 2-axis acceleration) and high level measures (number of steps in a minute, sleep indicator and lying indicator). Outdoors position is provided by a strap-fixed GPS receiver. For office activity analysis the system uses a web camera (capturing audio and video) and a software inspector capable to record programs used and keystrokes pressed.

All commercial sensors used in this project work in different sampling rates and using different temporal criteria. This fact makes a preprocessing step mandatory. To get to know which activity is held in a single moment we need to know the samples for all the
sensors in that particular moment\textsuperscript{1}. A Matlab\textsuperscript{2} user interface is provided for visualization purposes once the preprocessing step is done.

It is obvious that we should only process the relevant information contained in the data (sometimes over a Gigabyte per day). Therefore feature extraction is the following mandatory stage.

Once the information is obtained, a clustering algorithm that minimizes an energy function based upon a published and trustworthy method is used to build clusters with certain features. Taking into account the trends of the information we want to classify, two additional terms proposed by the author will help providing a better output. Temporal coherence and supervised clustering terms, combined with the basic algorithm will provide us the activities developed by the user under study.

After estimating the distance between the incoming data and the labeled database built in the clustering process, a decision taking step based upon statistic methods such as Viterbi Algorithm, will lead us to the final grouping.

It is important to point out that the work reported in this document is oriented in a postprocessing point of view\textsuperscript{2}, which allows the use of a high level programming language. Future work will require a low level implementation allowing real-time skills. An interface to communicate the different devices that take part in the system will also have to be designed.

In order to evaluate the overall behavior of the system seven days for two different people were recorded. The users under study were also responsible to label the activities developed in the days recorded. The results of the algorithm proposed are compared with the labeling and contrasted with other state-of-the-art clustering algorithms.

It is encouraging knowing that, although there are several constraints on the problem, we achieve around a 90\% of correct state classification. This allows us to state that we can build a system able to achieve great accuracy.

Western Countries Population is greying, in fact according to predictions by 2015 one third of population of working age in Europe will be 50 years and over. This means that we

\textsuperscript{1}We chose a precision of one second for the whole analysis. Code for precision changes (more or less precision) is also provided.

\textsuperscript{2}The user collects data and at the end of the day data is put together in a computer which processes it.
must be prepared for elderly caring. Lost of memory and Alzheimer’s Disease are problems that technology can potentially help handling. The work reported in this report, which tries to make computers aware of the state of the people, is following these lines of action and will be really helpful in a near future.

1.2 Previous Work

Any design, no matter what area or field we are working in, should keep the natural evolution of any technical work and must naturally leverage off the work that preceded it. This chapter will focus on relevant context-aware applications, their use of context and their ability to support reuse of sensing technologies in new applications. This chapter will also introduce the data processing techniques used to solve the activity recognition problem.

This section is divided into the two sources of information that feed this project: Computer Awareness and Clustering.

1.2.1 Computer Awareness

Although this section is named after one of the fields of study that concerns to this Project, several other areas could have given the title to this review of prior work. In fact, fields like Human Computer Interaction, Interruptability, Pervasive Computing, Computer Awareness, Wearable Computing are going to be indistinctly referenced in this section.

The basic aim of Computer Awareness and wearable computing is to build intelligent machines that provide automatic and autonomous support in people’s everyday lives. Accurate and comprehensive recognition of user’s context is an important step towards that goal. In the following, several applications of wearable computing and computer awareness and their dependence on context recognition are discussed.

Back in 1945, it is interesting to look at what we can consider the origins of daily recording. Vannevar Bush\textsuperscript{3} has left his footprint in many lines of academic thought, and

\textsuperscript{3}Vannevar Bush (March 11, 1890 - June 30, 1974) was an American engineer and science administrator, known for his political role in the development of the atomic bomb, and the idea of the memex seen as a pioneering concept for the World Wide Web. He build one of the first analog computer, advised Claude Shannon and founded the National Science Foundation, by his recommendation in a technical report[8].
1.2. Previous Work

ours is no exception. V. Bush proposed a system called Memex\(^4\) (which stands for Memory Extender) a device electronically linked to a library able to display books and films from the library and automatically follow cross-references from one work to another. Some state that Memex is the origin of hypertext because the theoretical approach proposes a linkage system similar to the one implemented in the hypertext. But focusing on what concerns to this project, V. Bush proposed that the Memex user could generate new information (on microfilm) in order to keep a continuous record of activities. Figure 1.1 shows a scheme proposed by V. Bush of this device.

V. Bush identified three really important useful skills about the information to be recorded, Continuous Recording, Complete Storage and Accessibility.

The revolutionary ideas of Bush can be found today in several projects, in what we could call 'enlarged intimate supplement to memory'. MyLifeBits (see figure 1.2), Memories for Life, Total Recall[^6] and What Was I Thinking are only some of the projects that can be found on the Continuous Archival and Retrieval of Personal Experiences (CARPE) website[^5].

Along these lines it is interesting to point out the work by Krause et al [30], who propose a method for unsupervised and dynamic identification of physiological and activity context based on data collected from the BodyMedia© SenseWear armband.

Most of these projects, in a framework we can call 'Diary Applications', focus on orga-


\(^5\)[http://www.sigmm.org/Members/jgemmell/CARPE](http://www.sigmm.org/Members/jgemmell/CARPE)
Health Care and Interruptability are two fields where wearable computing and computer awareness can also help.

Extremely high costs of healthcare and an increasing shortage of caregivers are the main drivers for these fields to succeed in helping wellness. For diagnosis and medical studies a history of continuous sensor readings and activity patterns is more accurate than a periodic questionnaire and can very well complement information won by verbal discussions with the patient. An overview of the current effort done in this lines is reported in [48].

Likewise the medical environment, a potential context-aware mobile devices application is to proactively present information when the user needs it, a field we call Interruptability. An increasing number of devices and applications are competing for the user’s attention, phone calls, reminders, email notifications, voice messages, instant messaging services, etc. Sensors such as microphone and video have been used in a Wizard of Oz Study [21] to determine when an office worker can be interrupted. In this particular case self-reports are used to identify and evaluate a statistical model for interruptibility assessment in an stationary scenario. In [22], office-based sensors (such as a silence detector, phone activity, door status and analysis of desktop computer usage) are used to evaluate the proposed model. However is interesting to consult the work by Siewiorek, Smailagic et al on SenSay[56], a context-aware cellular phone, shown in figure 1.2.

Social sciences, similarly to the two previously explained fields, find in pervasive computing an invaluable source of information for analysis. Measuring social interest based on
audio features, head movement and Galvanic Skin Response (GSR) has been reported by Madan et al in [39]. Oliver and Horvitz’s approach [45] report results for online office activity recognition using selective perception from simple computer peripheral devices (mouse, keyboard and webcam).

From a social integration point of view, wearable cameras have been used to recognize American Sign Language [7, 58]. In [44] video data is used for facial expression recognition and face detection whereas audio is proposed for Human Computer Interaction applications in [4].

By contrast, a much simpler application of these technologies lies in tracking. Accelerometers are common sensors to infer activity and have been used in several human computer interaction projects ([20, 12, 3] for example). Through them activities such as sitting, standing, lying, walking and running can be easily extracted when operating at high sampling rates. Similar accelerometer approach was adapted to a sensor jacket by Farringdon, Moore et al in [20] (see figure 1.2) where the task to recognize where only positional (sitting, standing, lying, walking and running).

Several tracking systems can be found using GPS devices, because Global Positioning System (GPS) localizers have also become common sensors in Human Computer Interaction studies [2, 30, 57, 40]. In [19] any kind of trajectory (sign language data measurements, GPS coordinates, etc) is classified and recognised using Hidden Markov Models [50].

Finally, completely out of the academic and even the real world, it is funny that examples of activity recording can even be found in Science Fiction literature and Movies. Robert Sawyer, in his book Hominids[53], relates a transmission system that gives information about location and 3D images of people’s activity. Hollywood has also made films on activity report, The Final Cut\(^6\) is the most recent example. Even though they can not be seen as prior work in the field, they are related previous work and the view of the upcoming future by writers and directors that are usually away from the academic environment.

---
\(^6\)The Final Cut (2004), Directed by Omar Naim; Cast: Robin Williams, Mira Sorvino; Website: \(http://www.finalcutfilm.com\)
1.2.2 Clustering

Clustering is one of the main problems in Machine Learning and has been widely studied by the research community. A review on clustering is beyond the scope of this chapter, the reader is forwarded to Chapter 5 and appendix A to better understand the basics of this problem.

Clustering is the classification of objects into different groups, the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait - often proximity according to some defined distance measure.

Clustering has been applied to many fields and has evolved in many algorithms and variations. For the approach of this work is relevant to take a look at the prior work on spatiotemporal and semi-supervised clustering.

Spatiotemporal clustering (clustering of time series or spacial data) has been widely applied in biological signals analysis, specially to ECG and EMG signals [10] and magnetic resonance data clustering [41]. Still, work on time series has been developed recently for several other fields [34].

Temporal Clustering has also been the object of research in applications closer to the general users. The analysis of supermarket customers routines using this technique [36], temporal studies for web usage [35] or clustering homemade digital photos [28] are some other applications of this temporal method.

Spatial clustering has found many areas of application, but among them astronomy [46] and Geographical Information Systems (GIS) [18] have taken advantage of the evolution of this field for its research and activity.

By contrast semi-supervised learning combines labeled and unlabeled data during training to improve performance and is applicable to both classification and clustering. In unsupervised clustering an unlabeled dataset is partitioned into groups of similar examples, typically by optimizing an objective function that characterizes good partitions. In semi-supervised clustering, some labeled data is used along with the unlabeled data to obtain a better clustering. Relevant work (in which this Project has some of its origins) has been recently done on supervised or semi-supervised clustering [31, 14, 25]. It is worth to point
out that the Machine Learning Group at University of Texas is working hard in this field, being a referent for semi-supervised clustering.

1.3 Constraints of this Project

The aim of this Project is recognizing daily activities using a set of simple and independent sensors. The information is collected and post-processed in a Personal Computer.

Clustering and its spatiotemporal and supervised variations are used to determine which of the states shown in table 1.1 the user is performing.

<table>
<thead>
<tr>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sleeping</td>
</tr>
<tr>
<td>Walking</td>
</tr>
<tr>
<td>Working no PC</td>
</tr>
<tr>
<td>Internet Surfing</td>
</tr>
<tr>
<td>Working PC</td>
</tr>
<tr>
<td>Talking</td>
</tr>
<tr>
<td>Restroom</td>
</tr>
<tr>
<td>Phone</td>
</tr>
<tr>
<td>Away</td>
</tr>
</tbody>
</table>

Table 1.1: List of defined activities or states.

Although many other states based on partial information from some of the sensors could be defined\(^7\), the goal of this Project is to correctly recognize a reduced set of activities to show that the theoretical background on clustering and new approaches work properly.

Machine learning typically regards data clustering as a form of unsupervised learning. In contrast one of the novelties reported in this Project is the use of supervised information for clustering.

Simplicity is also one of the constraints of this work. Following these lines the sensing layout consists of commercial (not custom) sensors, the features extracted are as simple as possible and there is a reduced set of states to be predicted. This feature of the job done shows that simple systems can work with a reasonable accuracy. Adding trends to the

\(^7\)For example practising sports from the physical awareness device, going to supermarket from the GPS position information or traveling by car from the GPS speed.
system (increasing the number of sensors or the number of features, making the algorithm more complex, etc) will definitely better the results in a higher spectrum of possible states. This will be the lines to follow for further work and beyond the scope of this Project.

The partial goals achieved in this Project are the following:

- Implement an specific module (or use the manufacturer’s one) for each sensor in order to retrieve the data collected.

- Develop an autosynchronization tool to synchronize the data extracted.

- Build a User Interface for data visualization.

- Make a feature extractor for each of the data channels.

- Adapt pre-existing clustering algorithms and implement the spatiotemporal and semisupervised behaviors.

- Adapt the User Interface to allow data supervision.

- Data Collection Process

To the best of the author’s knowledge, no work has been published that makes use of spatio-temporal clustering to automatically recognize activities.

1.4 Organization of this Project

The aim of the writer is offering the information as easy for the user as possible. Following these lines this document is organized in self-contained and modular chapters. This document’s architecture allows the reader to target only the information he is interested in.

This first chapter settled the background of the work done for the Project. This means that after reading it the reader should be able to understand which are the objectives and constraints of the information he will find in this document. It is also important that the reader is aware of history and the state-of-the-art technologies in the field of activity recognition.
The second chapter reviews the main tools (both theoretical and practical) that will be used and cited all along the document. The aim of the chapter is describing which tools is the author working with, which are the media used for data acquisition and the main signal processing basis upon which this project will work.

A third chapter details the implementation of our approach, showing the architecture followed for activity recognition. With a deep explanation of each of the stages of this architecture, the reader will get to know which are the problems this Project has faced and which are the solutions proposed. This chapter takes only into account the work related to algorithms involved in this Project (code used, algorithms, interaction between blocks, etcetera).

In the completely opposite point of view, chapter 4 will detail the User Interface developed, its uses and features. The chapter will not explain which information is shown but how is it shown. Processing several data is useless without a visualization module to inform the user.

Practical Issues on data collecting and results and evaluation of the overall system are reported in chapter 5.

Finally the last chapter will take conclusions from a retrospective point of view on this work, suggesting the lines of research in which this project could be continued.
Chapter 2

Platforms and Tools

This chapter will take a look to the formal tools, both theoretical and practical, used in this Project. The objective of the chapter is settling a framework and describing which stages should be covered by an Activity Recognition System.

A first section will introduce the classical Recognition Scheme, giving general skills of each of the blocks involved in it. This section will open the path to organize the rest of the chapter. From then on, the chapter will explain thoroughly each of the most important blocks listed in this section.

The second section details the Sensing Platform that is going to be used for this Project. Not only the specifications of the devices used but also the kind of information that can be extracted from them will be explained in this section. The features that will be used in further steps are also detailed together with the description of the sensors. The Sensing Platform and the Feature Extraction stages are two of the main blocks from the Canonical Recognition Scheme.

The third and perhaps most interesting block of the canonical Recognition Scheme will be covered in the third section of this chapter. The Classifier, split into Synchronization Process, Clustering and Distance Measure and Decision Taking, is the centre of this section. Again, the Classifier has a relevant paper in the block scheme proposed in the first section.

Additionally the last section of this chapter will offer a non-canonical block in a recognition framework: post-processing. The particular case of Viterbi Algorithm will be intro-
duced. This brief overview is due to the relevance of this block for the algorithms developed for activity classification.

2.1 Canonical Recognition Scheme

Pattern recognition aims to classify data (patterns) based on either a priori knowledge or on statistical information extracted from the patterns. The patterns to be classified are usually groups of measurements or observations, defining points in an appropriate multidimensional space.

A complete pattern recognition system, as stated in [16], consists of a sensor that gathers the observations to be classified or described; a feature extraction mechanism that computes numeric or symbolic information from the observations; and a classification or description scheme that does the actual job of classifying or describing observations, relying on the extracted features.

Following the mentioned system, [16] proposes the scheme in figure 2.1 as a generic architecture. This is what we call Canonical Recognition Scheme.

![Figure 2.1: Canonical Scheme for Pattern Recognition from [16].](image)

The Classifier Block is detailed in figure 2.2. The recognition process depicted in this figure is based upon a distance measurement from each of the features (incoming data) from a Data Base. From the distance measurement the system can take further decisions.

There are many approaches for the Feature Data Base population. General classification algorithms use labeled databases, populated from training processes. Applications with lots of data usually perform a clustering step.
2.2 Sensing Platform

Figure 2.2: Detail of Classifier Block in the Canonical Scheme for Pattern Recognition.

Although the scheme shows two different blocks for distance measurement and decision taking, usually they can not be distinguished and they are very algorithm dependant.

It has also to be mentioned that Source Real Data coming from the real world has to be synchronized if it is not coming from a single sensor. This fundamental stage is important for recognition but crucial for coherence among data.

The following sections will go on detail for some of the blocks shown in figures 2.1 and 2.2.

2.2 Sensing Platform

The hardware sensing platform used in this Project is based on low-cost sensors and leverages off commodity hardware. As the project is designed to work in an office environment and the processing of the data is done after the collection (not at the same time) some sensors are wearable and some of them are fixed in the office. This provides the following sensing layout:

- Physical awareness device (Wearable)
- GPS receiver (Wearable)
- Web Camera (Fixed)
- Software Inspector (Fixed)
Although other sensors were initially included in the sensing layout of the Project \(^1\), the author initially stated that the layout listed above should give enough information for human activity recognition. The final decision was also constrained by a combination of wearability, amounts of information gathered, cost and availability as main factors.

It is obvious, though, that the more information gathered the easier the classification step will be, but anyway there is a tradeoff between wearability and amounts of information collected.

---

\(^1\) An Omnidirectional Camera and a personal wearable recorder were used in the first version of the project. They were discarded afterwards because of the amount of information recorded (over 4 Gb/day for the Omnidirectional Camera) and the portability of the end user (he has to carry the wearable GPS and the body sensor, the personal recorder was too much).

---

Figure 2.3: Sensing Platform, sensing devices and examples of data acquired with them

In this section we describe our multimodal acquisition system, its parameters and the features we extract for clustering.

Figure 2.3 shows the sensing platform and examples of the kind of data we can extract from each of the devices. At the same time the figure shows the timeline (on an orange
2.2. Sensing Platform

Figure 2.4: Body sensor fixed on the arm (left) and position of the Sensors at the backside of SenseWear (right).

background) with a user under study in different instant of a day.

2.2.1 Body Sensors

We use SenseWear armband [59] from BodyMedia (shown in fig. 2.4) as a physical awareness device. SenseWear armband combines five different sensors: two accelerometers, galvanic skin response, skin temperature, heat flux, and near-body ambient temperature.

The accelerometer is a 2-axis (transversal and longitudinal) micro-electronical-mechanical sensor (MEMS) device that measures motion. The Galvanic Skin Response (GSR) represents electrical conductivity between two points on the wearer’s arm. GSR reflects evaporative heat loss and can be an indicator for the onset, peak, and recovery of maximal sweat rates. Skin conductivity is affected by the sweat from physical activity and by emotional stimuli. Skin temperature is measured using a highly accurate thermistor-based sensor located on the backside of the armband near its edges and in contact with the skin.

The proprietary heat flux sensor measures the amount of heat being dissipated by the body. In particular, it measures representative values of the heat convection part of the total thermal energy dissipated to the surroundings. The near-body temperature sensor measures the air temperature immediately around the wearer’s armband using a highly accurate thermistor-based sensor. Near-body ambient temperature is defined as the temperature at the outer edge of the heat flow sensor.
The SenseWear is synchronized\textsuperscript{2} with the computer and it is worn during the entire day on the back of the upper right arm (see fig. 2.4). Data can be easily retrieved from the SenseWear and converted to Microsoft\textregistered® Excel format using the proprietary InnerView\textregistered Research Software from BodyMedia. A simple Matlab\textsuperscript{©} code can easily parse the information from the spreadsheet to a Matlab\textsuperscript{©} compatible format.

Additionally, the InnerView\textregistered Software from BodyMedia allows data downloading from the sensor and gives a high-level information channels, such as \textit{sleeping, lying, Physical Activity} or \textit{Number of steps per minute}. The information of these channels is extracted using Machine Learning Techniques embedded on the proprietary software.

As features for clustering purposes from the SenseWear sensor we use the high level information (sleeping, physical activity, lying and standing up) as well as features from the continuous, low-level channels (such as peaks in the GSR and mean values over time).

### 2.2.2 Video and Audio

We use the Logitech Quickcam Pro 4000 to record audio and video from the user under study’s desktop. Table 2.1 shows the typical specifications\textsuperscript{3} for the audio-video recording. Typically, one hour of audio-visual information takes about 210 Mb (depending on the codec), most of which is the video signal.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video Frame Size</td>
<td>$320 \times 240$</td>
</tr>
<tr>
<td>Video Frame Rate</td>
<td>15 frames per second</td>
</tr>
<tr>
<td>Video Compressor</td>
<td>Microsoft MPEG-4 V3</td>
</tr>
<tr>
<td>Audio Channels</td>
<td>Mono</td>
</tr>
<tr>
<td>Audio Sampling Rate</td>
<td>22.05 kHz</td>
</tr>
<tr>
<td>Audio Sample Rate</td>
<td>8 bits</td>
</tr>
<tr>
<td>Audio Compressor</td>
<td>GSM 6.10</td>
</tr>
</tbody>
</table>

Table 2.1: Typical parameters for audiovisual recording.

\textsuperscript{2}Each time it is connected to the computer to download data or configure its parameters, the device gets the time and date from the personal computer.

\textsuperscript{3}A C\# application was developed in order to record with the web camera (see subsection 3.1.1 on page 33). This application used DirectX to record audio and video and fixed a precise timestamp for synchronization purposes.
2.2. Sensing Platform

Given a video sequence, our main goal is to detect whether the user is in front of the camera or not. This is done using existing Intel® OpenCV code for face detection [61] (filtered using color information) and other features, including motion and RGB statistics.

Learning behaviors from training videos\(^4\), 7 features were selected. The selected features are listed in table 2.2. They are conveniently thresholded using the training videos to provide the seven binary streams.

<table>
<thead>
<tr>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Whole Image red Channel (R) mean value</td>
</tr>
<tr>
<td>Whole Image green Channel (G) mean value</td>
</tr>
<tr>
<td>Whole Image blue Channel (B) mean value</td>
</tr>
<tr>
<td>Interior Image red Channel (R) mean value</td>
</tr>
<tr>
<td>Interior Image green Channel (G) mean value</td>
</tr>
<tr>
<td>Interior Image blue Channel (B) mean value</td>
</tr>
<tr>
<td>OpenCV Face detector</td>
</tr>
</tbody>
</table>

Table 2.2: Features extracted from video data.

In table 2.2 we refer to the whole image (taking the values of resolution from table 2.1) and the Interior image. This Interior image consists of a subimage of the frame captured. The subimage is extracted as shown in figure 2.5. Taking a rectangle of the 60% of the width per the 60% of the height and centering in the overall frame.

Fixing the camera on a strategic location, it has been tested from training videos that most of the face of the user is inside this interior area when he is in front of the camera.

Audio information is processed separately. We classify every 20ms into 4 different states: user talking, other people talking, typing and no-one talking. To do this task a hierarchical classifier is used (see figure 2.6). A Silence Detector based on signal power firstly separates voice from silence. In a second level we classify Mel Frequency Cepstral Coefficients (MFCC) of the non-silence 20ms-chunks using a Support Vector Machine [11], as proposed in [63].

In a training phase, we use the sound Classifier Support Vector Machine with one-minute sound files from each group, extracting twelve (12) cepstrum coefficients every 20ms with

\(^4\)Several sequences of different users staying in front of the camera in its usual position and other sequences with no user in front of the camera. Entering and leaving events were also performed.
The code used for Support Vector Machines is a mix of several sources adapted to the needs of the Matlab\textsuperscript{\textregistered} interface, but mainly extracted from [9].

The high time resolution of the audio testing (20ms), together with the overlapping windows and the need of a lower precision (1 second or more) binary stream as an input for clustering algorithms make a coarsening process mandatory. A set of histograms punishing the no-one talking state perform this task.

Finally it is interesting to point out that the audio and video information, given the trends of the acquisition process, do not need synchronization, which makes the data synchro a little bit easier.

### 2.2.3 Monitoring the computer

To record the interaction of the user with the computer we use Activity Monitor from Softactivity (\texttt{www.softactivity.com}), a monitoring software system for real-time monitoring and continuous tracking of users’ activities on networked computers. Although this software is originally designed for local area network (LAN) users’ control (see example for 5 users in figure 2.7), we configure the program to record and monitor a single computer in a

![Figure 2.5: Delimitation of the interior area versus the whole image. Values in percentage of the frame dimensions.](image)
2.2. Sensing Platform

Activity Monitor records a log file of the URLs visited, keystrokes, programs that the user runs and work duration in every application. These logs are easily exportable to formats that can be parsed. In addition to the information cited, the software is able to take shots of the user’s screen and take Remote Control for all the machines in the network, these features, though, are not used for this project.

From the computer information we can know which programs the user has been using in a given period of time. Each of the programs is classified every second in one of the following groups: work, non-work or internet surfing. The classification algorithm is pre-populated with common programs detected but every time a new program shows up the user is prompted to classify it in one of those groups. The look-up table is saved in a XML file and is easily modifiable.

On the other hand the number of keystrokes pressed by the user is also used as a feature for the clustering algorithm. This information is relevant to check if the user is in front of the computer or not because the execution duration of a particular program does not check if the user is actually in front or if the user left the computer with the program running.

The information extracted from this *sensor* are two streams indicating which kind of program is being used and the number of keystrokes in each time unit. In order to binarize
the information complementary binary streams can be extracted from the program-group vector and thresholding can be used for keystrokes.

2.2.4 Wearable GPS Logger

As a position and speed measuring device for outdoor environments we use a wearable wrist strap-fixed GPS receiver. With an accuracy of 49ft RMS (about 15 m) and an updating frequency of 1Hz the GARMIN Foretrex 201⁵ (see fig.2.8) meets both portability and logging capabilities. Its memory and batteries are dimensioned to record for more than a day. It has got serial connection to the computer in order to retrieve the stored data to a GPX standard format[24], which can be parsed by our system.

The data retrieval from the device can be done using any proprietary software (GARMIN, for example, has got its own software to retrieve the information, to be acquired separately) or using any standard compatible software, like EasyGPS⁶ (www.easygps.com). We use the EasyGPS tool which directly converts the data contained in the device to the GPX format, parsable by a Matlab© script.

Using the coordinates logged by the device (longitude and latitude) we directly obtain

---

⁵ Any commercial portable GPS device able to transfer data to a computer and output it in GPX format would work.

⁶ EasyGPS is trademark of TopoGraphix. © 2000-2006 TopoGraphix
the position of the user under study (with a very precise timestamp information) and we can estimate mean speeds at the sampling rate.

We always have to keep in mind the constraint that GPS logging is only working outdoors so it is only giving relevant information about where the user goes when he is not in front of the camera. Whenever no signal is detected, the user is assumed to remain in the office building.

For featuring purposes we define a set of common places (MyPlaces) where the user stays (typically user’s house and working place are the main common places). The features for this sensor are simply the distances from each one of the user’s locations during the day to the places defined in MyPlaces. The feature that will be an input for the clustering algorithm is a set of vectors (one per location pre-defined) taking binary values that show if the user is closer than 49ft from that specific location.

On the other hand a hard velocity estimation\(^7\) is also calculated out of the position. This data can be easily thresholded with different speeds for featuring purposes.

Although only MyPlaces’ streams are used for clustering purposes, the visualization possibilities of the position information gives us the opportunity to provide different visualization options. The visualization program done for this project (Data Inspector, see section 4.1 on page 54 in chapter 4) counts with a mapping tool for previously introduced maps\(^8\). A conversion for visualization using GoogleEarth\(^\text{©}\)'s (http://earth.google.com/) KML format, a popular Geographical Information System (GIS) software tool. An example of a

---

\(^7\)Distance between two samples divided by the difference between their timestamps.

\(^8\)Maps are entered using geo-referenced images, assuming equiscalavility. Codification for geo-referenciation and visualization in Matlab\(^\text{©}\) is provided with this Thesis.
track (portion corresponding to a trip from home to work by a user) is shown in figure 2.9, using GoogleEarth© software.

2.3 Classifier

As explained in section 2.1 (on page 14), the Classifier is a critical part of the Canonical Recognition Scheme. This section, following the blocks depicted in figure 2.2, will explain the use and necessities of those blocks. Additionally an implicit block (Synchronization) is justified, making a special emphasis to its relevance.

2.3.1 Synchronization

Working with multiple sensors instead of a single one implies a set of extra problems. The most evident and relevant one is synchronization.

The only objective of synchronization is fixing the same time reference for all the data collected with different transductor systems. The relevance of this step is really high, a failure in the alignment of the data would lead to high error rates in recognition.

Possibly classified as a Pre-Processing stage, synchronization is a step that should be performed before classifications, with the objective of providing the cleanest data possible.
2.3. Classifier

to the Classifier block.

Several techniques can be performed to achieve the synchronization process, also called alignment process. Handling different starting and ending time, different sampling rates and even asynchronous events are trends this block should have. In this field it is interesting to point out the Dynamic Time Warping [43] or DTW, widely used in speech recognition for synchronization and distance measuring.

All in all the relevance of an alignment process justifies its role and necessity in systems with several inputs.

2.3.2 Clustering

Clustering is a process of partitioning a set of data (or objects) in a set of meaningful sub-classes, called clusters⁹. Data clustering is a common technique for statistical data analysis, which is used in many fields, including machine learning, data mining, pattern recognition, image analysis and bioinformatics. For example, if we gathered a set of pebbles from a stream shore, noted their attributes of size, shape and color, and sorted similar pebbles into the same piles, we would be physically performing a cluster analysis. Each pile of similar pebbles would be a cluster.

Mathematical methods of cluster analysis accomplish this mathematically. Instead of sorting real objects, these methods sort objects described as data. Objects with similar descriptions are mathematically gathered into the same cluster.

There are three basic reasons for interest in clustering procedures. First, the collection and labeling of a large set of sample patterns can be surprisingly costly and time consuming. If a classifier can be crudely designed on a small, labeled set of samples, and then ‘turned up’ by allowing it to run without supervision on a large, unlabeled set, much time and trouble can be saved. Second, in many applications the characteristics of the patterns can change slowly with time. If these changes can be tracked by a classifier running in an unsupervised mode, improved performance can be achieved. Finally, in the early stages of

⁹In [51] Cluster Analysis is defined as a generic name for a variety of mathematical methods that can be used to find out which objects in a set are similar. Many other definitions can also correctly define Clustering.
an investigation it may be valuable to gain some insight into the nature or structure of the data. The discovery of distinct subclasses or major departures from expected characteristics may significantly contribute to the understanding of the data.

Several classifications can be applied to clustering algorithms. Some of them are shown in table 2.3, and more detailed information may be found in appendix A.

<table>
<thead>
<tr>
<th>Types of Clustering</th>
<th>Membership of the Data</th>
<th>Steps Performed</th>
<th>Starting Point (Hierarchical only)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Exclusive</td>
<td>Hierarchical</td>
<td>Agglomerative</td>
</tr>
<tr>
<td></td>
<td>Overlapping</td>
<td>Partitional</td>
<td>Divisive</td>
</tr>
</tbody>
</table>

Table 2.3: Different Classifications for clustering algorithms.

Considering the **Membership of the Data** as a classification criteria for clustering algorithms, two groups can be identified. In **Exclusive Clustering** (see K-means on appendix A.2) data are grouped in an exclusive way, so that if a certain datum belongs to a definite cluster then it could not be included in another cluster. On the contrary the second type, the **Overlapping Clustering** (see C-means on appendix A.3), uses fuzzy sets to cluster data, so that each point may belong to two or more clusters with different degrees of membership. In this case, data will be associated to an appropriate membership value.

Regarding the **Steps Performed** in the process, data clustering algorithms can be **Hierarchical** (see section A.1 on the appendices) or **Partitional**. Hierarchical algorithms find successive clusters using previously established clusters, whereas partitional algorithms determine all clusters at once. Hierarchical algorithms can be **agglomerative** ("bottom-up") or **divisive** ("top-down") depending on the **Starting Point**. Agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters. Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters.
2.3.3 Distance Measure and Decision Taking

Distance Measure and Decision Taking are two blocks of the scheme shown in figure 2.2 but, as we previously pointed out, they are often difficult to separate.

Distance Measure, keeping in mind figure 2.2, gives an objective empirical result for the similarity between the incoming data and the preexisting labeled data (in the Data Base). This measure follows a particular distance (Euclidean distance\(^{10}\), Manhattan distance\(^{11}\), Mahalanobis distance\(^{12}\), etc) selected regarding the particular application. Although there is an extensive literature on distance measures, a closer look to it is out of the scope of this section. The interested reader can easily consult bibliography on distances in mathematical related bibliography (interesting information might be found in http://mason.gmu.edu/~montecin/disedbiblio.htm).

Decision Taking is tightly related to Distance Measure. Usually the decision taker chooses the group of the unknown sample as the group of the closer labeled sample (\(k\)-nearest neighbor, using a certain distance). Some algorithms measure the distance using statistical and probabilistic means, this means that for a certain incoming datum there is a certain belonging probability to each of the possible groups\(^{13}\). This approach allows further statistical processing, using for example Bayes Theory (see section 2.4 on page 28), instead of simple maximum likelihood.

What is relevant to point out is that there are several combinations for Distance Measure and Decision Taking. All those alternatives lead to different results, so this two blocks have to be carefully chosen taking into account a wide range of factors. The nature of the data, the relationship between data samples, the representation in the features space, the interaction between blocks and the statistical approach we want to take are only some of the factors to regard when choosing or designing a distance measure and a decision maker.

\(^{10}\)The straight line distance between two points.
\(^{11}\)The distance between two points measured along axes at right angles.
\(^{12}\)Mahalanobis distance is based on correlations between variables by which different patterns can be identified and analysed. It is a useful way of determining similarity of an unknown sample set to a known one.
\(^{13}\)The sum of all the probabilistic must be then 1.
2.4 Post-Processing: Viterbi’s Algorithm

Overlapping clustering algorithms that work with temporal sequences have a great advantage: their formulation can lean on fields such as statistics theory and communication theory to solve some of their main problems. In the particular case of this Project the communications background will help solving more likely paths, which means applying the Viterbi Algorithm.

The Viterbi Algorithm [62] is a dynamic programming algorithm for finding the most likely sequence of hidden states, called the Viterbi path, that results in a sequence of observed events, especially in the context of hidden Markov models [50]. The forward algorithm is a closely related algorithm for computing the probability of a sequence of observed events. These algorithms form a subset of information theory.

Originally conceived as an error-correction scheme for noisy digital communication links, the algorithm has universal applications in decoding the convolutional codes used in both CDMA and GSM digital cellular, dial-up modems, satellite, deep-space communications, and wireless Local Area Networks. It is now also commonly used in speech recognition, keyword spotting, computational linguistics, and bioinformatics.

The Viterbi Algorithm is based upon three basic assumptions, all them valid in the system described in this Project. Those three assumptions are the following:

- **Viterbi Algorithm operates on a finite state machine assumption.** At any time the system we are modeling is in a certain state, out of a finite number of states. Multiple sequences of states (paths) can lead to a given state, but one is the most likely path to that state.

- **Transition from a previous state to a new state is marked by an incremental metric,** usually a number. This transition is computed from the event.

- **Events are cumulative over an additive path path.**

Those three features are accomplished by the output of the system so it is possible to perform post-processing based on Viterbi Algorithm. Section 3.6 (page 50) will show the
practical application to this Project and implementation of the Viterbi Algorithm.
Chapter 3

Implementation

Achieving an objective in engineering requires several steps and multiple iterations in each step. Taking a look to the big picture before even trying to attempt the problem is usually a good practice. This chapter will explain each of the steps, modules, blocks and relationship among the elements playing any role in the system but, first and foremost, an explanation of the proposed architecture for this project’s system is mandatory.

The architecture proposed for an Activity Recognition system based on clustering is structured under different criteria. Keeping the layer model\(^1\) in mind, a first criteria is based on the level of the information being treated. This means going from raw data (low-level sensor specific data) to user interface understandable data.

However, each level is fragmented in either several heterogenous blocks that cover the same function but under different standards (lower levels) or a group of collaborating blocs with different roles (higher ones).

Anyway, figure 3.1 shows the architecture proposed and the different blocs and layers that will be detailed in this chapter. This figure will be the reference during all the chapter.

Some of the components have been specifically developed for this Project (those which are going to be thoroughly detailed) and others are commercial available products (which are going to be cited but not detailed).

The first two lower layers of the scheme are explained in their respective sections. After

\(^1\)For example the ISO OSI layer model.
that, each of the blocks in the Signal Processing Layer is justified in its own section, because the function developed by each of them is complementary. The clustering section receives a particular review as it is one of the novelties reported in this Project.

3.1 Data Retrieval

As shown in figure 3.1, the Data Retrieval layer is sensor specific. The following table shows the different software solutions used for each of the sensors.

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Data Retrieval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Body Sensor</td>
<td>BodyMedia InnerView Software</td>
</tr>
<tr>
<td>Video and Audio</td>
<td>CaptureTest</td>
</tr>
<tr>
<td>Computer Monitoring</td>
<td>SoftActivity Activity Monitor</td>
</tr>
<tr>
<td>GPS</td>
<td>EasyGPS</td>
</tr>
</tbody>
</table>

Table 3.1: Software Required to Retrieve Data from each Sensor.
All the software except from the Video and Audio capture, is commercially available.

The Body Sensor data retrieval software (BodyMedia Innerview) is provided with the armband. Its use and installation are detailed in the web help developed for this Project (see section 4.2 on page 60) and in the datasheet of the sensor (some of the details are available online at www.bodymedia.com).

The computer monitoring software itself is able to retrieve the data collected in excel format. For further details on the program (Activity Monitor) check its website at www.softactivity.com.

EasyGPS is a free software (www.easygps.com) that allows data retrieving from the positioning device.

Finally the only software developed for data retrieval is the CaptureTest application which is detailed in the following subsection.

### 3.1.1 Camera Capturer

The audiovisual information gathered with the camera could be either stored or processed in real time (feature extraction). As this Project proposes a post-processing scheme it seems logical to record the information and process it \textit{a posteriori}. Recording allows, at the same time, to compare several features and properties in order to take as much information as possible and to provide easier data to inspect for the end-user (see data inspector in section 4.1 on page 54).

Even though capturing images from the camera can be done by many commercial programs (usually web cameras come with a recording software together with the drivers) the synchronization problem let us to build our own Camera Capturer software.

The \textit{CaptureTest} is a simple user interface (see figure 3.2) programmed using C# under the Visual Studio .NET Framework. Although it was initially planned to be programmed using OpenCV, the software is based on ActiveX\textsuperscript{2}.

The particularity of the program is that the user inputs the name of the folder where

\footnote{OpenCV does not allow to record audio and video in the same file, it is only a Computer Vision Library, not Sound.}
the files will be recorded but the files are named with a precise time reference. This will ease the synchronization process.

Additionally the software automatically detects all the recording parameters (for audio and video) available in a particular terminal in running time, allowing the user to select the sources, codecs and recording parameters (see detail of the menus in figure 3.2). This was specially useful during the testing phase of the webcam, in order to choose the most suitable parameters (resolution, codecs, number of bits, etc), afterwards the parameters used were the ones in table 2.1.

### 3.2 Feature Extraction

Once the data is retrieved from the sensors a feature extraction layer is in charge of extracting information from the data. This is because the algorithms implemented must have as input vectors of features and not the whole raw data.

Figure 3.3 shows the role of the Feature Extraction Layer in the whole architecture diagram (see figure 3.1), between the Data Retrieval level and the Signal Processing one.

This stage of the system required specific code implementation for all the sensors, as the features are chosen for each of the particular cases. Table 3.2 shows the tools used for feature extraction for each of the devices.
Both the Body Sensor and the Computer Monitoring software require a translation of data from Microsoft Excel format to Matlab understandable format. This conversion is performed with the `XLSREAD` function of Matlab that requires Microsoft Excel to be installed in the PC to work properly.

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Feature Extraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Body Sensor</td>
<td>MS Excel interface for Matlab</td>
</tr>
<tr>
<td>Video and Audio</td>
<td><strong>VirtualDub Scripting</strong> + Matlab interface for OpenCV</td>
</tr>
<tr>
<td>Computer Monitoring</td>
<td>MS Excel interface for Matlab</td>
</tr>
<tr>
<td>GPS</td>
<td>Matlab GPX parsing function</td>
</tr>
</tbody>
</table>

Table 3.2: Software Required for Feature Extraction

As mentioned in chapter 2.2, the features extracted from the Body Sensor are the high level information (sleeping, physical activity, lying and standing up) as well as features from the continuous, low-level channels (such as peaks in the GSR and mean values over time).

On the contrary and as well mentioned in chapter 2.2, the computer monitor information is a stream containing the software activity performed in a particular instant (work, non-work or internet surfing), and the number of keystrokes per unit of time.

The information coming from the GPX needs two steps for feature extraction. The first of them is parsing the information coming from the GPX file into Matlab (which is done using a function for parsing developed specifically for this Project). The second one is to build the Myplaces proximity vectors. As mentioned in section 2.2.4 (page 22), *MyPlaces*
are user-defined locations. The distances to this user-defined places are used as features for the algorithm.

Finally audio and video processing are the most resources and time consuming sources of information. The features extracted from both of them are detailed in section 2.2.2 (page 18). Anyway for practical reasons two codes were developed to speed and ease the audio and video processing: VirtualDub Scripting and OpenCV interface for Matlab. Those resources are explained in the following two subsections.

### 3.2.1 VirtualDub Scripting

As it is mentioned in the previous section, the video information is recorded on disk which implies that a huge amount of information is contained in a single video file. Processing such an enormous file is time consuming and sometimes the operating system is not able to handle this kind of files.

Some test using Matlab scripts to process one of each 10 frames in a 15fps 6 hours video took more than 2 days processing\(^3\). This fact reveals that indexing frames in enormous videos introduces a considerable delay.

Assuming that the information must be recorded on disk the alternative proposed to reduce the processing time is making chunks of the big videofiles. Small chunks (30 min) allow the operating system to handle videos easily and faster and allows Matlab to access the audiovisual information in a reasonable time.

The problem is then imposing that the pre-processing plus the processing time of the chunks is less than the processing time of the large videofile.

The video slicing is performed using VirtualDub\(^4\), which is a video capture/processing utility for 32-bit Windows platforms (95/98/ME/NT4/2000/XP), licensed under the GNU General Public License (GPL). A definitely interesting part of this software (a part of the GNU GPL that lets us use it without purchasing) is that it can work using scripts. The VirtualDub Scripting Language, which reference and details on members and functions

---

\(^3\)In a 1Gb RAM and 2GHz Intel Celeron powered Personal Computer with plenty of hard disk space.  
\(^4\)http://www.virtualdub.org/
3.2. Feature Extraction

can be found at http://www.virtualdub.org/docs/vdscript.txt, allows invocation from Matlab. This eases the process taking advantage of the versatility of Matlab and the high performance of a low-level programmed video application.

The pros of using this shortcut outweigh its cons. Nonetheless it requires a little bit more of programming (handling counters for synchro in several video files and multfile support for all the routines), the improvement in the performance, in terms of processing time, are impressive. The processing time is reduced between a 80 and a 90 %, which speeds-up the overall feature extraction process.

3.2.2 Matlab interface for Intel OpenCV

OpenCV is an open source computer vision library originally developed by Intel. The library is cross-platform, and runs on both Windows and Linux. It focuses mainly towards real-time image processing. Mainly programmed in C/C++, implements several image processing algorithms optimizing them for Intel processors.

The interest of OpenCV in this Project is its object detector, an implementation of [61]. The library comes with a previously trained XML file for face detection. The accuracy of this library is high for pictures but depending on the training file (frontal faces, upper bodies and other training sets) and several other parameters the detection rate achieved varies. In figure 3.4 we can see the result of applying OpenCV’s face detector to a picture.

As the library is implemented in low-level language we are enforced to code a C++ interface in order to build a Dynamic Linked Library (DLL) able to be accessed from Matlab. This task is performed using the Visual Studio .NET software pack and the build-
in Matlab compiler.

3.3 Synchronization

Figure 3.1 shows an interaction between the synchronization block and both the Data Visualization in the User Interface and further Signal Processing Blocks.

In fact the synchronization information is used to get the correct data to visualize, because the synchro means alignment and therefore coherence among data. On the other hand, as will be pointed out throughout this section, further processing makes no sense without coherent alignment of the information.

Synchronizing the data is a crucial in Multimodal Data Summarization. Different sample rates, different references and different starting times are only some of the causes for the data to be unsynchronized. The very first step for activity recognition is to make sure all the samples of the several sensors correspond to the same instant of time, afterwards those synchronized samples are processed or displayed, but this first step is mandatory.

3.3.1 Sampling Rates

One of the problems in Multimodal Data acquisition is dealing with several sampling rates, some of them constant and some of them variable. We propose simple solutions for these problems.

Subsampling (less samples than the desired sample rate) is solved holding the previous sample till a new one comes. By contrast oversampling (higher sample rate than the desired one) is solved using sets of histograms with a window of the sample period’s length, taking the most common value (mode). Variable sampling rates are solved with a combination of the solutions for subsampling and oversampling.

Typically audiovisual information (image and sound) suffer oversampling, while the body sensor and the activity monitor are cases of subsampling. The GPS data, as has only information every time it receives signal from the satellite, needs a combination of the two sampling solutions.
3.3.2 Time Reference

On the other hand different time references are also a problem for multimodal data. The GPS receiver has the satellite time reference, the computer its own one and the Body Sensor a third one. To solve this problem a synchronization process is proposed.

Using a free software tool for synchronization of the computer to an atomic clock make the satellite and computer time references the same. Additionally the body sensor software allows synchronization with the PC’s clock.

We use SP TimeSync (http://www.spdialer.com/timesync), which lets you synchronize your computer’s clock with any Internet atomic clock (time server), using Network Time Protocol. Then the body sensor gets the time from the PC and a unique time reference is set\(^5\).

After several days of data collection it was observed that the armband at the end of the day was about 3 seconds delayed from the PC’s time, but we considered this delay negligible in front of about 7 hours of data collection.

3.3.3 Criteria for Data Selection

As the different sensors do not start collecting data at the exact same time (GPS and body sensor may start later than the webcam) the valid start and end reference times are chosen to be the more restrictive (less amount of samples per day). This means that the whole day will start whenever the last sensor starts sampling and will end when the first sensor stops working\(^6\).

3.4 Our Clustering Approach

Given the set of multimodal features extracted in previous sections, our goal is to segment the data into temporally coherent chunks. In this section we extend standard clustering

\(^5\)Although this method is valid and useful is a little bit tough to follow, future versions of this Project should implement an autosync process.

\(^6\)If we only have two sensors and one of those starts sampling at 8 o’clock and ends at 6 o’clock and another one starts at 9 o’clock and ends at 7, the valid time interval will be from 9 to 6 o’clock.
algorithms (e.g. K-means, see appendix A.2 or spectral graph methods) to incorporate temporal coherence and semi-supervised information. The result is what we call **Semi-Supervised Spatio-Temporal Clustering** (SSTC) (see published paper on appendix ??).

### 3.4.1 Notation

Bold capital letters denote a matrix $D$, bold lower-case letters a column vector $d$. $d_j$ represents the $j$ column of the matrix $D$. $d_{ij}$ denotes the scalar in the row $i$ and column $j$ of the matrix $D$ and the scalar $i$-th element of a column vector $d_j$. $d_{ji}$ is the $i$-th scalar element of the vector $d_j$. All non-bold letters will represent variables of scalar nature. $\text{diag}$ is an operator which transforms a vector to a diagonal matrix or takes the diagonal of the matrix into a vector. $\mathbf{1}_k \in \mathbb{R}^{k \times 1}$ is a vector of ones. $\mathbf{I}_k \in \mathbb{R}^{k \times k}$ is the identity matrix. $\text{tr}(A) = \sum a_{ii}$ is the trace of the matrix $A$ and $|A|$ denotes the determinant. $||A||_F = tr(A^T A) = tr(\mathbf{A A}^T)$ designates the Frobenious norm of a matrix. $\circ$ denotes the entrywise or Hadamard matrix product.

### 3.4.2 Objective Function

Discriminative Cluster Analysis (DCA) [32] is a discriminative clustering method that combines both clustering and dimensionality reduction in a unsupervised manner. DCA is based on a normalization of the energy function used in Linear Discriminant Analysis [23] conveniently expressed in matrix form:

$$E_{DCA}(B, V, G) = ||(G^T G)^{-\frac{1}{2}}(G^T - VB^T D)||_F$$

(3.1)

where $c$ denotes the number of classes and $n$ the number of samples. The columns of $D \in \mathbb{R}^{d \times n}$ contain the original data $d_i$ in columns. $G \in \mathbb{R}^{n \times c}$ is a dummy indicator matrix, such that $\sum_j g_{ij} = 1$, $g_{ij} \in \{0, 1\}$ and $g_{ij}$ is 1 if $d_i$ belongs to class $C_j$. $B \in \mathbb{R}^{d \times c}$ is a linear transformation of the data from the original dimension to the feature space that maximizes the distance between the means of the classes and minimizes the variance within clusters.
3.4. Our Clustering Approach

In fact, relaxing the constraints on $G$ (considering it in the continuous positive domain subject to $G1_c = 1_n$ and after eliminating $V$, [32] shows that equation 3.1 is proportional to:

$$E_{DCA} = tr(C^T(C^T)^{-1}CG^TG^{-1}G^T)$$  \hspace{1cm} (3.2)

Where $C = B^TD$. Using a gradient descent combined with a line search strategy to search for an optimum allows DCA to cluster computationally efficiently and effectively. In our case, we consider $B = I$, and $C^T(C^T)^{-1}C = D^T(DD^T)^{-1}D$ that provides additional normalization (i.e. k-means cluster the matrix $D^TD$).

3.4.3 Spatio-temporal clustering

In previous section, we reviewed previous work on k-means, discriminative cluster analysis and spectral graph methods in a novel matrix formulation. However, prior work does not consider any temporal coherence among the data. This means that all the algorithms described cluster each single data point separately, whereas real multimodal data requires temporal consistency to avoid undesired behaviors (e.g. glitches).

One of the benefits of relating the clustering problem to an objective function is that we can easily incorporate spatio-temporal coherence in a straightforward manner by adding a penalty term.

The objective is avoiding unnecessary transitions. It is desirable that the clustering of a single sample is similar to its neighbors in time. Mathematically we want $g_i$ and $g_{i+1}$ to have similar values$^7$, so we want the following expression to be minimum:

$$E_{tbasic} = \sum_{i=1}^{n-1} \|g_i - g_{i+1}\|_2^2 = \|G^T - G^T\|_F$$  \hspace{1cm} (3.3)

$^7$Values on $g_i$ might be seen as proximity indicators of the sample $i$ to each cluster.
Where $P$ is a permutation matrix:

$$
P = \begin{pmatrix}
0 & 0 & \ldots & 0 & 1 \\
1 & 0 & \ldots & 0 & 0 \\
0 & 1 & \ldots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \ldots & 1 & 0
\end{pmatrix}
$$

(3.4)

This formulation has two drawbacks. The first of them is that minimizing this term means minimizing the number of transitions (if the objective function was this only term all the samples would be classified in the same cluster) and that could harm the correct clustering. This problem can be solved learning the parameter that makes this term comparable to the main cluster objective function parameter. This means that whenever the temporal term tries to force all the samples to a single group the clustering objective function (k-means, LDA, etc) would rise its value retrieving that specific classification to be the minimum value for the overall function.

The second drawback is that the permutation matrix, although allows an easy and elegant matrix formulation, enforces a relationship between the first and last samples which, in general, is not true\(^8\). However, considering a sampling rate of one second makes more than 10,000 samples to cluster, which makes the relationship between the first and last samples negligible in front of the weight of the rest of terms.

In order to take also in account the dynamics of the states we reformulate equation 3.3 as:

$$
E_t = \left\| \left( G^T G \right)^{-\frac{1}{2}} \left( G^T - AG^T P \right) \right\|_F
$$

(3.5)

$A$ encodes the states dynamics in terms of transition among states and the matrix $G^T G$

\(^8\)If the sampling process lasted a cyclic 24h period that would make sense (if the user was doing the same activity every time he stops collecting data). This approach though, considers data collection in the range between 5 and 8 hours.
is a normalization weight matrix.

Then the spatio-temporal clustering algorithm will combine results from the previous equations 3.2 and 3.5 as follows:

\[ E_{st} = E_{\text{cluster}} + \lambda E_t \] (3.6)

where the cluster method can be \text{k-means}, DCA or spectral clustering. The parameter \( \lambda \) is a normalization factor that makes the two terms comparable and can be learned from training data (e.g. cross-validation).

The additive trends of the expression combined with gradient’s linear properties makes it easy to reformulate the gradient descent algorithm. To impose non-negativity constraints in \( g_{ij} \), we parametrize \( G \) as the product of two matrices \( G = V \circ V \) (as described in [37], and used in [32]). Then the \( V \) updating equation is:

\[ V^{n+1} = V^n - \eta \left( \frac{\partial E_{\text{cluster}}(V^n)}{\partial V} + \lambda \frac{\partial E_t(V^n)}{\partial V} \right) \] (3.7)

The dynamic \( A \) matrix will be updated in every iteration as follows:

\[ A = G^T P^T G (G^T G)^{-1} \] (3.8)

enforcing the similarity between two contiguous terms over time.

Developing the Frobenius norm and taking derives from equation 3.8 we can conclude that:

\[ \frac{\partial E_t(V^n)}{\partial V} = V \circ (2GA^T(G^T G)^{-1}A - 2G(G^T G)^{-1}AG^T GA^T(G^T G)^{-1} \ldots + 4G(G^T G)^{-1}GPAG^T G(G^T G)^{-1} - 2P^T G(A^T(G^T G)^{-1} - 2PG(G^T G)^{-1}A) \] (3.9)

The initialization of the algorithm is based only on the clustering term (DCA, LDA or K-means). Whenever the objective function converges to a reasonable value (e.g the 10% of the initial value), which means a preliminary clustering, the temporal term starts getting
relevance towards smoothing the output.

3.4.4 Adding semi-supervised information

In this section we extend previous clustering technique to incorporate side supervised information. Assuming pairwise constraints we formulate two kind of additive terms related to supervised information depending on the relationship between the pairs of points: the must-link term and the cannot-link term.

Similarly to the temporal term, the supervised information will be encoded as additive (penalty) or subtractive (reward) to the objective function as done in [31]. This means that for the must-link term we are favoring the pairwise labeled data to be similarly classified in terms of cluster belonging probability. The cannot-link, on the other hand, punishes similar classifications for data samples known to belong to different clusters.

Let \( \mathbf{d}_p, \mathbf{d}_q \in \mathbb{N}^s \) be two samples that belong to the same class (activity). \( \mathbb{N}^s \) is the set of must-link supervised information samples and \( \mathbf{e}_r \in \mathbb{R}^n \) is an indicator vector for data point \( \mathbf{d}_r \) so that \( \mathbf{D} \mathbf{e}_r = \mathbf{d}_r \). We formulate the must-link supervised additive term as follows.

\[
E_{sML} = \sum_{i,j \in \mathbb{N}^s} ||\mathbf{g}_i - \mathbf{g}_j|| = \sum_{i,j \in \mathbb{N}^s} ||\mathbf{G}^T \mathbf{e}_i - \mathbf{G}^T \mathbf{e}_j|| = ||\mathbf{G}^T \mathbf{E}_{ML}||_F
\]  

(3.10)

Where \( \mathbf{E}_{ML} \in \mathbb{R}^{n \times l} \) is a matrix with \( l \) columns corresponding to the number of pairs of data points that belong together and each column contains the vector \( \mathbf{e}_p - \mathbf{e}_q \) that defines a pair of must-link points.

On the other hand, being \( \mathbb{N}^D \) the set of cannot-link supervised pairs, the formulation for the points that do not belong together is the following:

\[
E_{sCN} = \sum_{i,j \in \mathbb{N}^D} ||\mathbf{g}_i - \mathbf{g}_j|| = \sum_{i,j \in \mathbb{N}^D} ||\mathbf{G}^T \mathbf{e}_i - \mathbf{G}^T \mathbf{e}_j|| = ||\mathbf{G}^T \mathbf{E}_{CN}||_F
\]  

(3.11)

Where \( \mathbf{E}_{CL} \in \mathbb{R}^{n \times l} \) is now a matrix with \( m \) columns corresponding to the number of
pairs of data points that do not belong together and each column contains the vector $e_p - e_q$ that defines a pair of cannot-link points.

The total objective function then remains:

$$E_{sm,t} = E_{	ext{cluster}} + \lambda E_t + \beta_1 E_{sML} - \beta_2 E_{sCL}$$

(3.12)

Changes in the objective function are reflected in the $V$ updating equation as follows:

$$V^{n+1} = V^n - \eta \left( \frac{\partial E_{\text{cluster}}(V^n)}{\partial V} + \lambda \frac{\partial E_t(V^n)}{\partial V} + \beta_1 \frac{\partial E_{sML}(V^n)}{\partial V} + \beta_2 \frac{\partial E_{sCL}(V^n)}{\partial V} \right)$$

(3.13)

Developing equations 3.10 and 3.11 and taking derives in order to perform a gradient descent algorithm we get to the expression:

$$\frac{\partial T_{s,i}(V^n)}{\partial V} = V \odot (2E_i E_i^T G)$$

(3.14)

being $i$ 1 for the must-link and 2 for the cannot-link.

Unlike the temporal term, the semi-supervised term is imposed from the first iteration of the algorithm together with the main objective function.

It is really interesting in terms of processing time and resources to take profit of the sparse matrix formulation of the problem, something easily done with Matlab c⃝.

3.4.5 Initialization

Clustering huge amounts of data is always hard and needs clever strategies to face the problem correctly and efficiently. As we reviewed in the spatiotemporal clustering section, bottom up attempts usually make no sense (each sample is classified on its own) and there is need to apply smoothing in order to get a more realistic clustering.

In fact a top-down approach seems, at first glance, an easier and more natural strategy.
A given set of samples contiguous in time are less relevant (in terms of information) than the same number of samples homogenously distributed over the whole time series to cluster.

Following the top down idea we propose a Multiresolution scheme for clustering (see figure 3.5). This means that we first coarse the data (applying decimation) and run the algorithm described in previous sections. The decimate output is redefined (expanded by repeating samples) and is the initialization for the next step. This loop is repeated till the desired resolution of the output is reached.

This working scheme has two main benefits. The first benefit is that the influence of the initialization with the previous clustering eases the minimization of objective function (speeds the whole process). Secondly this strategy allows the temporal term to affect not only to contiguous samples but till the decimation factor, bettering the smoothing.

In our particular case the decimation steps are powers of 2, using typically 7 steps, coarsening for maximum factor of 128 and redefining back till the desired output resolution (up to one second or less).

### 3.4.6 Synthetic Example: Temporal Term

To show the proper work of the temporal term we propose a simple toy problem clustering example. The aim of this toy problem is to illustrate that a certain clustering algorithm
3.4. Our Clustering Approach

Figure 3.6: Toy Data and 2-cluster classification for K-means, DCA and DCA with temporal coherence.

should be chosen depending on the kind of data to cluster.

Let \( S \) be a signal that takes only two discrete values \( \{5, 10\} \). This signal is contaminated with a random gaussian additive noise \( N \) with zero mean and a standard deviation of 0.3. With this data K-means is supposed to behave correctly for clustering purposes, because it is a simple 2-cluster, 1-dimension gaussian centered problem. In those conditions K-means can be proved to be optimal.

If we introduce impulsive noise into the system (glitches of wrong data in a random number of samples) the signal would not be correctly clusterized using K-means. It seems obvious that in this case K-means is not optimum any more because it is going to cluster each sample separately, not taking into account the environment, so the samples with impulsive noise will be misclusterized. We can see this in figure 3.6.a.

LDA (figure 3.6.b) is not going to perform good either for the same reason as K-means, again independence is assumed for each data point.

In this case it seems obvious that knowing the nature of the data we can impose temporal restrictions in order to smooth the classification. This way we can see in figure 3.6.c that
the additive temporal term gives a better classification for the data, in terms of higher level information.

This kind of noise can be found in Multimodal Diaries, because the information from the camera and the audio sources is sometimes fastly changing and therefore generating glitches.

3.4.7 Synthetic Example: Semi-Supervised information

To show the proper work of the Semi-Supervised clustering term we propose another simple toy problem clustering example.

Let $D \in \mathbb{R}^{2 \times n}$ be a 2-dimensional gaussian distributed Data that belongs to five different groups. Among these groups three of them are close to each other and the other two are away (see the figure in 3.7). Running any clustering algorithm described in previous sections looking for 3 clusters it is easy to predict that the three groups closer to each other will be in a cluster and the two other will constitute the other two clusters.

We would like to know if any supervised information would change the clustering so we impose that one of the groups in the three-group mentioned before actually belongs to the same cluster as one of the outer clusters. Using the terms in previous sections we impose a must-link pairwise constraint.

Imposing only the supervised term to the objective function of the clustering method gives us unexpected results: only the samples labeled change clusters outputting a really bad clustering. In this case it is interesting to count on the spatio-temporal additive
term additionally because it gives a smoothing trend to the algorithm that eases clustering changes.

The result is shown in the right image of figure 3.7, where we can see that although the proximity of the three groups pushes them to be in the same cluster, imposing the must-link the classification has been modified as we desired.

### 3.5 Similarity Measure

Distance Measuring in the approach proposed in this project is an intrinsical procedure. In fact as we stated in page 40, we consider the relaxation of the constraints in $G$ in equation 3.1, which means that $G$ will take real values instead of binary ones. Keeping in mind that $G$ has dimensions of number of samples per number of clusters, we can treat this output of the algorithm as a certain distance of the datum to the cluster.

In fact, taking a step further, considering $G1_c = 1_n$ we are imposing provability conditions to the matrix. This means that in some sense we are outputting a vector with likelihood that a certain sample $d_i$ belongs to class $C_j$.

From this assumptions we can process the output using several methods for decision taking. The first and simplistic one would be decide that the sample $d_i$ belongs to the group $C_n$ which belonging probability is higher, which would mean:

$$\text{Group}(i) = C_n = Max_j(g_{ij})$$  \hspace{1cm} (3.15)

From this expression on, diverse procedures could be used to take a decision for each sample. Our approach for decision taking is based upon several samples (not only one) and taking advantage of provability formulations previously explained in section 2.4 (on page 2.4). In the next section its practical application and implementation are detailed.

As we stated in the previous chapter, separating distance measuring and decision taking is not usually trivial. This project it is not an exception.
3.6 Viterbi Algorithm Based Decision

The similarity of the information contained in the $G$ matrix to the incoming symbols in a multisymbol communications scheme made the authors use Viterbi Algorithm [62] for more likely path identification. A brief introduction to the algorithm and its conditions are related in section 2.4, on page 28.

In order to perform the Viterbi Algorithm on the data we need two more pieces of information, apart from the $G$ matrix (containing the cluster belonging probabilities). Those two missing pieces are the probabilities for the first sample and the transition matrix.

The first piece, the initial probabilities, can take two values. A first attempt could be assuming equiprobability (the same probability that the first sample belongs to each of the groups). The second one takes advantage of the labeling application and assumes that the first sample is labeled by the user under supervision, which directly classifies it.

Both attempts have been tested and behave similarly but the second one, using supervised information, was the finally adopted one.

On the other hand the transition matrix is built from the labeled data (see section 5.1 for labeling procedures). The information contained in the transition matrix is the probability of transition from a certain state to any other. In a general case it is not a symmetrical matrix, which means that going from a given state to another is not the same as going from the second to the first one$^{10}$.

In order to train the transition matrix we used the labeling information from 3 days and counted transitions among states. After normalization and exceptional case handling$^{11}$ the matrix was fixed. This matrix would then be used for further calculations in all the days (not only the ones used for training).

The use of Viterbi Algorithm shows to fix deeper relationships among data than the ones

---

$^{10}$It seems evident that the probability of transition from Sleeping to Breakfast is not the same as the opposite way.

$^{11}$Zeroes in the in the Matrix lead to runtime errors, so were substituted by half the minimum value of the whole matrix. Similarly the same state transition, this is the number of times a state transitioned to the same one (values of the diagonal of the transition matrix), were also altered. As normalization resulted in almost identity matrices, the values of the diagonal matrix were fixed to the double of the greatest value in the rest of the matrix. Afterwards the matrix was normalized. This criteria, although it is heuristical, performed good.)
established with the temporal term of the clustering algorithm, resulting in an improvement of the overall system. Moreover it is a novelty to apply Viterbi Algorithm in Overlapping clustering as a postprocessing block.

The actual implementation of the algorithm we used was a function from the popular Hidden Markov Model Toolbox for Matlab\textsuperscript{12} written by Kevin Murphy. Concretely the viterbi_path function, which parameters are the initial probabilities, the $G$ matrix and the transition probability matrix.

\textsuperscript{12}Downloadable for free from http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html
Chapter 4

User Interface

The User Interface is a really important component in a Human-Computer Interaction applications. It does not make sense to make several steps for activity recognition if no front-end is placed between what we can call the *machine world* and the *human* one. This chapter covers all the aspects of the User Interface in this Project.

In our particular case, the Graphical User Interface (GUI) designed and implemented, covers different Human-Computer Interaction levels. In fact the interface covers two main purposes:

- Labeling
- Visualization

The labeling purposes are the link between the user annotated data (raw data) and the Signal Processing block, as shown in figure 4.1. It is simply a friendly way to prompt labeled data to the user.

In contrast the visualization purposes refer to the very end of the process. The User Interface covers two main visualization utilities: synchronized data visualization and results visualization. Both utilities are detailed in this chapter in their respective sections.

The first section of this chapter will cover what *strictly* is the User Interface (functions described in the text above).
Additionally and in order make an exportable system, a help on the User Interface and the installation process of the whole system is provided. The second section of this chapter covers this help documentation.

## 4.1 Data Inspector

**Data Inspector** is a MATLAB© Graphical User Interface that allows data collection and visualization from different sensors for training and clustering purposes.

A general view of the User Interface of the Data Inspector will easily reveal that the information is organized in groups of panels. Each of the panels contains the information of one of the sensors or a control of a particular function.

Apart from the panels, Data Inspector has a Menu, at the top of the window, that allows the user to take advanced actions, such as outputting information and detail inspection.

An overview of the whole User Interface is shown in figure 4.2 and the legend for this figure is located in table 4.1.

Its main functions (Data Visualization, Labeling Tool and Results Visualization) are detailed in the following subsections.
4.1 Data Inspector

4.1.1 Data Visualization

Visualizing the Data is the original and main goal of Data Inspector. This utility allows the user to move along all the gathered data and view a particular instant of the day.

As it can be seen in figures 4.1 and 3.1, the module for Data Visualization takes information from two main sources, the synchronizing system (see section 3.3 on page 38) where the time reference is taken and the data retrieved (see section 3.1 on page 32) that contains
the original gathered data.

Using those two functional blocks, the User Interface shows 5 information panels and a control one (view figure 4.2 and table 4.1).

- **Body Data Panel**: Composed by a chart, a listbox and a textbox, this panel shows the synchronized data collected from the Physical Awareness Body Sensor. The list on the right allows the user to choose the variable to inspect in the chart. One the target is chosen, the chart displays the data with a vertical red line indicating the instant the Timer Control is indicating. At the same time the value (or values\(^1\)) of the particular sample of the variable is (are) displayed in the textbox. An additional button allows the user to clear the chart whenever he (or she) desires to.

- **GPS Panel**: Three radio buttons and two buttons, together with a chart displaying a situation map\(^2\), let the user check the synchronized data from the GPS logger. The three radio buttons allow the user to enable the whole panel, display (in green) in the chart the last 4 locations and display MyPlaces (see section 2.2.4 on page 22) locations with their radius of action respectively. The two buttons allow playing an animation of the whole day track and to configure MyPlaces.

- **Personal Camera Panel**: The synchronized video is shown in this panel, allowing the user to enable OpenCV’s Face detector (see section 3.2.2 on page 37). A part from that possibilities, the panel has got an enable/disable radio button and a Play video, that shows the last minute of video in the chart\(^3\).

- **Sound Panel**: Similarly to the previous case, the Sound Panel shows the last 60 seconds of recorded sound in the instant indicated by the Timer Control. The same

---

\(^1\) Acceleration has two axis, so both components are displayed at the same time if acceleration is chosen in the listbox.

\(^2\) The situation Map is introduced using a georeferenced bitmap image. Although the georeferencing process is implemented using scripts (non user interface code), this process could be added to the GUI’s configurability options. The maps used for this project are taken from Google Maps\(\copyright\) for research purposes only. Any kind of scaled map or georeferenced satellite picture could be used instead.

\(^3\) As the video files this system is dealing with are so extensive, playing a video in Matlab may be time consuming. This is because the implementation of this button is a lecture frame by frame from the file and display of the resulting frame. The Video played will not have any sound.
4.1. Data Inspector

enabling radio button is offered for the panel and the playing button is completely analogous to the Video one.

- **Programs in Use Panel**: This panel only displays the programs being executed in the instant indicated by the Timer Control.

- **Timer Control**: Controlling the instant of time displayed and tightly related to the synchronization tool, the timer control is able to move forward and backwards in the all-day-long information gathered, using a slider bar. Three status text boxes display the time of the first and last samples and the time of the current sample displayed. Another text box displays the percent of the day the program is displaying. This text box also allows inverse functionality (introducing a percentage the timer control will go to the instant of that particular instant, after pressing the ‘GO’ button).

Together with the mentioned information available in the multiple panels of the User Interface, the Menus offer advanced options for the user. Only three of the advanced options will be mentioned in this chapter because detailing all the menus is out of the scope of this chapter and the writer forwards the interested readers to the web help or to the code itself (both attached in the CD submitted with this document). The advanced options detailed are the Global View Map, the KML output option and the video recording utility.

The GPS information allows really interesting treatment and visualization possibilities. The visualization option chosen for Data Inspector is showing in each time sample a zoom (about 500 x 300 metres) on the overall map. The ‘Global View Map’ option, selectable from the Menu, opens a new window with the entire georeferenced map and the track followed by the user all the day drawn on it.

Similar utility has the GoogleEarth® KML Output (see figure 2.9). This option prompts the user for an output filename of a track file viewable with the mentioned software.

Finally the video recording utility allows the user to record a summary video of the whole day. This video consists of the result of changing the Timer Control in little time steps and capturing images. Those images are merged in a single video file and outputted where the user chooses to.
4.1.2 Labeling Tool

Having a unique interface for all the functionalities is always easier for the user, because a single application collects all the information he needs and all the inputs and outputs he might provide or receive. Moreover the labeling information is tightly related to the data the user can visualize with Data Inspector, something that leads us to integrate the labeling tool in the same window.

Figure 4.1 shows the labeling module of the user interface and its role in the system. This module collects the user labeled data (Raw Data coming from the user) and outputs a set of vectors that will be used in the clustering process of the Activity Recognition part (see chapter 5).

The utility of the labeling panel (figure 4.3 and number 7 in figure 4.2) is to label the activities held in a day for training purposes. The panel counts with a slider which controls the timing, an option text box that allows the user to choose among the defined activities and some other control to ease the labeling process.

To label the activities held in a day the user can move the Labeling Slider at the top of the panel (see figure 4.3). When the Labeling Slider is moved Data Inspector shows the data corresponding to that particular sample of time.

In order to label a certain activity held in a time interval the user sets the Labeling Slider to a definite position position, selects the activity held between the last marking and that instant of time from the Activity Pop-Up Menu and presses the Label Button. The
last position of the Labeling Slider will be the starting (if we slide forward) or ending (if we slide back) of the next labeling mark.

The Reset Button restarts all the labeling and the Legend Button shows the activity legend (see figure 4.3) corresponding to the activities shown in the Activity Pop-Up Menu.

To make the process easier for the user, an incremental labeling tool is provided. This allows the user to move the Labeling Slider forward or backward (if the + or − buttons are pressed) a percentage of the total number of samples introduced by the user (0.5 % of the total samples is the default value).

Whenever the user has finished labeling the daily activity, the Done Button should be pressed. That action will pop-up a dialog window requesting a filename to save the labeled data.

Finally a radio button allows the permutation between the labeling panel and the results (clustering) one.

4.1.3 Results Visualization

The results panel (clustering panel) is located in the same place as the labeling one, the user selects to see either one or the other. The unique goal of this panel is outputting the results of the whole system when the clustering and decision taking processes have been performed.

Its structure (see figure 4.4) is a chart showing the activities held in a particular day, together with a text box that indicates the activity performed in the particular time indicated by the time control. Likewise the chart offered in the body sensor channel (see section 4.1), the results panel has a vertical red line indicating the current time sample.
Other means of result visualization have been developed for this project (graph state representation, radial representation, etc) but not merged with Data Inspector. Details on these representations can be found in the CD attached with this document (Matlab® code).

4.2 Web Help

The portability of the system described in this Project is one of its weaknesses. It is evident that each one of the sensors, and this is something the user has easily noticed in chapter 5, has its own trademark software to retrieve data. This makes the whole system non-user friendly (keep in mind that this is a research project and the goal of it is testing the conditions and checking if state recognition is possible with several constraints detailed in section 5.2 on page 65) and requires a know-how that must be documented for future reproducibility in similar conditions.

On the other hand all the programs, and specially those whose targets are the general public, require guidelines for the user. This means detailing functionalities, menus, applications and specifications of the system where the program must be running.

The two functionalities described in the last paragraphs are covered with the development of a help documentation. Centered in the installation process of the system and the use of Data Inspector (see section 4.1 on page 54), the help is written in HiperText Markup Language (HTML).

Having the documentation in HTML is useful for several reasons, but the most important are:

- The information is available for all the potential users easily (from a website).
- Matlab® can easily interface a web page.

These two reasons lead us to develop the Help in this language.

The accessibility of the web-based help from Data Inspector is from one of its menus, that opens Matlab® explorer with the help site on it. This way the interaction between
those two components results in the whole User Interface designed for this Project (see figure 4.1).

The entire web help is submitted in the CD attached to this Project available online (temporary at http://www.cs.cmu.edu/~cagell/Work/html/Index.htm).
Chapter 5

Data Collection and Analysis

5.1 Practical Issues in Data Collection

Designing a multimodal acquisition system is complicated because there are several factors that can make the whole system malfunction, but once the system is definite there is a mandatory test and verification process which, in the case of this Project, was also performed by the author.

Collecting Data might seem an easy and approachable task because it is a passive action. One may think that a data acquisition system should work by itself and should not alter the activity of the user under inspection. The author’s experience collecting data and supervising both the method and the data collected states that the sentence before is everything but true.

About a month long data collection period with two users under inspection (two complete set of sensors and algorithms running in two separate computers) resulted in only 14 complete days recorded (out of 60 possible\(^1\)). Even though it was the project development process, this period is part of this project and it is worth mentioning.

The following paragraphs will relate some of the technical and practical problems suffered during the data collection period.

Wearing the sensor platform is something plain people, like the author, is not used to

\(^1\)One month for two separate people
do. In an initial phase of the data collection period the user under observation has to get used to wearing the sensors. But wearing them is not the only task he has to do, but looking after the correct data collection is a plus. Charging the batteries of the portable devices and correct retrieval of the data is also something to keep in mind.

However, the fixed sensors (camera and software inspector) can lead to other kind of problems. Different angles and positions were tested with different features in order to extract as much information as possible. Fixing the camera was one of the priorities for this sensor. Figure 5.1 shows one of the proposed locations of the camera and its particular fixing method.

Any power supply failure, either the network or the batteries of the devices, would ruin a data collection day. But this was not the only limitation, accurate and reviewable criterions forced to record the whole raw information. This means enormous amounts of memory, which easily filled any hard disk. DVD storage solutions were early taken.

Trying parameters for video recording and Body Sensor (see figure 5.1 where a user is showing the armband) sampling rate, also consumed several days.

GPS detection was one of the hardest problems to deal with. Interior detection was, of
course, impossible, but in fact sometimes more than 3 minutes were necessary for satellite location outdoors. This means lots of missing data, another problem the algorithm should have to deal with.

A part from that, whenever a day was considered to be \textit{complete} in terms of data collection, a new task, labeling, had to be performed by the user under observation. The aim of labeling was to compare \textit{real} classification with the one provided by the algorithm. Even though it might seem a trivial activity, labeling required, sometimes, about 2 or 3 hours per day\textsuperscript{2}.

All in all was a complete month of data collection, that included several naps in a park after lunch (all the states have to be tested...), something that gave the authors several hints to better the \textit{usability} of the system.

5.2 Results

Referring to the output of the system described in this document as \textit{results} may be a confusing term because the output it is not either \textit{correct} or \textit{incorrect} and it is not trivial to define a scale of correctness for it. As this is a complete prototype design, the objective of this section is to \textit{evaluate} the results achieved with it.

In this sense the engineering objective is only evaluating the project checking whether it works or not. On the other hand the scientific community always asks for a quantitative result fo

We have collected and labeled data over different days in an office scenario for two different people, a total of 14 datasets. Fig. 2.3 (on page 16) shows a typical example of all the data gathered. This multimodal data (between 5 and 9 hours per day) is classified into 9 types of activities: Sleeping, Walking, Working (no PC), Internet Surfing, Working out, Working on PC, Talking, Phone and Away (as explained back in section ).

\textsuperscript{2}Handling video and synchronized info with Matlab can be really time consuming.
5.2.1 Running the Algorithms

We used the methods described in sections 5, A.2 and in [32] to cluster MultiModal Diaries’ Data using a precision of 1 second. The data was collected and post processed using implementations of the algorithms in Matlab© in an equitable framework (same parameters and initializations for all algorithms).

As the aim of the benchmarking is to compare the clustering part in the whole system, the implementation of k-means (which clusters but does not recognize) is supported by a best matching process \(^3\). For the rest of implementations the recognizing process explained in previous chapters is performed.

5.2.2 Accuracy Measure

In order to compare several methods objectively we define a sample-level accuracy:

\(^3\)This means that once the clustering is performed, every cluster is assigned to the most similar activity using the labeled information.
5.2. Results

\[ Acc = \frac{\text{correct clustering samples}}{\text{total number of samples}} \]  \hspace{1cm} (5.1)

This accuracy measure requires correct and precise labeling information for each day (user annotated data), which is an important constraint of the whole evaluating system. Several day-long data were collected and labeled manually as accurately as possible from two different subjects under different office environments.

Although this measure is not good when there are shifts between the clustering output and the labeled data, the length of the data and its precision make this shift mistakes despicable in front of the huge amount of data processed.

5.2.3 Output Analysis

Table 5.1 shows the results (using the accuracy in equation 5.1) for all the algorithms described in section 4 with the 5 days of data collected for two people.

The results for the three first algorithms correspond to the best matching classification of the clusters found.\(^4\)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Simple</th>
<th>Multiresolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-means</td>
<td>67.77±2.1%</td>
<td>–</td>
</tr>
<tr>
<td>LDA</td>
<td>73.04±1.3%</td>
<td>75.16±4.4%</td>
</tr>
<tr>
<td>LDA+ST</td>
<td>76.13±1.2%</td>
<td>77.80±5.1%</td>
</tr>
<tr>
<td>LDA+ST+SPV</td>
<td>83.10±0.9%</td>
<td>89.21±4.7%</td>
</tr>
</tbody>
</table>

Table 5.1: Accuracy for the different clustering methods

The statistics show how the algorithm and variations proposed in this paper incrementally better the results (in terms of accuracy) of pre-existing state-of-the-art algorithms (k-means). This means that the additional terms help LDA to improve the accuracy of the overall system. Figure 5.2 shows how similar is the output of the LDA + Time + SupervisedInformation to the user-labeled information for a 6h 10min record.\(^5\)

\(^4\)This means that once clusters have been found a mapping process (based on the labeled samples) assigns a particular cluster to every activity.

\(^5\)In this figure it is evident the difference between results and evaluation. While the figure corresponding to the output of the system is really similar to the labeled user input, the accuracy is only about 80%. The
Besides statistical concerns, a closer look to the methods reveals that the best algorithm, using both spatiotemporal and supervised terms, enters into the world of classification. This means that clustering algorithms themselves only create groups whereas this algorithm identifies the groups taking advantage of the information from the supervised term. This means that the term \textit{supervised clustering} can be interpreted as \textit{supervised classification} and compared to the several algorithms of this field, something out of the scope of this project.

The number of supervised samples used by the algorithm is also a determinant factor for accuracy. As it could be easily predicted the accuracy improves increasing the number of samples. Figure 5.3 depicts this behavior. It is interesting to point out that when the number of samples is null results are the same as \textit{LDA + Time} and that the graph asymptotically tends to 100\% of accuracy (with a lot of samples it is easy to predict the whole activity).

It is also worth commenting that the greater the number of supervised samples, the better initialization the algorithm gets, which means that not only the data is more reliable

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{chart.png}
\caption{Accuracy versus number of labeled samples in the supervised term (number of supervised samples over total number of samples in permil ($\%$))}
\end{figure}
5.2. Results

but also the path to follow is more likely to be correct.

Centering the scope of this study into a particular case (the one taking a 1.1 % of the samples as supervised samples) it is interesting to get to know which states behave better than others in terms of detection. This means checking local accuracy for all states and evaluating confusion matrices.

Table 5.2 shows accuracy statistics for all the states taken into account.

<table>
<thead>
<tr>
<th>State</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sleeping</td>
<td>64.05±5.2%</td>
</tr>
<tr>
<td>Walking</td>
<td>91.02±2.1%</td>
</tr>
<tr>
<td>Working no PC</td>
<td>69.45±2.5%</td>
</tr>
<tr>
<td>Inet Surfing</td>
<td>89.78±1.4%</td>
</tr>
<tr>
<td>Working PC</td>
<td>92.10±2.7%</td>
</tr>
<tr>
<td>Talking</td>
<td>60.26±8.1%</td>
</tr>
<tr>
<td>Restroom</td>
<td>73.21±6.3%</td>
</tr>
<tr>
<td>Phone</td>
<td>81.10±7.2%</td>
</tr>
<tr>
<td>Away</td>
<td>94.12±1.3%</td>
</tr>
</tbody>
</table>

Table 5.2: Accuracy for the different states defined

Looking at the big picture the highest accuracy is achieved in the away state, there is no doubt that this is a consequence of the great performance of the video features. On the contrary the worst result, obtained in the talking state, is a result of the confusion of this state with activities happening at the same time (working or internet surfing).

It is also interesting to look at the standard deviation of the results, which denotes the confidence of each result. Usually this value decreases when the accuracy rises.

Confusion matrices were used in the design and evaluation phases of the algorithm refinement. Some of the information provided by these means was used to understand many mistakes committed and codification in Matlab is enclosed in the CD of this project.

All in all results are, anyway, slightly better that what we could achieve with the $k$-means approach, keeping in mind the structural differences involved.

Finally the objective of the overall system, detecting daily activities, is achieved when talking in evaluation terms. This means that the output of the system gives a state vector that makes sense compared to the daily activity.
Chapter 6

Summary, Conclusions and Future Work

Effective time management is key to having a successful personal and professional life. Time management is about planning time efficiently, and following that plan effectively. Although unfortunately nobody better than oneself can plan the activities to follow, Activity Recognition, as we have seen in this work, can help the user check that a plan has successfully accomplished and learn for future planning.

Making an inventory of the user’s daily schedule at the end of the day (Activity Recognition) is a widely studied problem and the different commercial subsystems related to this problem reveal that it is a growing field of interest. The relevance of activity recognition and classification will probably progressively grow in a near future, not only because of the wide range of applications in the human machine interaction field but also because of the need of machine assistance for an increasingly elderly society.

We proposed a system which will help the user to summarize his daily activity but, besides summarization capabilities, its aim is to be a tool that identifies which factors affect productivity. Daily activity reporting has recently become a duty when working in certain company environments. Some law firms, for instance, make their lawyers detail what they have been doing in their labor day and, in general, weekly or monthly reports are becoming
more and more common. Having a system that can compute statistics on the activity done over the day by analyzing long term multimodal data is a first step towards building intelligent personal agents able to manage time more efficiently.

Anyway, the possibilities of an Activity Recognition system go far beyond those cited. For example, for security purposes, workers who have to carry out dangerous works (nuclear station controllers, dangerous material transportation, etc) could be tested using activity recognition during a certain period of time in order to double check they are able to handle difficult situations.

Two main novelties have been discussed in this Project; first, a system that combines both physical and contextual awareness hardware/software to record synchronized audio-visual, body sensing, global position and computer monitoring data. Secondly, we proposed a new semi-supervised temporal clustering algorithm able to discover multimodal data efficiently and accurately.

We have shown preliminary results on constructing Multimodal Diaries (diaries of daily activity) from long-term multimodal sensing data testing from two different users during a period of data acquisition. After collecting data during several days in an office environment, we temporally clustered and classified the multimodal data in meaningful states of activities and we were able to identify some time-wasting or low-yield jobs. The results reached improved the performance of some state-of-art algorithms and showed a promising behavior.

However, using a Multimodal Acquisition System made us realize about the amount of time we waste checking e-mails on the internet. The first time you use an activity log you may be shocked to see the amount of time that you waste! Memory is a very poor guide when it comes to this, as it can be too easy to forget time spent reading junk mail, talking to colleagues, reading the newspaper, having coffee, eating lunch, etc.

The main goal we achieved with this work is that we managed to build a whole Activity Recognition system from scratch, combining pre-existing and specifically designed modules. So the main conclusion we take is that the system we built actually accomplished the initial specifications: it recognizes a set of activities from data collected by a sensor layout.

In fact some of the results were even better than it was initially expected, getting
encouraging and competitive results.

The system was tested and is currently being used in other departments at Carnegie Mellon University which ensures its future improvement.

Although the results are promising, they are still not statistically significant and more data should have to be collected.

The in-deep study of a system like this resulted in many lines of improving the performance. However, the time to develop the system and the steps to follow sometimes make these refinements impossible to perform. The next paragraphs propose some of the lines the author understands are the next step this work should take.

In this first attempt we managed to get to know the activities held during a day (with an accuracy of almost 90%) but in further work higher level information about the sampled data could be extracted (for example enhancing new sensors or proposing new feature extraction means).

On the other hand it would also be really useful to have an activity summarization tool to personally get to know how productive you are in terms of time spent doing working tasks versus low-value jobs (e.g. reading junk mail, making coffee or socializing with colleagues). Additionally a step further could take statistics from several months of data, keeping an historical record of activity and getting to know cycles of activity, productivity or stress, for example.

A part from the Activity Recognition functionality itself, the aim of our system is to be a tool that opens the doors to a complete new world of possibilities in terms of research. By identifying which factors affect productivity, this system can lead to deep studies on human behavior. For instance, it is interesting to discover which factors affect efficiency based on the breaks that the user takes, if the user takes a nap, the times and amounts the user eats, the quality of the nutrition, the environment the user works in, if he/she has internet or not, etc.

Beyond the time management application and particular company oriented applications, we plan to assist the elderly and individuals with disabilities by providing home capabilities that will monitor health trends and assist in the inhabitant’s day to day activities in their
own homes. The result will save money for the individuals, their families, and the state.

One trillion dollars of US healthcare costs per year are directly attributable to people’s lifestyle choices and the government of the US spends less than 5% of that addressing this issue. What if there was an unobtrusive, accurate way to gather the physical and mental states of people in their natural environments, in real time and over long periods of time? If such information could be obtained, we could start to address the fundamental issue in health and wellness: behavior modification. I am sure this project has taken us a little step closer to this goal.
Chapter 7

Bibliography

Bibliography


[8] Vannevar Bush. *Science, the endless frontier; a report to the President on a program for postwar scientific research.* National Science Foundation, Washington, 1960. 3


[28] Matthew Cooper Jonathan. Temporal event clustering for digital photo collections. 1.2.2


[34] J. Lin, M. Vlachos, E. Keogh, and D. Gunopulos. Iterative incremental clustering of time series, 2004. 1.2.2


[37] W. Liu and J. Yi. Existing and new algorithms for nonnegative matrix factorization. In *University of Texas at Austin*, 2003. 3.4.3

[39] Anmol Madan, Ron Caneel, and Alex Pentland. Groupmedia: Distributed multi-modal interfaces. 1.2.1


[61] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features, 2001. 2.2.2, 3.2.2, B


[63] V. Wan and W. Campbell. Support vector machines for speaker verification and identification, 2000. 2.2.2


Appendices
Appendix A

Main Clustering Algorithms

A.1 Hierarchical Clustering

Originally one of the most used techniques in Data Clustering\(^1\), the hierarchical clustering techniques pursue a goal of grouping clusters to make a new one or dividing them to result in two new ones so that this agglomeration or division maximizes any similarity measure among clusters in agglomeration or maximizes any divergence measure among clusters in the division.

The traditional representation of this hierarchy is a tree data structure (called a dendrogram, as the one shown in figure A.1.

Let's take for example an agglomerative example. Let \( n \) be the number of elements to cluster, grouped in \( C = n \) groups, one group per element, resulting the level \( K = 0 \). The next level will agglomerate those two clusters or groups with more similitude. In the \( K \) level there will be \( C = n - K \) groups.

The grouping procedure described above makes that in a particular level \( K \) two groups join together, they will keep together for the rest of levels.

The procedure of the hierarchical methods is really simple. For example the agglomerative algorithms explained above starts taking as much groups as number of elements

\(^{1}\text{In [51] it is outlined that in 1984 most of the clustering techniques were hierarchical: 'The literature search also showed that applications of hierarchical cluster analysis outnumber applications of nonhierarchical cluster analysis by more than ten to one.'}
to cluster \((C = n)\). Next the two nearest values are grouped following a certain measure (which will be described after). These steps are performed till the exit condition accomplishes. Tipically there are three exiting conditions:

- We reach the main group that contains them all \((C = 1)\)
- We reach a prefixed \(C = C_0\) number of clusters
- A clustering measure reaches a particular value (threshold)

Once two clusters are grouped new measures have usually to be recalculated. This fact makes this method really bad for large-scale data sets.

Distance Measure can be done in different ways:

- Single-linkage clustering (also called the connectedness or minimum method), where we consider the distance between one cluster and another cluster to be equal to the shortest distance from any member of one cluster to any member of the other cluster.

- Complete-linkage clustering (also called the diameter or maximum method), where we consider the distance between one cluster and another cluster to be equal to the greatest distance from any member of one cluster to any member of the other cluster.

- Average-linkage clustering, we consider the distance between one cluster and another cluster to be equal to the average distance from any member of one cluster to any member of the other cluster.
A.2. K-means

member of the other cluster.

A variation on average-link clustering is the UCLUS method of R. D’Andrade [1] which uses the median distance, much more outlier-proof than the average distance.

The divisive hierarchical does the reverse by starting with all objects in one cluster and subdividing them into smaller pieces. Divisive methods are not generally available, and rarely have been applied.

The two main problems of hierarchical clustering are its lack of scalability (time complexity of at least $O(n^2)$ being $n$ the number of objects) and the fact that they can not undo a previous cluster assignation.

A.2 K-means

K-means [16], [38] is one of the simplest and most popular unsupervised learning algorithms to solve the clustering problem.

The most popular exclusive and partitional clustering algorithm starts by partitioning the input points into $k$ initial sets, either at random or using some heuristic data. It then calculates the mean point, or centroid, of each set. It constructs a new partition by associating each point with the closest centroid. Then the centroids are recalculated for the new clusters, and algorithm repeated by alternate application of these two steps until convergence, which is obtained when the points no longer switch clusters (or alternatively centroids are no longer changed).

From a more rigorous point of view, K-means clustering splits a set of $n$ objects into $c$ groups by maximizing the between-clusters variation relative to within-cluster variation. That is, K-means clustering finds the partition of the data that is a local optimum of the following energy function:

$$J(\mu_1, \ldots, \mu_n) = \sum_{i=1}^{k} \sum_{j \in C_i} ||d_j - \mu_i||_2^2$$  \hspace{1cm} (A.1)
where $d_j^2$ is a vector representing the $j^{th}$ data point and $\mu_i$ is the geometric centroid of the data points for class $i$.

K-means usually starts with a random initialization and iterates between two steps until convergence. At first, K-means assigns each of the $n$ points to the closest of the $c$ means. In a second step, each of the means ($\mu_i \forall i$) is recomputed with the average of the points that are closer to it.

Although it can be proved that the procedure will always terminate, the K-means algorithm does not necessarily find the optimal configuration over all possible assignments. The algorithm is significantly sensitive to the initial randomly selected cluster centers, and it typically runs multiple times and the best solution is chosen. Despite these limitations, the algorithm is used fairly frequently as a result of its ease of implementation and effectiveness.

The optimization criteria in eq. A.1 can be easily rewritten in matrix form as:

$$E_{k-means}(M, G) = \| D - MG^T \|_F$$

subject to $G1_c = 1_n$

where $G \in \mathbb{R}^{n \times c}$ is a dummy indicator matrix such that $\sum_j g_{ij} = 1, g_{ij} \in \{0, 1\}$ and $g_{ij}$ is 1 if $d_i$ belongs to class $C_j$, $c$ denotes the number of classes and $n$ the number of samples. The columns of $D \in \mathbb{R}^{d \times n}$ contain the original data points, $d$ is the dimension of the data. It is easy to show that the two steps of K-means perform coordinate descent in $E_1(M, G)$. Given the actual value for the means $M$, the first step finds for each data point $j$, the $g_j$ such that one of the columns is one and the rest 0 and minimizes eq. A.2. The second step optimizes over $M = DG(G^T G)^{-1}$, equivalent to compute the mean of each cluster.

Having the matrix formulation, it is easy to verify (by eliminating the variable $M$) that eq. A.2 is equivalent to:

$$E_{k-means_2}(G) = \| D - DG(G^T G)^{-1}G^T \|_F = tr(D^T D)$$

$$-tr((G^T G)^{-1}G^T D^T DG) = \sum_{i=c+1}^{\min(d, n)} \lambda_i$$

\(^2\)See section 3.4.1 for notation details.
A.2. K-means

where \( \lambda_i \) are the eigenvalues of \( D^T D \). Minimizing eq. A.3 is equivalent to maximizing 
\[ \text{tr}((G^T G)^{-1} G^T D^T D G). \]
Ignoring the special structure of \( G \) and considering the continuous domain, the optimum is computed by the eigenvectors of the covariance matrix \( D^T D \).

Observe that maximizing \( \text{tr}((G^T G)^{-1} G^T D^T D G) \) is equivalent to maximizing the Rayleigh quotient \( \frac{|G^T D^T D G|}{|G^T G|} \). A similar reasoning has been reported by [15, 65], where they show that a lower bound of eq. A.3 is given by the residual eigenvalues. The continuous solution of \( G \) lies in the \( c-1 \) subspace spanned by the firsts \( c-1 \) eigenvectors with highest eigenvalues [15] of \( D^T D \). It is interesting to remark that PCA are the continuous solutions to the discrete cluster membership indicators for K-means clustering.

Finally, it is worthwhile to point out the connections between K-means and standard spectral graph algorithms such as Normalized Cuts [55, 13], by means of kernel methods. The kernel trick is a standard way of lifting the points of a dataset to a higher dimensional space [54], where points are more likely to be linearly separable (assuming the right mapping is found). Let us consider a lifting of the original points to a higher dimensional space, 
\[ \Gamma = [\phi(d_1) \phi(d_2) \cdots \phi(d_n)] \]
where \( \phi \) is a high dimensional mapping. The kernelized version of eq. A.2 will be:

\[ E_{sc}(M, G) = ||(\Gamma - MG^T)W||_F \quad (A.4) \]

where we have introduced a weighting matrix \( W \) for generality.

Eliminating:

\[ M = DW^T G(G^T WW^T G)^{-1} \quad (A.5) \]

it can be shown that (from equations A.4 and A.5 that

\[ E_{sc}(G) = -\text{tr}((G^T WW^T G)^{-1} G^T WW^T \Gamma^T \Gamma WW^T G) \quad (A.6) \]

where \( \Gamma^T \Gamma \) is the standard affinity matrix in Normalized Cuts [55]. After a simple
change of variable \( Z = G^T W \), the previous equation can be expressed as:

\[
E_{sc}(G) = -tr((ZZ^T)^{-1}ZW^T\Gamma W Z^T) \tag{A.7}
\]

Choosing \( W = \text{diag}(\text{diag}(\Gamma^T T_1)) \) the problem is equivalent to solving the Normalized Cuts problem. Observe that this formulation is more general since it allows for arbitrary kernels and weights, moreover K-means-like iterative algorithms can be used to minimize normalize-cut type of algorithms. Also, observe that the weight matrix could be used to reject the influence of a pair of data points with unknown similarity (i.e. missing data).

### A.3 C-means

Fuzzy C-means [17],[5] is an overlapping and partitional clustering algorithm, which allows one piece of data to belong to two or more clusters with a certain provability. This method is frequently used in pattern recognition.

Likewise K-means, the goal of the algorithm is minimizing an objective function, the one shown in eq A.8.

Let \( x_i \) be the \( i \)th data and \( c_j \) the center of the \( j \)th cluster. Using a definite norm we can state the objective function of C-means is the following:

\[
E_{C-means} = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^m ||x_i - c_j||^2 \tag{A.8}
\]

where \( m \) is a scalar (greater than 1), and \( u_{ij} \) is the degree of membership of \( x_i \) in the cluster \( j \).

Fuzzy partitioning is carried out through an iterative optimization of the objective function shown above, with the update of membership \( u_{ij} \) and the cluster centers \( c_j \). This two-step updating routine leads us to achieve the minimization goal.

After an initialization in the first iteration, the procedure consist of updating the center
vectors $c_j$ from the $u_{ij}$ values, using:

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^m x_i}{\sum_{i=1}^{N} u_{ij}}$$  \hspace{1cm} (A.9)

A second step will update $u_{ij}$ from the centers just calculated. The expression to do that is shown in equation A.10.

$$c_j = \frac{1}{\sum_{i=1}^{c} \left( \frac{||x_i - c_j||}{||x_i - c_k||} \right)^{\frac{2}{m-1}}}$$  \hspace{1cm} (A.10)

This two-step procedure is repeated till the exit condition is reached. This exit condition is usually related to the difference between the $u$ values in a iteration compared to the previous one.

This clustering algorithm and its variations have been deeply studied by the research community ([47] and [27] are only some of the examples) and has applications to fields like physics, medicine and spectroscopic imaging.

Its scalability and probabilistic meaning make it suitable for many data mining applications, specially those interacting with Hidden Markov Models[50] or any other statistical approach.

### A.4 Other Clustering Techniques

Although K-means and Fuzzy C-means are provably the most used clustering methods (see explanation and formulation in previous sections) there is a wide range of algorithms and variations that perform clustering in several other ways. The detail of those techniques is far beyond the scope of this Thesis and the writers will forward any curious reader to wider publications in the clustering area[49] or even the Machine Learning one [42].

It is worthwhile, thought, to take a look at the applications of other fields to clustering. Along these lines there are two interesting cases to point out, the Mixture of Gaussians
Technique and the Self Organizing Maps, where model-based signal processing and neural networks respectively, interfere with clustering.

The Mixture of Gaussians Technique (reviewed in [42]) proposes another way to deal with clustering problems: a model-based approach, which consists in using certain models for clusters and attempting to optimize the fit between the data and the model. In practice, each cluster can be mathematically represented by a parametric distribution, like a Gaussian (continuous) or a Poisson (discrete). The entire data set is therefore modeled by a mixture of these distributions. An individual distribution used to model a specific cluster is often referred to as a component distribution. The great advantage of this technique is that it is based upon well-studied statistical inference basis (similar, for example, to the one used in the Telecommunications field), which eases the modeling process. As Fuzzy C-means is a partitional and overlapping type of clustering.

On the other hand Self Organizing Maps (SOM) [29] propose a training phase where the algorithm learns from the data (as a training process in Artificial Neural Networks, because, in fact SOM is an ANN) and a testing set, where the algorithm assigns the closest cluster in the map (exclusive assignment) to the incoming data (partitional procedure, non hierarchical). This approach is far beyond the constraints of this Thesis because it requires several training sets and some would say that is no more a clustering technique but a classification one. Anyway combinations of Self Organizing Maps and other algorithms (for example K-means in [33]) have been performed in several environments. Even [30] uses them for daily activity discovery.
Appendix B

Clustering Application:
Recognizing People in a Meeting

The goal of the CALO Project\(^1\), which stands for Cognitive Agent that Learns and Organizes, is to create cognitive software systems, that is, systems that can reason, learn from experience, be told what to do, explain what they are doing, reflect on their experience, and respond robustly to surprise. Funded by DARPA\(^2\) and coordinated by SRI International, part of the Software was developed at Carnegie Mellon University, which is one of the 22 organizations where the studies are held.

The CAMEO\(^3\) [52], which stands for Camera Assisted Meeting Event Observer, is the physical awareness (PA) component of a larger effort to develop an enduring personalized cognitive assistant that is capable of helping humans handle the many daily business/personal activities that they engage in. The CAMEO, is a sensory system designed to provide an electronic agent with physical awareness of the real world. It consists of a set of four cameras oriented in such a way as to capture a panoramic video stream of the world. Instead of instrumenting meeting rooms with large numbers of calibrated cameras, CAMEO is intended to be used more like a speaker phone for a conference call. That is, a CAMEO

\(^1\)Detailed information, publications and other information related to the CALO Project can be found in its website http://caloproject.sri.com/
\(^2\)The Defense Advanced Research Projects Agency
\(^3\)Details of the CAMEO CALO-PA can be found in its website http://www.cs.cmu.edu/ cameo/
device will be brought into a meeting and simply placed in the center of the room without requiring special calibration. As such, CAMEO is designed to be used in environments where those who are participating in the meetings agree to and welcome the use of such an electronic assistant\textsuperscript{4}.

Part of the CAMEO project developed for the last year of funding at Carnegie Mellon University\textsuperscript{5} consisted of making a people recognizer for a meeting.

The recognizing tool has several utilities, functions and modules absolutely out of the scope of this section. However, the aim of this section is to focus on the role of clustering in a chain of events.

Figure B.1 shows the block diagram for the proposed people recognizer based on CAMEO.

The goal of the whole process is stating how many people is there in a certain meeting and identify them using only video information from a pre-populated image database. Based on the work done in [52], the pre-existing software is able to track people in a local environment and specially when frontal faces are captured (because is a combination of the face detector

\textsuperscript{4}Initially the CAMEO device (Omnidirectional Camera and ambiental sound) was proposed to be part of the sensing layout of this Thesis. In fact 3 days were recorded with this information. The device was discarded because of the size of the files recorded and the processing time required for feature extraction. Audio information was proposed to be recorded together with the personal camera which made the data collection easier

\textsuperscript{5}Mainly implemented by Oriol Vinyals and Carlos Agell (writer of this Thesis). In [60] other details about this and other applications based on the CAMEO project are detailed.
in [61] and the tracker used in [64]). So the system builds a data structure containing all the faces from a 'tracker' but does not give any information about the relationship among 'trackers'.

The problem then is try to relate groups of pictures identified to be the same person (see second step in figure B.1). At the same time some of the trackers contain outliers\(^6\).

Here is where the crucial role of clustering takes part. As we are able to extract features from each group of faces (Tracker) we are then able to compare groups in an equitable framework. This means that we can cluster in an unsupervised manner the groups of faces so that the resulting clusters have the different trackers of each of the meeting assistants.

The clustering technique is here useful because it allows an easy way of organizing the information and group similar data in order to state that the people in each group is the same. It does not need the role of the supervisor and that is why here this method can be used.

![Figure B.2: People Recognition Input (above) and output (below)](image)

Any clustering method would be good for this case but the temporal and supervised terms would not make sense because the context does not ease its adaptation. K-means and DCA have shown to perform correctly.

The rest of the system architecture (Outlayer detector and Recognition modules) are implemented using completely different techniques\(^7\) which lead to successfully achieve the

---

\(^6\)Images identified to be faces but they are actually other things.

\(^7\)The outlayer detector uses color information as features to decide if the image contains a face or not.
recognition goal. Figure B.2 shows a frame of the video recorded with CAMEO and its corresponding output from the people recognizer.

---

On the contrary the Recognition Module extracts features from the clusters of trackers and compares them with the features in the database.
Appendix C

Paper Submitted

This Appendix contains the paper submitted to ICME’07 (International Conference on Multimedia & Expo) related to the work explained in this Project.

The paper was accepted on April 25th 2007 and will be exposed on July 4th 2007 in the Multimedia Modeling, Specification, User Interface category.
MULTIMODAL DIARIES

Fernando De la Torre    Carlos Agell
Robotics Institute. Carnegie Mellon University. 5000 Forbes Av., Pittsburgh, PA 15213 USA

ABSTRACT
Time management is an important aspect of a successful professional life. In order to have a better understanding of where our time goes, we propose a system that summarizes the user’s daily activity (e.g., sleeping, walking, working on the pc, talking, ...) using all-day multimodal data recordings. Two main novelties are proposed:

- A system that combines both physical and contextual awareness hardware and software. It records synchronized audio, video, body sensors, GPS and computer monitoring data.
- A semi-supervised temporal clustering (SSTC) algorithm that accurately and efficiently groups large amounts of multimodal data into different activities.

The effectiveness and accuracy of our SSTC is demonstrated in synthetic and real examples of activity segmentation from multimodal data gathered over long periods of time.

1. INTRODUCTION
Effective time management is an important aspect of a successful professional life. There exist many techniques to effectively manage your time, and several of them include the popular “to-do list”, that is, making an inventory of your daily schedule. In this paper, we propose a system that summarizes the user’s daily activity from multimodal sensors. Having a system that can compute statistics on the activities done over the day is a first step towards building intelligent personal agents able to manage time more efficiently.

Two main novelties are discussed. Firstly, a system that combines both physical and contextual awareness sensors to record synchronized audio-visual, body sensing, global position and computer monitoring data; secondly, we propose a semi-supervised clustering algorithm able to temporally segment multimodal data efficiently and accurately. Fig. 1 shows an example of an office scenario recording, where several synchronized modalities are used to summarize the user’s activities.

2. SENSORS
In this section, we describe the sensors and the features used for temporal segmentation of activities.

Fig. 1. Synchronized multimodal recording (body sensor, video, audio, computer monitoring and GPS.)

2.1. Physical Awareness Sensors
We use the SenseWear armband from BodyMedia (fig. 2.a) as a physical awareness device. The SenseWear armband [1, 2] combines five different sensors: 2-axis accelerometers, galvanic skin response, skin temperature, heat flux, and near-body ambient temperature. The SenseWear is synchronized with the computer, and it is worn during the entire day on the upper right arm. We have used sleeping, physical activity, lying down, and standing up as features for activity segmentation.

2.2. Context Awareness Sensors
For context awareness we have used the Logiteck Quickcam camera to record audio and video, a computer monitoring software to record the programs that the user is running and a wearable GPS device (see fig. 1).

The video features consist of a binary stream, where 1 corresponds to detecting the user’s face and 0 otherwise. We use the OpenCV face detector from Viola and Jones [3], and color filters to reduce the number of false positives. To extract
audio features, we compute the Mel Frequency Cepstral Coefficients (MFCC) over 20 ms. These features are used to train a Support Vector Machine (SVM) [4] for classifying the audio signal into four states: user talking, other people talking, typing and silence.

To record the interaction of the user with the computer we use Activity Monitor from Softactivity. We classify at 1 Hz the programs that the user is running into 3 categories: work, non-work and internet surfing. Every time an unclassified program shows up the user is prompted to classify it into one of those groups.

To track people outdoors, we use a wearable wrist-strap GPS receiver. The GARMIN Foretrex 201 has an accuracy of 15 m and an updating frequency of 1Hz, see fig. 2(b). Using the coordinates logged by the device (longitude and latitude) the programs that the user is running into 3 categories: work, non-work and internet surfing. Every time an unclassified program shows up the user is prompted to classify it into one of those groups.

To track people outdoors, we use a wearable wrist-strap GPS receiver. The GARMIN Foretrex 201 has an accuracy of 15 m and an updating frequency of 1Hz, see fig. 2(b). Using the coordinates logged by the device (longitude and latitude) and the time stamp, we can estimate the mean speed.

3. SEMI-SUPERVISED SPATIO-TEMPORAL CLUSTERING (SSTC)

Given the set of multimodal features described in the previous section, our goal is to segment the data into temporally coherent chunks. In this section, we extend standard clustering algorithms (e.g. K-means or spectral graph methods) to incorporate temporal coherence and semi-supervised information.

3.1. Discriminative Cluster Analysis (DCA)

DCA [5] is a clustering method that combines both clustering and discriminative dimensionality reduction in an unsupervised manner. DCA minimizes:

$$E_{DCA}(B, V, G) = ||(G^T G)^{-\frac{1}{2}}(G^T - VB^T D)||_F$$ (1)

where $D \in \mathbb{R}^{d \times n}$ (see notation\(^{1}\)) is a data matrix such that each column $d_i$ corresponds to a sample of multimodal features at one time instant. $G \in \mathbb{R}^{n \times c}$ is a dummy indicator matrix, such that $\sum_j g_{ij} = 1$, $g_{ij} \in \{0, 1\}$ and $g_{ij} = 1$ if $d_i$ belongs to class $C_j$, $c$ denotes the number of classes and $n$ the number of samples. $B \in \mathbb{R}^{d \times k}$ and $V \in \mathbb{R}^{c \times k}$ are reduced rank approximation matrices. Considering the simpler case where $B = I_d$, after eliminating $V$, eq. 1 is proportional to:

$$E_{DCA}(G) \propto tr(D^T (DD^T)^{-1} DG (G^T G)^{-1} G^T)$$ (2)

Relaxing the constraints on $G$, so that $g_{ij} \geq 0$ and $G1_c = 1_n$, a gradient descent strategy can efficiently find a local optimum of eq. 2, see [5] for details.

3.2. Temporal term

DCA does not take into account any temporal coherence of the cluster labels or incorporate semi-supervised information. One of the benefits of relating the clustering problem to the optimization of an objective function (e.g. [5]) is that we can easily add additional constraints as a penalty term.

In order to penalize non-smooth changes (over time) on the labels, we encourage that $g_{i \cdot}$ and $g_{i \cdot +1}$ have similar values by minimizing:

$$E_t = \sum_{i=1}^{n-1} ||g_{i} - g_{i+1}||_2^2 = ||G^T - G^T P||_F,$$

where $P$ is a known permutation matrix (left shift of the identity matrix). Moreover, adding dynamic information and a normalization factor, the temporal term transforms to:

$$E_t = ||(G^T G)^{-\frac{1}{2}}(G^T - AG^T P)||_F$$ (3)

A encodes the state dynamics and the matrix $(G^T G)^{-\frac{1}{2}}$ is a normalization factor for DCA. If $G$ is known, the optimal $A$ can be computed as: $A = G^T P^T G (G^T G)^{-1}$.

3.3. Adding semi-supervised information

In this section, we add two types of semi-supervised information to the clustering: the must-link and the cannot-link term.

Let $\mathbb{N}^n$ be the set of samples that belong to the same class. $e_r \in \mathbb{R}^n$ denotes an indicator vector for data point $d_r$ so that $De_r = d_r$. We formulate the must-link supervised additive term as follows:

$$E_{sML} = \sum_{i,j \in \mathbb{N}^n} ||g_{i} - g_{j}||_2^2 = ||GE_{ML}||_F$$ (4)

where $g_{i} - g_{j} = G(e_i - e_j)$ and $E_{sML} \in \mathbb{R}^{c \times l}$ is a matrix with $l$ columns corresponding to the number of pairs of data points that belong together, and each column contains the vector $e_i - e_j \in \mathbb{N}^n$ that defines a pair of must-link points.

An analogous cannot-link term ($E_{sCL}$) can be defined out of the set $\mathbb{N}^D$ of cannot-link pairs, defining $E_{CCL} \in \mathbb{R}^{c \times l}$ as a matrix, where each column contains the vector $e_i - e_j$ that defines a pair of cannot-link points.
3.4. Optimization

Combining all the terms, the semi-supervised spatio temporal clustering algorithm optimizes:

\[ E_{\text{semi}}(G) = E_{\text{cluster}} + \lambda E_1 + \beta_1 E_{\text{SML}} - \beta_2 E_{\text{SCL}} \]  

(5)

\( E_{\text{cluster}} \) can be K-means, DCA or spectral clustering, see [5] for the details. The parameters \( \lambda, \beta_1, \beta_2 \) are normalization factors to make \( E_1, E_{\text{SML}}, E_{\text{SCL}} \) and \( E_{\text{cluster}} \) comparable in terms of energy.

To cluster, we perform gradient descent in eq. 5 with a line search strategy. To impose non-negativity constraints on \( g \) terms of energy.

\[ \frac{\partial E_1(V^n)}{\partial V} = V \circ (2GA^T(G^T G)^{-1}A \ldots -2G(G^T G)^{-1}AG^T(G^T G)^{-1} \ldots +4G(G^T G)^{-1}G^T PGA(G^T G)^{-1} \ldots -2P^TGA^T(G^T G)^{-1} - 2PG(G^T G)^{-1}A) \] 

\[ \frac{\partial E_{\text{SML}}(V^n)}{\partial V} = V \circ (2E_1E_1^T G) \]

Optimizing eq. 5 w.r.t \( G \) is a non-convex optimization problem that, without a good starting point, is likely to get stuck at a local minimum. To improve clustering results, we use a top-down approach where a multiresolution scheme is employed. That is, we first decimate the data and apply the clustering scheme at the lowest resolution level and propagate the result to higher levels. The multiresolution scheme has two main benefits: firstly, it is faster and more accurate; secondly, the first order temporal constraints (i.e. \( E_t \)) imposed in the lower resolution are expanded to higher order terms in the full resolution.

4. EXPERIMENTS

In this section, we report results in both synthetic and real examples of the proposed semi-supervised temporal clustering.

4.1. Synthetic experiments

Fig. 3 shows a two-level piece-wise constant signal at levels 5 and 10, with added Gaussian noise(N(0,0,3)) and some glitches. These glitches occur naturally in our system by the discontinuity in the audio-visual classifiers. Ideally, we would like to segment this 1D signal into a square wave. Using standard DCA or k-means does not lead to a correct clustering because of the glitches and noise (see fig. 3 top). However, the multiresolution version of DCA with temporal consistency finds the desired solution (see fig. 3 bottom).

Fig. 4 illustrates the use of the semi-supervised term. Fig. 4 shows an example where five 2-dimensional Gaussians can be clustered differently on two clusters based on the supervised term (must-link pairwise constraint).

4.2. What did I do today?

For many people (including the authors), it often seems that at the end of the day not all the expected work has been done. Inevitably, the same question comes to mind: What did I do today? In this section, we use the SSTC algorithm to segment our daily activity from multimodal data. Later we provide statistics of the time spend in each task for user self-awareness (e.g. amount of time doing low-value jobs such as reading junk e-mail).

We have collected data every second, over a period of three days in an office scenario for two different people. Fig. 1 shows a typical example of all the data gathered at a particular time instant. All this multimodal data (between 5 and 9 hours for each person/day) is manually classified into 8 types of activity: Sleeping, Walking, Away (inside the building), Away (outside the building), Working(no PC), Internet Surfing, Working on PC and Talking.

We use our SSTC algorithm to temporally cluster this
data. The program runs on Matlab, and we use a multiresolution strategy with 7 levels. Fig. 5.b shows the results of using semi-supervised spatio-temporal DCA clustering for 6 hours and 10 minutes of recording.

We compare the results obtained from several clustering methods: k-means, DCA, DCA + temporal term (DCA+TT) and DCA + temporal + semi-supervised term (DCA+TT+SST). The accuracy of the clustering is given by the number of correct samples over the total number of samples. This accuracy measure requires correct and precise labeling information for each day (user annotated data). Table 1 shows the clustering accuracy for all the algorithms described in section 4 with the data collected. A video with the results of the clustering for a particular user can be downloaded from www.cs.cmu.edu/~ftorre/IcmeVideo.mpg.

It is also interesting to analyze which activities are easier to cluster. Table 2 reports the clustering accuracy for each of the activities.

5. CONCLUSION

In this paper we have proposed a context and physical awareness system to monitor the daily activities of a user. To temporally segment the activities, we have extended traditional clustering algorithms by adding side information and temporal consistency to the clusters. We are currently working on extending the number of activities and analyzing which factors make the user more productive.

<table>
<thead>
<tr>
<th>State</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sleeping</td>
<td>64.05±5.2%</td>
</tr>
<tr>
<td>Walking</td>
<td>91.02±2.1%</td>
</tr>
<tr>
<td>Working no PC</td>
<td>69.45±2.5%</td>
</tr>
<tr>
<td>Internet Surfing</td>
<td>89.78±1.4%</td>
</tr>
<tr>
<td>Working PC</td>
<td>92.10±2.7%</td>
</tr>
<tr>
<td>Talking</td>
<td>60.26±81.1%</td>
</tr>
<tr>
<td>Away(inside)</td>
<td>73.21±6.3%</td>
</tr>
<tr>
<td>Away(outside)</td>
<td>94.12±1.3%</td>
</tr>
</tbody>
</table>

Table 2. Activity accuracy.

Acknowledgements This work has been partially supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. NBCHD030010.

6. REFERENCES