THESIS

UBIQUITOUS USER INTERFACES:

A FRAMEWORK FOR CAMERA PROJECTOR BASED UBIQUITOUS USER INTERFACE

Submitted by

IDRISSOU Amina

HASSANI Mohamed Amine

Department of computer sciences

In partial fulfillment of the requirements

For the Degree of Masters on networks and distributed systems (academic)

University A.MIRA of Bejaia

Fall 2013

Master's Committee:

Advisor: Dr D.IDOUGHI

Mr.ALOUI

Mr.BAADACHE

Mr.ALLICHE

Ms.BOUTRID

ABSTRACT

Ubiquitous User interfaces is a very large field, involving many aspects and giving many choices. It aims to provide interaction between the system and the user in natural, transparent and calm manner. One of the promising interfaces is Camera-Projector Ubiquitous user interface. Some its concepts, such as multimedia, multimodal, wearable, tangible, or augmented reality, make this kind of interfaces very complex systems. This paper presents a state of art about these user interfaces and proposes a framework integrating the cited aspects, for design and implementation of Ubiquitous User interfaces based on cameras and projectors. This system supports distributed interfaces, smart objects and many modalities. It recognizes human gesture and augments the physical objects; it displays information according to the user that interacts with the system and its context.

RÉSUMÉ

Les interfaces utilisateur ubiquitaires est un domaine très vaste, impliquant de nombreux aspects et donnant beaucoup d'alternatives. Il vise à assurer des interactions entre le système et l'utilisateur d'une manière naturelle, transparente et calme. Une des interfaces prometteuses est l'interface utilisateur Ubiquitaire basée Camera-projecteur. Certains de ses concepts, tels que le multimédia, la multi modalité, portabilité, l'interaction tangible et l'augmentation de la réalité, font que ce genre d'interfaces utilisateurs sont très complexes. Ce document présente un état de l'art sur ces interfaces utilisateurs et propose un Framework intégrant les aspects cités, pour la conception et la mise en œuvre d'interfaces utilisateur Ubiquitaires basée sur des caméras et des projecteurs. Ce système prend en charge la distribution d'interfaces, les objets intelligents et de nombreuses modalités. Il permet la reconnaissance de gestes humains et l'augmentation des objets, il fournit les informations en fonction de l'utilisateur qui interagit avec le système et de son contexte. Une validation par scénario d'U-Learning a été appliquée pour prouver l'applicabilité du Framework sur le terrain.

ACKNOWLEDGMENTS

This study was conducted at University of Abderrahmane Mira in Bejaia.

First, we would like to thank the god Almighty for giving us the courage and the will to realize this project.

We also wish to thank all those who contributed meadows directly or indirectly in the development of this modest work in particular:

We wish to thank our supervisor, Dr D.IDOUGHI, for introducing us to the fascinating field of Ubiquitous Computing, We benefit a lot from his valuable academic experience and insightful comments on our project work.

I would like to thank my thesis committee members for excepting to judge our work, for their valuable time and their important remarks.

Thanks are due to our families, to our parents, who from afar still provide unending love, encouragement and support. We would also like to thank our friends for theirs care and encouragements, we cite :
' patati', 'yas', 'Assia', 'Zouza' , 'Med', 'Rim', 'Mido'

Thanks to our class mates.

Thanks for you all.

CONTENTS

Figures liste **7**

Introduction **8**

CHAPTER I. Overview of Ubiquitous Computing **14**

1. Definition of Ubiquitous Computing 14
2. Aspects and Challenges 15
   - 2.1. Intelligent infrastructure 15
   - 2.2. Context awareness 15
   - 2.3. Resource-Constraints 16
   - 2.4. Privacy and trust 16
   - 2.5. User interfaces 16
   - 2.6. Experience and social impact 17
3. Ubiquitous user interface 17
   - 3.1. Caracteristics 17
   - 3.2. Classes of UUI 18
   - 3.3. Input and output technologies 24
   - 3.4. Cycle of life 25
4. Conclusion 29

CHAPTER II. Camera projector based user interface (CPUI) **30**

1. Computer vision-based user interfaces 31
   - 1.1. Computer vision 31
   - 1.2. Image Formation 34
   - 1.3. Vision based user interface 38
   - 1.4. Computer vision in UUI 39
2. Projector based user interface 39
   - 2.1. Presentation of projection 39
   - 2.2. Projector based user interface 41
   - 2.3. Projector based user interface in UUI 43
3. Camera projector based user interface as UUI 44
4. Conclusion 45

CHAPTER III. STATE OF ART (POSITIONING OF THE SEARCH PROBLEMATIC) **46**

1. IBM's Steerable Interface EDML Framework [70] 46
2. Distributed Wearable Augmented Reality Framework for rapid prototyping software infrastructure [71] 48

3. Molyneaux and Gellersen's Distributed Framework [72] 50
4. real-time framework for vision based hand detection and recognition system [73] 53
5. Hand gesture based user interface for computer using a camera and projector[76] 55
6. Synthesis 55
CHAPTER IV. A framework for a camera projector ubiquitous user interface 59
1. Framework description 59
1.1. The input manager 61
1.2. The external services for input and the external services for output 62
1.3. The profiler 62
1.4. The application 63
1.5. The output manager 64
1.6. External context manager 65
2. Conclusion 66
CHAPTER V. Validation of the search proposition 67
1. Introduction 67
2. Presentation of the scenario 67
3. Presentation of the elements of the scenario 69
3.1. The human actors 69
3.2. Non human elements: 70
4. Application of our framework on the scenario 73
4.1. Implementation aspect 73
4.2. Environment and adaptation to the palace: 73
4.3. Passive execution 74
5. Conclusion 76
Conclusion and perspectives 77
References 79

FIGURES LISTE

Fig. 1. Trends in Computing sales. 9
Fig. 2. (a) Urp: A Workbench for Urban Planning and Design [23]. (b) transBOARD: a networked digital whiteboard with hyperCARDs. 19
Fig. 3. Datafountain: Money Translated to Water. 20
Fig. 4. The ambientROOM. 21
Fig. 5. Two users interactiong with a DiamondTouch: collaborative work environment. 21
Fig. 6. Low-Cost Interactive Whiteboard Using the Kinect 22
Fig. 7. Mobile AR restaurant guide. (a) User with MARS backpack, looking at a restaurant. (b) Annotated view of restaurant, imaged through the head-worn display. 23
Fig. 8. Virtual Cable augmented outside a car. 23
Fig. 9. Multimodal shopping environment. 24
Fig. 10. (Left) Wearable Multitouch Interaction Everywhere. (Right) LuminAR: Portable Robotic Augmented Reality. 30
Fig. 11. Interaction Techniques for Projected Displays. 31
Fig. 12. The area that involves Computer vision 33
Fig. 13. What is computer vision 34
Fig. 14. filtered monochrome images (color separations) madefrom the four box scene (from left to right) Red,Green,Blue. 37
Fig. 15. Complex object fast reconstruction using ToF camera 37
Fig. 16. shape detection using multiflash camera. 37
Fig. 17. (left) High-end Digital Light Range DL.1 to DL.3 steerable projector (right) Mitsubishi Pocket Projector 40
Fig. 18. The Projector system family decomposed by mobility and display size, compared to traditional desktop monitors. 41
Fig. 19. (Left) Low-Cost Multi-Touch Sensing through Frustrated Total Internal Refl ection. (Right) User Interface by Virtual Shadow Projection. 41
Fig. 20. (left) The iLamps Project developed a handheld projector-camera system, (right) Detecting circular fiducial markers to augment a wall scene with projection 42
Fig. 21. Adaptable projection 43
Fig. 22. From GUI to Tangible User Interfaces 45
Fig. 23. IBM’s Steerable Interface EDML Framework. 47
Fig. 24. A generic functional decomposition of Ubiquitous Augmented Reality User Interfaces . 49
Fig. 25. Functional decomposition of DWARF specific user interface components. 50
Fig. 26. D.Molyneaux and H.Gellersen’s Distributed System Architecture Overview. 52
Fig. 27. D.Molyneaux and H.Gellersen‘s Detection Method Selection. 53
Fig. 28. A Real-time Hand Gesture Detection and Recognition Framework 54
Fig. 29. Diagram of gesture-based user interface system 55
Fig. 30. The framework proposed for a camera projector based ubiquitous user interface 60
Fig. 31. The input manager 61
Fig. 32. The output manager 64
Fig. 33. The class room from student’s point of view 68
Fig. 34. The class room from teacher’s point of view 68

INTRODUCTION

This paper is introduced as a part of the final project of studies to obtain a masters degree. Its studies ubiquitous user interfaces, and proposes a generic framework for a camera projector based ubiquitous user interface. Taking advantage from and considering solutions and constraints of ubicomp, computer vision and projected user interfaces. The system proposed is mainly meant to recognize human gesture and augments the physical objects, with additional digital information. It displays information according to the user that interacts with the system.

Following previous concepts, this paper is divided into five chapters: First chapter presents an overview of Ubiquitous computing and ubiquitous interfaces. Chapter II gives a survey about camera-projector systems. Chapter III discuses a set of systems with points in common with our project. Chapter IV is dedicated to the presentation of the proposed framework and its generic architecture. To validate our proposition, Chapter V presents a scenario of application showing its feasibility and usability.

In the computers per person classification [1], the modern computing is divided to three eras (fig. 1). To the third and last one is given the term: Ubiquitous computing, or Ubicomp. After the mainframe era, a single large time-shared computer owned by an organization and used by many people at the same time. Then the PC era, a personal computer primarily owned and used by one person, and dedicated to them. Ubiquitous computing, representative of the present time, is characterized by the explosion of small-networked portable computer products in the form of smart phones, personal digital assistants (PDAs), and embedded computers built into many of the devices we own. Resulting a world in which each person owns and uses many computers. Each era has resulted in progressively larger numbers of computers becoming integrated into everyday life.

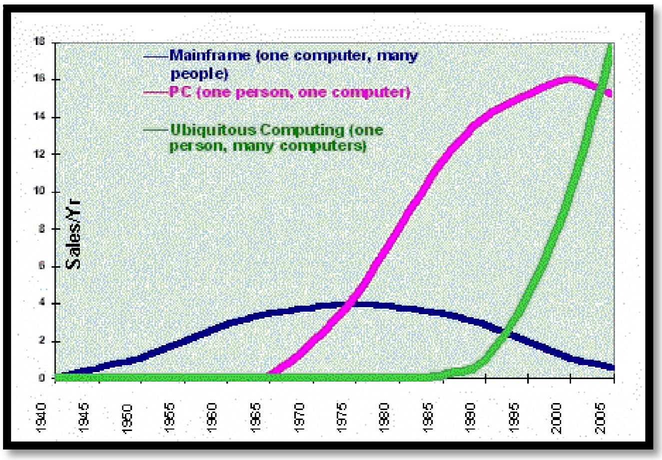


Fig. 1. Trends in Computing sales [1].

The early informative research in this area began in the late 1980s and was pioneered by Xerox Palo Alto Research Center (PARC), IBM Research, Tokyo University, University of California (UC) Berkeley, Olivetti Research, HP Labs, Georgia Institute of Technology (Georgia Tech), and Massachusetts Institute of Technology (MIT) Media Laboratory. and in the 1990s, the ubiquitous trend has began to be joined by commercial actors, such like Apple which created the PDA term with their first ubiquitous product "Apple Newton", then got joined by Fujitsu with their series of tablet and palm-based devices, and later the Archos and Apple MP3 players joined the trend.

The quantity and the diversity of the entities and the goals that served the ubiquitous computing development since it beginning has shaped the nowadays Ubicomp being and philosophy, but the high dependency on technological innovations still make its future unpredictable.

The original term ubiquitous computing was coined by Mark Weiser in 1988, at Xerox PARC, while serving as the director of the Computer Science Laboratory (CSL). In his description, he envisioned a future in which computing technologies became embedded in everyday artifacts, were used to support daily activities, and were equally applicable to our work, managing our homes, and for play. In his famous paper [2] figures his highly descriptive citation: "The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it". In his works, Weiser also introduced the notion of bits-per-second per-cubic-meter to the Ubicomp vision [3], wish resulted from his ubiquitous vision that the mobile and embedded processors can communicate with each other and the surrounding infrastructure, a concept that needs to manage the bandwidth reuse in first place by setting limitations on the range of communication.

Today, demonstrating the most convincing evidence of the value of ubiquitous computing, the cell phone, or more precisely the “Smartphone” takes center stage crossing a threshold of processor performance, memory/disk capacity, and connectivity both cellular and local, making it the most widely adopted and ubiquitous computer there has ever been. In the remaining sections, we follow the path of research that has defined ubiquitous computing since its beginning, and discuss the various approaches and some of the philosophies that have grown up around the work.

It is a matter of fact that Ubiquitous computing represents an evolution from the notion of a computer as a single device, to the notion of a computing space comprising personal and peripheral computing elements and services all connected and communicating as required.

In Ubicomp, computer functions are built into the basic objects (embedded) without notice their presence (calm). In other words, this model must be introduced to the integrity of daily life in a way that it will be indistinguishable from it. It is an innovative concept that necessitates new generation of technology. And require new methods and techniques of modeling to include new forms of human-machine interfaces. However, some nowadays systems are considered ubiquitous but not evolved enough to reach the entire vision of Weiser [4].

Ubicomp involves several fields of continues research such as artificial intelligence, and sensors networking, telecommunication, HMI, embedded systems design and even social sciences. It depends to their development and progress. Altogether, make the base that Ubicomp research advance.

Besides of that, this new vision of systems has created more affordances and new problems to solve. Firstly, Ubicomp is more rigorous in the material specification. The old generation of technology cannot all be adopted for the next generation. The new devices should be smaller, more complicated but must still cheap and accessible. Secondly, their embedded software must consider eventually a big number of scenarios according to the collected context but with a way that allow saving energy and keeping the security of personal information.

The user interface, in the industrial design field of human–machine interaction, is the space where interaction between humans and machines occurs. The goal of this interaction is effective operation and control of the machine on the user's end, and feedback from the machine, which aids the operator in making operational decisions. Examples of this broad concept of user interfaces include the interactive aspects of computer operating systems, hand tools, heavy machinery operator controls, and process controls. The design considerations applicable when creating user interfaces are related to or involve such disciplines as ergonomics and psychology.

A user interface is the system by which people (users) interact with a machine. The user interface includes hardware (physical) and software (logical) components. User interfaces provide a means of: (1) Input, allowing the users to manipulate a system, and (2) Output, allowing the system to indicate the effects of the users' manipulation.

Generally, the goal of human-machine interaction engineering is to produce a user interface, which makes it easy, efficient, and enjoyable to operate a machine in a way to produce the best result. This generally means that the operator needs to provide minimal input to achieve the wanted output, and that the machine minimizes undesired outputs to the human.

With the increased use of personal computers and its development over the time, the following types of user interface are the most common and known [5]:

- Command line interfaces, or text-based interfaces, where the user provides the input by typing a command string with the computer keyboard and the system provides output by printing text on the computer monitor. Used by programmers and system administrators, in engineering and scientific environments, and by technically advanced personal computer users. It is the first interface designed for the first generations of computers.

- Graphical user interfaces (GUI) accept input via devices such as computer keyboard and mouse and provide articulated graphical output on the computer monitor. That allowed to enlarge the use of computers to non-specialist users. The GUI has tended to dominate what is considered a user interface.

- Menu-based Interfaces are graphical user interfaces based on recognition of the command names, rather than recollection. Further, in a menu-based interface the typing effort is minimal as most interactions are carried out through menu selections using a pointing device. This factor is an important consideration for the occasional user who cannot type fast. Composing commands in a menu-based interface is not possible and the design of a menu-based interface is to structure large number of menu choices into manageable forms is a real challenge.

- Direct Manipulation Interfaces are graphical user interfaces presented to the user in the form of visual models (i.e. icons or objects). For this reason, direct manipulation interfaces are sometimes called as iconic interface. In this type of interface, the user issues commands by performing actions on the visual representations of the objects. Important advantages of iconic interfaces include the fact that the icons can be recognized by the users very easily, and that icons are language-independent. However, direct manipulation interfaces can be considered slow for experienced users.
- Touch user interface are graphical user interfaces using a touchpad or touchscreen display as a combined input and output device.
- Intelligent user interfaces are human-machine interfaces that aim to improve the efficiency, effectiveness, and naturalness of human-machine interaction by representing, reasoning, and acting on models of the user, domain, task, discourse, and media (e.g., graphics, natural language, gesture).

Clearly, a keyboard, screen, and mouse with GUI elements tied to every device affording computational interaction cannot be the future. The trend is going to more informal, intelligent and natural interfaces [4].

The user interfaces is one of the main research fields in Ubicomp. Ubiquitous interfaces aims to provide interaction between the system and the user in natural, transparent and calm manner. Everyday objects will could collect and process information. It is not a simple task to make an interface ensuring Ubicomp criteria. The huge amount of interaction required by this kind of systems then the necessity to make it most natural than possible make that the old generation of interfaces inadequate and new kinds are created to replace them. We can site many interesting and promising varieties of interfaces.

Cameras are widely used in ubiquitous systems as natural input. They are useful to get tangible interaction, gesture detection or even as sensors to get the context. Projectors are integrated in many successful projects to provide multimodal interaction and augmented reality system and have proved that they are suitable for creating useable applications that incorporate many qualities.

In this thesis, we tried to address some research aspects In Ubiquitous Computing. Before addressing the research question we carried out an as thorough as possible review of literature relating to the ubiquitous computing systems and Ubiquitous User interfaces. We highlighted some of the main challenges that have been identified in both of these areas.

From the studied ubiquitous user interfaces, we focused on camera projector user interfaces. One of the main problems is the difficulty of building a user interface in ubiquitous environments. By this work, we aim to help designers and developers by building a basic framework that they can carry on.

Once enlightening the problematic, we studied some works concerning the same search domain as our problematic and synthesized theirs contributions. We traced our goal by inspiring on these works to propose our own solution in form of a framework.

After detailing the proposed framework, we proceeded to its validation by showing one of its deployment scenarios.

Following these concepts, this paper introduces a framework proposal and a system based on camera projector interface and directed for learning and public environments.

This system recognizes human gesture and augments the physical objects, in a learning environment, with additional digital information. It displays information according to the user (student or teacher) that interacts with the system.

This paper is organized as follows. Section II presents an overview of Ubiquitous computing and ubiquitous interfaces. This section gives a survey about camera-projector systems. Section III presents a set of systems with points in common with our project. Section IV is dedicated to the presentation of the supporting framework, highlighting the generic architecture.

Finally, in Section VI, conclusions are made and future work is presented.

CHAPTER I. OVERVIEW OF UBIQUITOUS COMPUTING

Nowadays, most citizens interact with a variety of IT systems at work and home. However, most of them are unaware of the variety and the amount of IT systems that they use every day. Concurrently, more and more of these IT systems are getting interconnected. These intelligent environments developed as communicating IT systems could be described as so-called "Ubiquitous Systems". This trend aims to make many computers available throughout the physical environment, while making them invisible to the user. This is the Third Wave of computing.

Most areas of computer science research, such as programming language implementation or distributed operating system design, are defined largely by technical problems, and driven by building upon and elaborating a body of past results. Ubiquitous computing, by contrast, encompasses a wide range of disparate technological areas brought together by a focus upon a common vision. It is considered as the conjunction of embedded and mobile devices and systems, wireless communications, and distributed, mobile and context-aware computing [7]. It is driven, then, not so much by the problems of the past but by the possibilities of the future[8].

1. DEFINITION OF UBIQUITOUS COMPUTING

Ubiquitous computing is the method of improving computer use by integrating computation into physical environment, but making them effectively invisible to the user[2,3].

Under the term "ubiquitous computing", we understand the ubiquity of information technology and computer power, which in principle pervade all everyday objects. By other words, Computer power and information technology can thus be applied in many areas ranging from industrial production up to private, everyday life with a new quality [9].

Ubiquitous computing suggests countless very small, wirelessly intercommunicating microprocessors, more or less invisibly embedded into objects. Such objects have a new, additional quality "awareness". For example, location of other things in the neighborhood and what happened to them in the past [10]. Moreover, Ubicomp systems aim for a heterogeneous set of devices: personal devices such as laptops, public devices, mobile devices such as smart phones and very large devices such as wall-sized displays and everyday objects such as cars and furniture. All these devices have different operating systems, networking interfaces, input capabilities, and displays [11].

In the long term, ubiquitous computing can pervade all spheres of life: it promises to increase comfort in the private home area and to improve energy efficiency; “intelligent” vehicles may make roads safer; adaptive personal assistance systems could raise work productivity in the office; and in the medical field, implantable sensors and microcomputers monitor the health of the user [10].

The property of ubiquity is reflected in a large number of concepts, such as “pervasive computing”, “ambient intelligence” and “the Internet of things”. In practice almost identical, the difference between these terms is of rather an academic nature [9, 12]. Common to all is the goal of assisting people as well as a continuous optimization and promotion of economic and social processes by numerous microprocessors and sensors integrated into the environment.

2. ASPECTS AND CHALLENGES

2.1. Intelligent infrastructure

With the multitude of requirement especially “invisibility” and “interconnectivity” involved to make ubiquitous computing a reality, the infrastructural requirements are more challenging than the previous ones. The networking or communication infrastructure will be ahead of infrastructural needs. Communicating devices will need ways to identify themselves and describe their behavior to others in the network [13].

The software infrastructure for running ubiquitous applications must be capable of finding, adapting and delivering the appropriate applications to the user depending on a user’s context [14]. and other factors. The software infrastructure will also have to support many applications with various needs depending on whether it is for private or public use.

2.2. Context awareness

As seen before, ubiquitous computing systems have to be context-aware to become invisible. They have to be cognizant of the user’s state and surroundings, and have the ability to modify their behavior based on this information.

Many applications draw context from location, it is generally approved that it consists of more than just information about location. the context can be defined as the answer of the five W’s (who, what, where, when and why) [15]. Another general definition of context is any information that can be used to characterize the situation of an entity data or knowledge collectable by sensors and inputs [16]. Context includes information from the person (physiological state), the sensed environment (environmental state), and computational environment (computational state) that can be provided to alter an applications behavior. Moreover, It can be mined and inferred [1].

The main challenge in this area is the problem of how to collect all the information that is required to and making the whole picture. Another challenge related to context is how to represent it within the system and how to operate all the collected information to provide the right scenarios.

2.3. Resource-Constraints

Another challenge in Ubicomp applications and systems is that they involve devices with limited resources. A wide range of new devices are built and introduced, which are much more resource-constrained. Mobile devices such as PDAs, mobile phones, and music players have to be smaller to allow users to move conveniently around with them. This size constraint limits other resources such as screen size, processing power, network connectivity and battery life, compared to a standard PC, which in turn influences other factors like connectivity and the development of services and applications. In addition, embedded platforms such as sensor networks are also very limited by energy. Therefore, when creating systems support in an Ubicomp setting, it is important to recognize the constraints of the target devices [11, 13].

2.4. Privacy and trust

Collecting more context information would result in more intelligent and proactive systems; it also results in the continuous monitoring of a user's activities. Ubicomp fundamentally alters privacy by creating continuous detailed data flows. This growing use of private information and intelligent communications make the concern of Privacy more important than classic networks. particularly acute the case of home-based Ubicomp and health care domain where vulnerable populations risk enforced technological intimacy [17].

The legal challenge is the self-determining protection of user's information from the operators of UbiComp applications. In addition, questions may arise from the expected utilization of autonomous information systems under private law. The question of the correct regulatory instruments also arises in addition to the substantive requirements of specific regulations to ensure the best possible balance of the actors interests [9].

2.5. User interfaces

In order to be able to interact with invisible, embedded information systems, innovative user interfaces are necessary which permit a natural interaction like speech or physical interaction. The new type of interaction includes also the automatic capture of the context, which is not just about the registration of external parameters but increasingly also to identify the user's emotional states or his intended actions. Only with the most accurate knowledge of each context is it possible to offer services in response to individual locations and situations and to delegate certain tasks completely to technology [9, 18]. All these issues and challenges require that we come up with new methods and tools to help us design new user interfaces that are appropriate and take advantage of the possibilities of the ubiquitous computing paradigm [13].

2.6. Experience and social impact

The development of everyday computing aims to change relationship between human and computers by supporting informal and unstructured activities: As computing moves beyond the confines of tool usage, leaving the solitary desktop platform intending a large penetration of the environment around us. By the way of migration from interaction between the individual and a single device to a profusion of networked mobile and embedded computing devices that individuals and groups employ across a variety of settings creating new forms of interaction, as well as social norms [13, 15].

Evaluate this kind of systems is difficult. The first major difficulty in evaluating a Ubicomp system is simply having a reliable system to evaluate. Deeper evaluation results require real use of a system and a deployment into an authentic setting. That's make another concern the social issues.

The social implications of these technologies will often come after people invent new, unforeseen, uses of these technologies. Moreover, these systems can have both positive and negative impact beside their goal.

3. UBIQUITOUS USER INTERFACE

3.1. Caracteristics

To reach the goals of Ubiquitous User Interfaces and ensure the totality of their characteristics, Developers of UUI should consider these rules from the start of the ideation process [4]:

- Bliss: Learning to interact with a new UUI should not require people to learn another skill or complex command language;

- Distraction: Do not demand constant attention in a UUI. Inattention is the norm not the exception;
- Cognitive Flow: Ubicomp systems that are everywhere must allow the user to retain total focus on the task at hand ;
- No Manuals: Do not require a user to read a manual to learn how to operate the current UUI. sufficiency of learning by experience;
- Transparency: Do not rely on users to hold application state in the mind to operate the UUI.
- Modelessness: Avoid "modes" where the system responds differently to the same input stimulus dependent on some hidden state information;
- Fearlessness of Interaction: Provide easy means to undo actions, otherwise users may become paralyzed with fear when using the UUI;
- Notifications: Feedback to the user can be piggybacked and layered into interactions with their physical environment;
- Calming: Interfaces will support situated actions, interfaces will rely on a wide array of human inputs and human senses;
- Defaults: Good interfaces judiciously exploit what the system knows or can deduce.

3.2. Classes of UUI

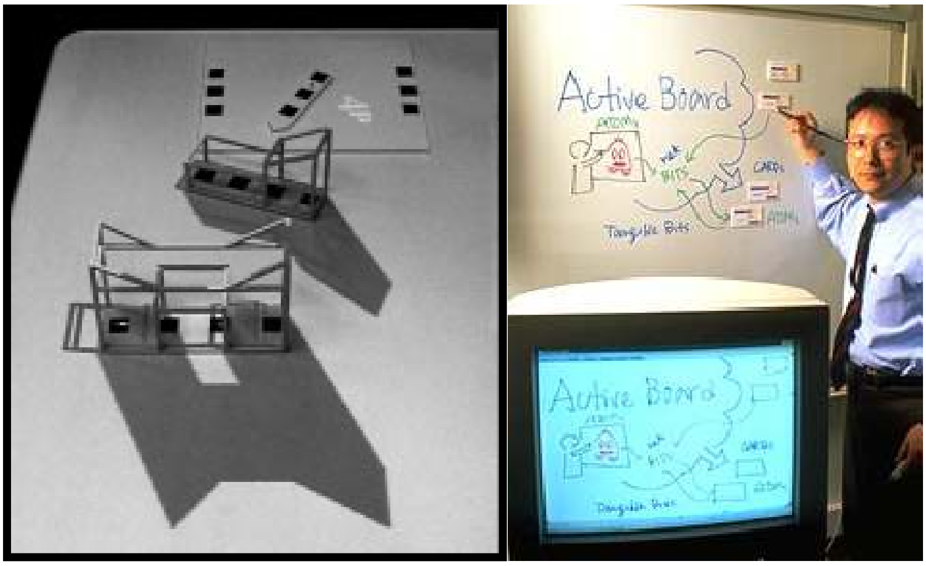
By the way, of trying to reach the ubiquity paradigm, new classes of interfaces have been developed far surpassing the traditional well-known interfaces. We consider some as very useful to the development of Ubiquitous systems, the three main classes are Tangible Interfaces (TUI), Ambient interfaces (AUI) and Surface interfaces (SUI) [4]. Nearby these classes, there are some alternative categories of interfaces gestural user interfaces [19], touch-based user interfaces, pen-based user interfaces, exertion interfaces [20], Augmented Reality user interfaces and Multimodal ones.

3.2.1. Tangible Interfaces

Tangible User Interfaces (TUI) was initially named Graspable User Interface; this term is no longer used. Tangible User Interfaces couple physical representations (e.g. spatially manipulable physical objects) with digital representations, TUIs augment the real physical world by coupling digital information to everyday physical objects and environments [21, 22].

The TUI give a physical form to computational artifacts and digital information, the user manipulation is then detected by the system, which responds, through the physical artifact in form of a spatial or haptic feedback. Such physical interactions are, so, very natural and intuitive. That makes the TUIs having both input and output factored into the same physical artifact [4].

One of the most ambitious projects is “Urp” (fig. 2), one of the pioneers of TUI, it is a tangible workbench for urban planning and design. In Urp, the basic objects are physical architectural models (houses, shops, schools) that can be placed by hand onto the workbench [23].



(a) (b)

Fig. 2. (a) Urp: A Workbench for Urban Planning and Design [23]. (b) transBOARD: a networked digital whiteboard with hyperCARDs [21].

The transBOARD is a networked digitally-enhanced physical whiteboard designed to explore the concept of interactive surfaces which absorb information from the physical world, transforming this data into bits and distributing it into cyberspace. The transBOARD supports the use of "hyperCARDs" (barcode-tagged paper cards) as containers of digital strokes. These magnetically-backed cards, which can be attached to the vertical surface of the transBOARD[21].

3.2.2. Ambient user interfaces

Ambient user interfaces have the characteristic to operate in the periphery of the users and should remain unnoticed until required, exactly as does the mobile phone. The inputs should then come from nonintrusive sensing or inference from other actions. Ambient interfaces use the whole environment of the user for the interaction between the user and the system. They present digital information through subtle changes in the user's physical environment such as variations of light, sounds, or movements. The AUIs conforms closest to Weiser's vision of calm technology: they introduce computation in basic objects, environments and the activities of our everyday lives in such a way that no one will notice its presence [4, 24].

Fig. 3. Datafountain: Money Translated to Water [25].

An example of ambient user interfaces the Data fountain, it displays relative currency rates using three water fountains side by side. It is connected to money currency rates on the internet and the relative heights of each fountain change with respect to the currency changes [25].

The ambientROOM is a Tangible Bits platform which explores the use of ambient media as a means of communicating information at the periphery of human perception. It allows users to be aware of background bits using ambient display media such as ambient light, shadow, sound, airflow, water movement in an augmented architectural space. The ambientROOM also provides physical handles such as bottles and a clock to control ambient display of bits. The ambientROOM opens new possibilities of exploring seamless transitions between background awareness and foreground activity [26].

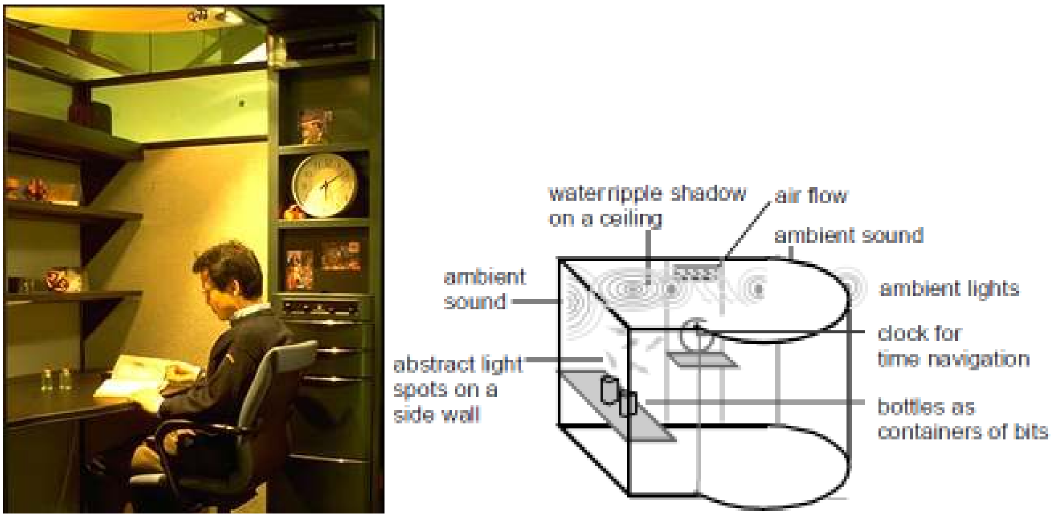


Fig. 4. The ambientROOM [26].

3.2.3. Surface user interfaces

Surface user interfaces are a class of user interfaces that relie on a self-illuminated or projected horizontal, vertical, or spherical interactive surface coupled with control of computation into the same physical surface. They can display information and users can interact with this information by the same surface either directly with their hands, using a stylus, or using some form of wearable hardware and a separate keyboard and mouse cannot or should not be used. SUIs are often used in public places or as small personal devices. Depending on their size, they can be ranged from small personal devices like PDA, medium as Tablet PC to largest ones similar to the interactive surfaces found in public places [4, 27].

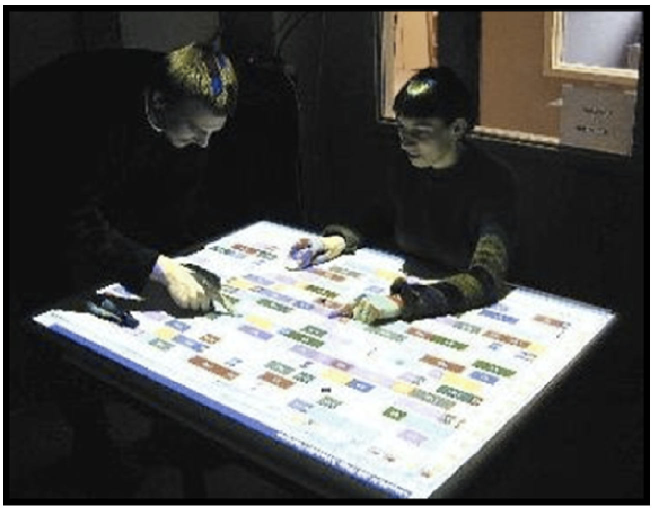


Fig. 5. Two users interactiong with a DiamondTouch: collaborative work environment [28].

An example of Surface user interfaces is DiamondTouch; it is a multi-user touch technology for tabletop front-projected displays. It enables several different people to use the same touch-surface simultaneously without interfering with each other, or being affected by foreign objects. It also allows the computer to identify which person is touching where [28].

An interactive whiteboard at a low-cost set up with a Kinect infrared sensor and common projector is a large interactive display that connects to a computer and projector. A projector projects the computer's desktop onto the board's surface where users control the computer using a pen, finger, stylus, or other device. The board is typically mounted to a wall or floor stand [29].

Fig. 6. Low-Cost Interactive Whiteboard Using the Kinect [29]

3.2.4. Augmented Reality user interfaces

Augmented reality is related to the concept of virtual reality (VR). VR attempts to create an artificial world that a person can experience and explore interactively, predominantly through his or her sense of vision, but also via audio, tactile, and other forms of feedback. AR also brings about an interactive experience, but aims to supplement the real world, rather than creating an entirely artificial environment. The physical objects in the individual's become the backdrop and target items for computer-generated annotations.

(a) (b)

Fig. 7. Mobile AR restaurant guide. (a) User with MARS backpack, looking at a restaurant. (b) Annotated view of restaurant, imaged through the head-worn display.

An example of Augmented Reality user interfaces is an early prototype of an outdoor MARS that presents 3D graphical tour guide information to campus visitors; registered with the buildings and artifacts the visitor sees [30].

Virtual Cable created by a company called Making Virtual Solid (MVS – California), is an augmented reality system, which projects images from car's GPS onto the windshield to create a navigation system that can be used easily. It is made to appear like a physical line in the sky, outside the car, that shows the path to the chosen destination [31].

Fig. 8. Virtual Cable augmented outside a car.

3.2.5. Multimodal user interfaces

Multimodal user interfaces provide the user with multiple modes of interfacing with a system. It provides several distinct tools for input and output of data. A modality may be the particular form used for rendering a thought, the way an action is performed, such as speech, pen, touch and manual gestures.

Fig. 9. Multimodal shopping environment [32].

A multimodal system must be equipped with hardware to acquire and render multimodal expressions in real time, that is, with a response time compatible with the user's expectations. It must be able to choose the appropriate modality for outputs. It must be able to understand multimodal input expressions [32].

3.3. Input and output technologies

3.3.1. Input

The downfall of IT costs helped to create new interfaces that rely on new input modalities, which require intensive computing such as gesture or voice recognition as well as various sensors [4] or inferred data from mining, historical data and user preferences.

Sensors are largely used in ubiquitous systems, they can be placed anywhere in the environment; they provide contextual data to the system.

In ubiquitous systems, a gesture input is a significant movement of the body or one of its parts like hand or head. Walking or manipulating physical artifacts is considered as a gestural input.

Voice input is resulting from the voice recognition in which a process interprets human speech that audio inputs provide. The method is really promising because it permits invisible but omnipresent computing. Used as well as an output, it forms a natural way to interact with ubiquitous systems. The voice input can also give additional information such as individual’s emotional state conveyed by the voice tone.

3.3.2. Output

Output modalities are also technologically advanced over the time, the classic types of output such text, graphics, audio, and video still suitable in ubiquitous computing. As well as haptics, ambient displays, environmental updates, actuators, automated actions and personalized behaviors.

Haptics or tactile feedback technology takes advantage of the sense of touch by applying forces, vibrations, or motions to the user.

Ambient display allows people to monitor information in a peripheral manner. Such outputs are intended to be ignorable but allowing users to perceive the information when required.

An actuator is a device responsible to control the starting and stopping mechanisms of moving system. It takes source of power and converts the energy to the motion.

Visual displays still the most common type of output device for interfacing to people.

3.4. Cycle of life

In the last decade, many successful Ubicomp projects was realized and embedded into real-world contexts, where the intensity of the interactions between a user's personal computing space and his surroundings increases [33]. These projects have made different efforts to address various issues of ubiquitous computing and have become the communal supports in making a reality of the invisible computing.

This goal brings new challenges to the design and evaluation of such interfaces and systems. The developers have to construct the complete scenarios, in order to understand the everyday practices of their users. However, the everyday situations are continuously changing, due to the heterogeneous contexts of use and the settings for interaction; it is very difficult to predict how users will react when designing an interactive service. Currently, the technical features for building such systems are visible, due to the existing sensing, data processing and context-recognizing technologies but there is still a lack in the criteria for designing and evaluating the features of the application itself [34].

Besides, many problems manifest in infrastructure costs, deployment difficulties, practitioner training needs and uptake difficulties. In order to explore these issues for ubiquitous computing, it is necessary to consider both the practitioner interactions, practices and capabilities (such as the ability of practitioners to integrate devices into their practices), and the characteristics of technology and its underlying infrastructure. This suggests that rather than develop a complete concept for a ubiquitous computing environment and then build a research prototype that is removed from a work practice context, a participatory bootstrapping approach is needed. In such an approach designers reveal the capabilities and characteristics of technology and technical infrastructure to practitioners in intelligible ways (practitioners often find the details of infrastructure both mundane and baffling) and practitioners try out and explore the possibilities for such technologies to enhance their practices in the context of their work environments [35].

3.4.1. Design

The determination of features in a design today largely depends on designers experiences and intuitions or on the specific styles supported by the particular infrastructure. Many works agree that Interaction designers make technology, particularly digital technology, useful, usable, and pleasurable to use [36]. They revolve around facilitating communication between humans through products (The Social Interaction Design View); and they define the behavior, functionality and feedback of artifacts, environments, and systems (The Behaviorist View) [37].

When considering the various aspects of interaction design for Ubicomp, it is important to recall the breadth of the technological scenarios envisaged. Often, these scenarios revolve around people making connections to other people through Ubicomp systems, not just connecting to the system itself [4]. Good interaction design can be achieved in a number of ways depending on the complexity of the system proposed, its novelty, its degree of stability or ubiquity, and its cost. Design methodologies of interest to Ubicomp include User-centered design (UCD),

User-centered design (UCD) focuses on the user’s needs, problems, and goals to guide the design of a system. Users are involved at every stage of the process. To help that the system developed actually meets their needs and allows them to achieve their goals. The applications can be refined by the repeated trips around a design cycle: first imagining it (design), then realizing it physically (implementation) and afterward testing it (evaluation).

However, the unique requirements (such as embracing the contextual changes, the implicit interaction and localized scalability) make the existing approaches in UCD ill-suited for pervasive applications. For example, a tighter coupling between applications and the contexts from ‘‘non-standard’’ input devices will make the traditional design approaches ineffective in some cases. In UCD, a major problem is how to determine the desired functions and technical features from witch are almost unexplored [34]. Therefore, UCD process is more adapted when the system just solves an existing and assumed problem but relies on new techniques, methods, or infrastructure.

Systems design is a systematic and compositional approach to development, based on the combination of components to realize a solution in essence the development of a system of systems. A UBICOMP system is typically composed of many systems including social systems (people), devices, appliances, computational artifacts, sensors, actuators, and services. A systems design approach forces a designer to consider the entire environment in which the UBICOMP system will be realized and not just one component of it. As an analytical approach, it requires careful modeling and understanding of the implications of the user's goals, feedback, and controls provided [4, 37].

Systems design approach can be used when the system provides a new way to interact, relying on a novel combination of modalities with new appliances.

Genius design is the process relying exclusively on the wisdom and experience of the designer to drives the design process. Based on their most informed opinion as to what users want and need, they design a product using their instinct, skill and knowledge. that are used to make successful products. We often tout the iPhone or indeed any Apple product as a case for Genius design, Apple is notorious for maintaining secrecy around unreleased products, so they don't do user testing, user surveys, or any of the other activities normally associated with user centered design. Users may be involved at the end for testing or usability testing. a quality user experience can be created with genius design. However, the risks of failure still more important with this approach [37].

Attempts to invent the future can be curtailed by the current generation of users who cannot imagine a future where this new device, service, or interface will be cheap, desired, or even required.

UCD, systems design, and genius design all have their strengths and weaknesses. Current generation of users who cannot imagine a future. However, this does not mean that the current generation of users should be ignored when researching and developing user interfaces for UBICOMP systems. approach to help bridge between these methods can form a basis for exploiting lessons learned in each approach in the design of new systems.

It is believed that a focus on design and the activities associated with the design process is an important issue and is essential to the success of a product.

3.4.2. Evaluation

Good design only has meaning for pervasive applications when testing it with the real users in their everyday lives over extended periods.

As mentioned before, a good design has meaning only for Ubiquitous applications when testing it with the real users in their everyday lives over extended periods. There have been efforts discussed how to adapt and apply the traditional methods [34]. Typically, computer scientists design infrastructure based on a desire for reuse [15], and evaluate infrastructure based on "classical" technical criteria: performance, scalability, security, robustness, and so on. These are all crucial metrics, and they must be accounted for if we wish to build workable systems. But there is a distinction between technical workability and value for end-users.

The criteria of usability and usefulness are often used to assess the feasibility and user-acceptance of an application prototype, but they are difficult to be measured due to the ambiguous definitions. These evaluation criteria should be field specific and distinctive in each phase. For example, easy-to-modify is used to early test the power of the proof-of-concept prototypes; easy-to-understand is useful for testing the experience prototypes to obtain the feedback; the latency is for testing the function prototypes of context-aware applications [34, 38].

Usability is a quality attribute that assesses how easy user interfaces are to use. For a UUI, [4] described a seven key interface usability metrics [39]:

- Conciseness: The actions, the interfaces and the terminology should be kept consistent, while allowing a limited number of exceptions for specific situations (e.g. asking for confirmation when a delete command is issued);
- Expressiveness: Your design should organize the user interface purposefully, in meaningful and useful ways based on clear models that are apparent and recognizable to users, putting related things together and separating unrelated things. The structure principle is concerned with your overall user interface architecture;
- Ease: design should make simple, common tasks simple to do, communicating clearly and simply in the user's own language, and providing good shortcuts that are meaningfully related to longer procedures;
- Transparency: the system after change adheres to previous external interface as much as possible while changing its internal behavior. The purpose is to protect from change all systems (or human users) on the other end of the interface;

- Discoverability: design should keep all needed options and materials for a given task visible without distracting the user with extraneous or redundant information;
- Invisibility enhancing computer use by making many computers available throughout the physical environment, but making them effectively invisible to the user;
- Programmability: Ability to be trained to do something or to react automatically.

Moreover, measuring usability issues related to efficiency and effectiveness in performance are not the only to consider, but also evaluate the aesthetical design of the system, since this aspect has an impact of how users perceive the usability of the system [40].

4. CONCLUSION

As a research field, Ubicomp still have to build and evaluate systems components to ease the design, implementation, deployment, and maintenance of real-world Ubicomp applications.

This chapter has outlined challenges related to Ubicomp research and Ubiquitous user interfaces development. Many user interfaces classes are considered promising for Ubicomp future. Camera-projector based user interfaces is one of the most suitable. The next chapter is dedicated for that kind of user interfaces.

CHAPTER II. CAMERA PROJECTOR BASED USER INTERFACE (CPUI)

When a projector is combined with computer vision, any surface in any environment can be turned into an interactive interface, without having to modify or wire the surface. Projected displays offer rich opportunities and pose new challenges for interaction based on gesture recognition. An HCI (Human-Computer Interface) with small device size, large display, augmented reality, and touch input facility can be made possible by a projector and camera[41].

In this chapter we will present the main lines and the background that makes the CPUIs reality, and the main needs

Fig. 10. (Left) Wearable Multitouch Interaction Everywhere[42]. (Right) LuminAR: Portable Robotic Augmented Reality.[76]

The use of camera devices as a tracking input allow advanced and natural interaction ways in the same time, all sort of gestures can be used (full body movement [41], hand waving, pointing, opening and closing hand.. etc), in addition to that, the development and the verity of imagery captors can get a lot more information about the user, such as distance, heat, shape, etc.; all is about the way we take advantage of the devices.

Projector-camera systems represent an effective approach towards constructing an indoor environment with variant display capabilities without massive equipment resources. Mouse, keyboard and monitor build up traditional input and visual feedback loop for desktop computer systems. New and suitable human-computer connection are being discussed and explored for new spatial relationship between projected display and user in a larger indoor area.

It offers an intuitive interface that is capable to access and manipulate a huge data space in an efficient and pleasant manner.

All those advantages makes the Camera projector based user interfaces easy to integrate a large number of environments and cover many tasks, from daily urban use (as wearable interactive device [42] for example) to office and industrial design [43] or learning systems[44].

The next figure (Fig 11) represents different kinds of CPIUs and their positioning relatively to its dependency on vision (cameras) and the surface (projected or not).

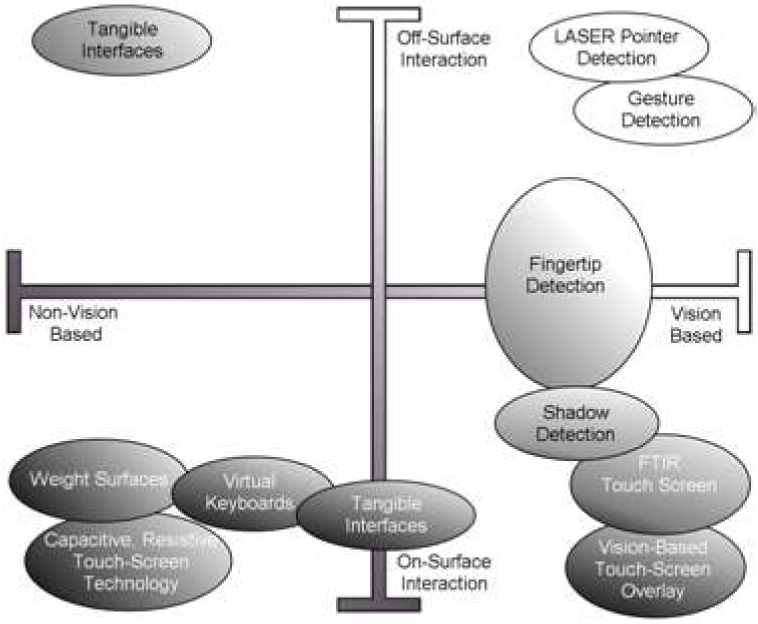


Fig. 11. Interaction Techniques for Projected Displays[45].

1. COMPUTER VISION-BASED USER INTERFACES

Cameras have been invented very long time ago, these optical instruments that records images took much time to be what they are now. Cameras now do not aim just to make memorable moments; images captured are used to get information. Current technology aim to allow to machines to see.

1.1. Computer vision

Ying Wu [47] gave a great and concise definition of the computer vision and its situation in the total computing domain, most of the following introduction is cited from his work.

“Computer vision is the science and technology of machines that see.”[48]

It was a dream that computers would be able to see and think, which has been driving us to explore various research issues to make this dream come true. Although computers become faster and faster, they are still quite dull, since they can neither see nor even perform simple reasoning's. Obviously, we are not satisfied to just use our computers as a calculator, a word processer, a CD player, or a game station; instead, we expect computers to do more intelligent things like our human beings. For example,

- Can computers identify me by looking at my face or even my gait?
- Can computers know where I am looking at and what I am doing?
- Can computers tell what is a car and what is not a car?
- Can computers learn something by themselves?
- Can computers summarize a video for me?
- …

Computer vision involves fundamental research in image processing, computer vision/graphics, machine learning, pattern recognition, biomechanics and even psychology. On top of it are several major application areas such as human-computer interaction, robotics, virtual environments, and multimedia.

The current search trend in computer vision is currently pursuing some intelligent ways by machine learning and pattern recognition, trying to achieve a kind of artificial intelligence, instead of taking some ad hoc approaches to audio and visual processing when the area was in its infantile stage.

Searchers have expected in the coming generation a disappearance of mouse and keyboards, and a rapid progress in the near future in such areas as intelligent human-computer interaction, robotics, virtual environments, intelligent environments, and multimedia.

Computers could understand our actions and our languages, they could think and feedback to us some kind of smart results in response to our commands, and they could even perform some missions on behalf of our human beings, Ying Wu says.

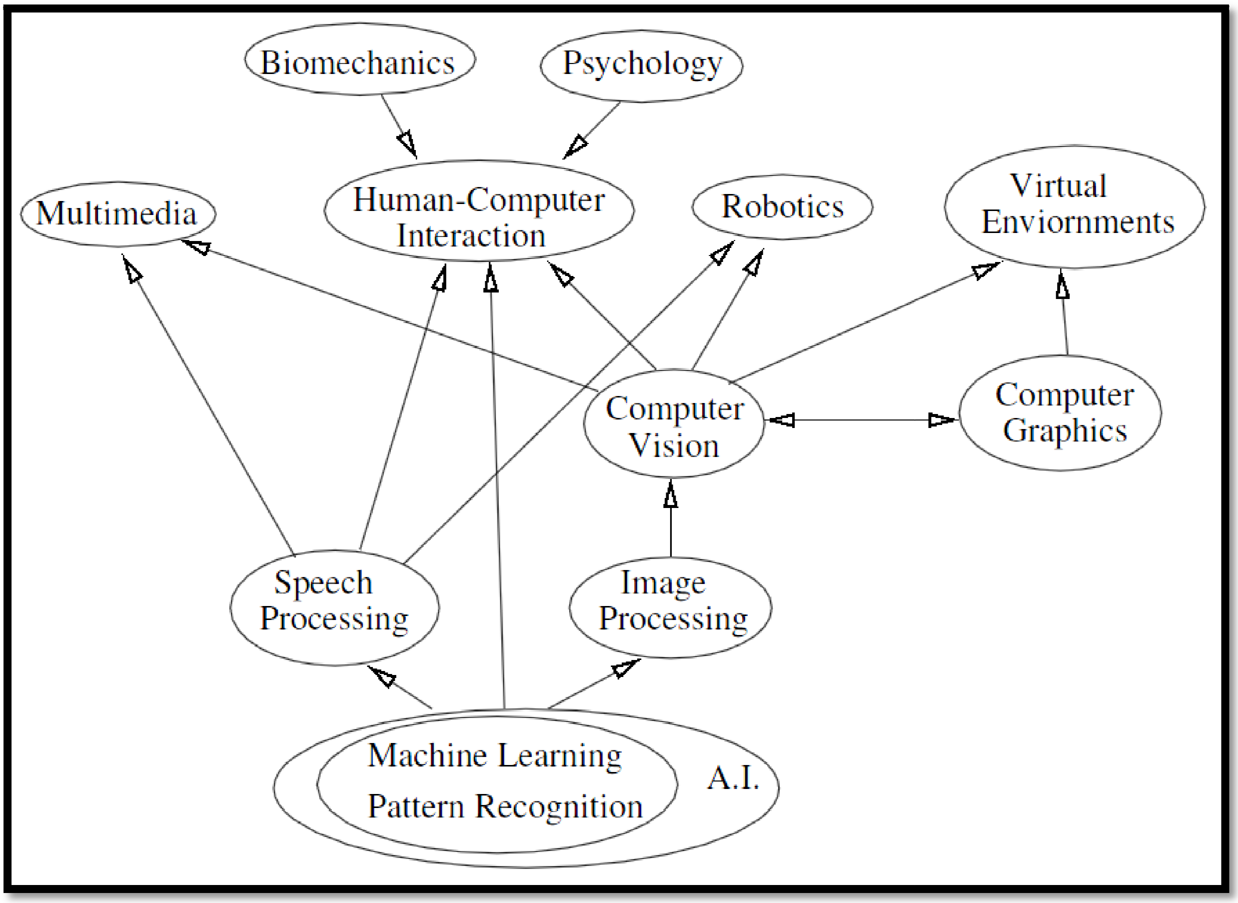


Fig. 12. The area that involves Computer vision [47,p3]

The fundamental research in image processing, computer vision, machine learning and pattern recognition is important part of the foundation of these application topics.

Image processing is a quite board research area, not just filtering, compression, and enhancement. Besides, we are even interested in the question, "what is in images?". i.e., content analysis of visual inputs, which is part of the main task of computer vision. The study of computer vision could make possible such tasks as 3D reconstruction of scenes, motion capturing, and object recognition, which are crucial for even higher-level intelligence such as image and video understanding, and motion understanding.

Vision perception itself is an intelligent process, not just an imaging process. Through vision, human beings are able to perceive the lighting, color, texture, shape and motion of the outside world. The intelligence lies in the inference of such high-level concepts based on imaging. It is quite easy for human beings, but it is still very unclear how computers can achieve that level of intelligence. Recognition is one of the most fundamental problems for machine, i.e., recognizing a pre-stored pattern in new situations by comparing inputs with a set of templates or models. However, the problem is how to construct these templates or models. For example, what will be the appropriate templates to recognize faces even under different view directions or different lightings? The most challenging aspect for visual recognition lies in the fact that there are too many aspects that affect imaging, and it is impossible to model every aspect such as lighting and motion. So, people ask, “can computers ‘learn’ the model from examples?" such that models could be learned implicitly, instead of constructed explicitly.

Computer vision, basically, is to infer different factors such as camera model, lighting, color, texture, shape and motion that affect images and videos, from visual inputs. A rough structure of machine vision could be illustrated by the next figure.

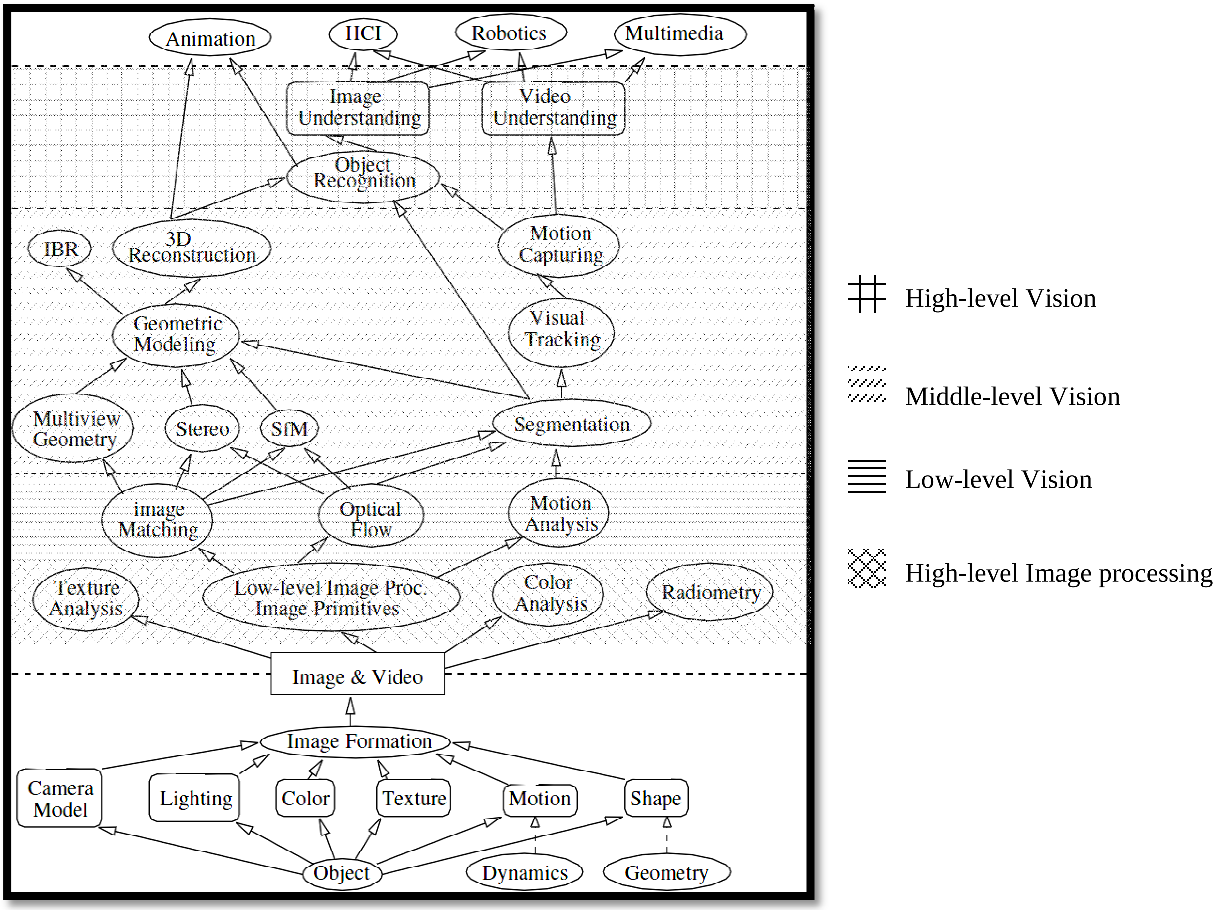


Fig. 13. What is computer vision [47, p5]

In a word, computer vision is an inverse processing of the forward process of image formation and graphics. In this sense, as many people agree, vision is a much more challenging problem than computer graphics, because it is full of uncertainties.

1.2. Image Formation

Image formation studies the forward process of producing images and videos. It is an important research topic for both vision and graphics. To produce a real image, the nature of the visual sensors, i.e., cameras should be studied.

In terms of geometrical aspects of camera, we have Pinhole cameras (restrict angle), Cameras with lenses and Omnidirectional cameras (wide angle).

In terms of physical aspects we study factors such as focal lengthes and dynamic ranges of CCD and CMOS cameras.

Besides the imaging device, it is also important to study the factors from objects and scenes themselves, such as lighting, color, texture, motion and shape, which largely affect the appearance of images and video.

As shown in Fig 14 Computer vision levels are: (1) low-level Image Processing, (2) low-level Vision, (3) middle-level vision, and (4) high-level vision.

Low-level Image Processing

Low-level image processing is not vision, but the pre-processing steps for vision. The basic task is to extract fundamental image primitives for further processing, it includes:

- Edge detection,
- Corner detection,
- Filtering,
- Morphology,
- Etc... .

Low-level Vision

Low-level vision tasks can be preformed based on low-level image processing, in this level that we have:

- Image matching : correspondences between two or more images (same scene taken from different viewpoints, moving scene taken by a fixed camera, or both)
- Optical flow: a kind of image observation of motion, but it is not the true motion. Since it only measures the optical changes in images.
- Computation and motion analysis.

- Constructing image correspondences is a fundamentally important problem in vision for both geometry recovery and motion recovery. Based on optical flows, camera motion or object motion could be estimated.

Middle-level Vision

There are two major aspects in middle-level vision: (1) Inferring the geometry, and (2) Inferring the motion. These two aspects are not independent but highly related. Algtought, to estimate geometry we need at least two images. They could be taken from two cameras or come from the motion of the scene.

Some fundamental parts of geometric vision include multi-view geometry, stereo and structure from motion (SfM), which fulfill the step of from 2D to 3D by inferring 3D scene information from 2D images. Based on that, geometric modeling is to construct 3D models for 6 objects and scenes, such that 3D reconstruction and image-based rendering could be made possible.

Other approaches for geometry detection have been involved, generally using an infra-red (IR) cameras associated to an IR light source:

- Using structured light: or Sheet of light triangulation Structured light is the projection of a light pattern (ray, plane, grid, encoded light, and so forth) under calibrated geometric conditions onto an object whose shape needs to be recovered [49].
- Using time-of-flight cameras: (ToF camera) is a range imaging camera system that resolves distance based on the known speed of light, measuring the time-of-flight of a light signal between the camera and the subject for each point of the image. The time-of-flight camera is a class of scanner less LIDAR, in which the entire scene is captured with each laser or light pulse, as opposed to point-by-point with a laser beam such as in scanning LIDAR systems [50]. Time-of-flight camera products for civil applications are still emerging, as a recent example the Microsoft XBox One uses a time-of-flight camera in its new Kinect sensor to acquire RGB-Depth data [51].
- Multi-Flash Camera [53].

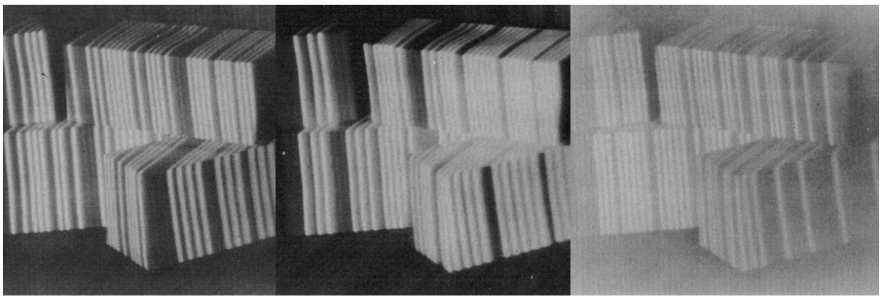


Fig. 14. filtered monochrome images (color separations) madefrom the four box scene (from left to right) Red,Green,Blue [49].

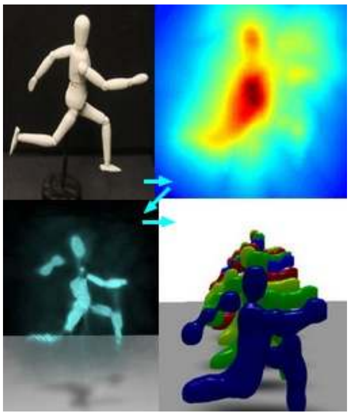


Fig. 15. Complex object fast reconstruction using ToF camera [52]

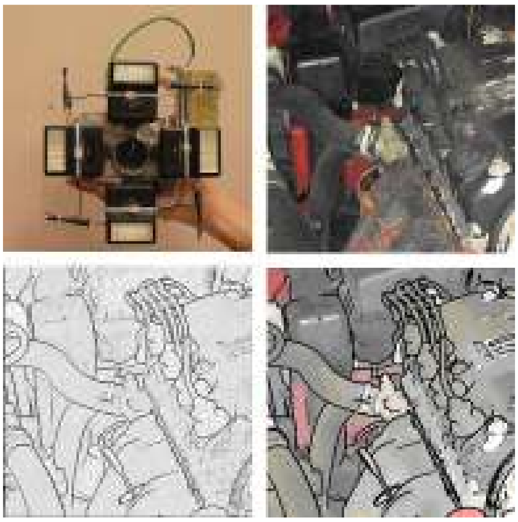


Fig. 16. shape detection using multiflash camera.[53]

Another task of middle-level vision is to answer the question how the object moves?

We should know which areas in the images belong to the object, which is the task of image segmentation. Image segmentation has been a challenging fundamental problem in computer vision for decades. Segmentation could be based on spatial similarities and continuities. When considering motion continuities, the uncertainty of segmentation can be alleviated. On top of that is visual tracking and visual motion capturing, which estimate 2D and 3D motions, including deformable motions and articulated motions.

High-level Vision

High-level vision is to infer the semantics, for example, object recognition and scene understanding. A challenging question in many decades is that how to achieve invariant recognition, i.e., recognize 3D object from different view directions.

There have been two approaches for recognition:

- Model-based recognition,
- Learning-based recognition.

It is noticed that there was a spiral development of these two approaches in history. Even higher level vision is image understanding and video understanding.

We are interested in answering questions like "Is there a car in the image?" or "Is this video a drama or an action?", or "Is the person in the video jumping?" Based on the answers of these questions, we should be able to fulfill different tasks in intelligent human-computer interaction, intelligent robots, smart environment and content-based multimedia.

1.3. Vision based user interface

In their book The Media Equation, Reeves and Nass [54] argue that people tend to equate media and real life. They performed a number of studies testing a broad range of social and natural experiences, with media taking the place of real people and places, and found that: "Individuals interactions with computers, television, and new media are fundamentally social and natural, just like interactions in real life" [54].

For example, people are polite to computers and display emotional reactions to technology.

In other hand Vision is clearly an important element of human-human communication. Although we can communicate without it, people still tend to spend endless hours travelling in order to meet face to face. All because there is a richness of communication that cannot be matched using only voice or text. Body language such as facial expressions, silent nods and other gestures add personality, trust, and important information in human-to-human dialog. We expect it can do the same in human-computer interaction.

Vision based interfaces (VBI) is a subfield of perceptual user interfaces which concentrates on developing visual awareness of people.

VBI interaction can be categorized into two aspects:

- Control is explicit communication to the system – e.g., put that object there.
- Awareness, picking up information about the subject without an explicit attempt to communicate, gives context to an application (or to a UUI). The system may or may not change its behavior based on this information. For example, a system may decide to stop all unnecessary background processes when it sees me enter the room, not because of an explicit command I issues, but because of a change in its context.

1.4. Computer vision in UUI

Current computer interfaces have little or no concept of awareness. While many research efforts emphasize VBI for control, it is likely that VBI for awareness is being useful for UUIs.

The awareness that we can get from VBIs is not the only advantage that can serve UUIs, in fact their:

- Non-pervasive aspect: because can be set in a room or a street without being noticed, and can be used without having to do a special act contrarily to grabbing the mouse or positioning hands over a keyboard to start using it.
- And the ability to lead to a truly natural way of user control, based on natural movements, such as pointing, waving, turning, standing, walking, etc...

Are also main qualities that make VIBs very suitable UIs to be a part of a UUI.

Anyway, VBIs alone are not satisfactory to create a UUI, adequate displays have to be implicated in order to create a complete and homorganic UI where the users not only uses, but get served too.

2. PROJECTOR BASED USER INTERFACE

2.1. Presentation of projection

A video projector is an image projector that receives a video signal and projects the corresponding image on a projection screen using a lens system. All video projectors use a very bright light to project the image, and most modern ones can correct any curves, blurriness, and other inconsistencies through manual settings.

Video projectors are widely used for many applications such as, conference room presentations, classroom training, home theatre and concerts. Projectors are widely used in many schools and other educational settings, sometimes connected to an interactive whiteboard to interactively teach pupils.

Traditionally a video projector, also known as a digital projector, may project onto a reflective projection screen, or it may be built into a cabinet with a translucent rear-projection screen to form a single unified display device.

Judging by the mobility there is three principal kinds of projectors: fixed, mobile, and Steerable projectors.

Fixed projectors are commonly used for home cinema, and conference room, most often have a good display quality projecting to a permanent projection screen, less adopted by projected user interfaces search community than the other types.

Mobile projectors are divided on two categories:

- Business projectors: Most common, portable but still have to be fixed when in use. Generally used for presentation porous in business and educational domains, lot of works on projected UIs has been done using this kind of projectors [41, 45].
- Mini projector or Pico projectors: the now coming generation, are a easily hand-carried projectors that can fit in users hand, also they can be wearable like a pendant to project in the space facing the user [42], they are being integrated to devices with small display capacity to allow them to enhance it [55].

Steerable projectors are a special kind of mobility; it is principally only rotation allowing the projector to display in almost all the surface of a room [56, 57].

Fig. 17. (left) High-end Digital Light Range DL.1 to DL.3 steerable projector (right) Mitsubishi Pocket Projector

These categories can be seen more clearly in Figure 19, when compared to a traditional desktop monitor.

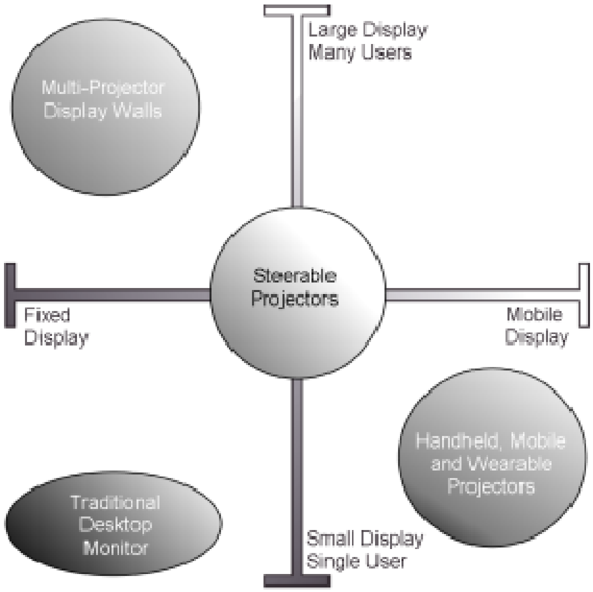


Fig. 18. The Projector system family decomposed by mobility and display size, compared to traditional desktop monitors.[45]

In contrast, projector-camera systems in the third category, with computer controlled steerable mirrors or pan and tilt platforms [59][60] allow a much larger system field of view and the ability to track objects moving in the environment. We use the generic term "Steerable Projectors" for these systems [45].

2.2. Projector based user interface

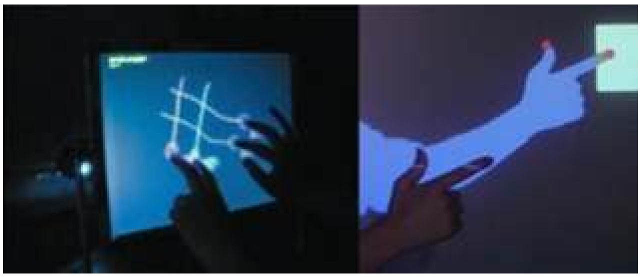


Fig. 19. (Left) Low-Cost Multi-Touch Sensing through Frustrated Total Internal Refl ection[74]. (Right) User Interface by Virtual Shadow Projection[75].

It's believed that projectors are capable of much more. Many researches aimed to embed projectors as part of an infrastructure as they become mobile and networked. By exploiting elements (that will become) available in a self-contained projector such as the camera module, a tilt sensor and wireless connection [61][62].

Further projected user interfaces goes beyond passive projection to show a truly interactive system in which projected information can be navigated and updated. Projected information can enhance a smart infrastructure.

Projected user interface offering virtually unlimited, interaction space for networked devices. On top, it offers an intuitive interface that is capable to access and manipulate a huge data space in an efficient and pleasant manner. A projected interaction space can extend the display capabilities of laptops and objects. The recent availability of small, cheap and bright video projectors makes them practical for augmenting objects with non-invasive displays.

But ordinary surfaces have varying reflectance, color, and geometry. These variations are commonly accounted for by using the camera and applying methods from computer vision making it a flexible way to interact with a system.

As an example of handheld projected UI Raskar et al. developed a concept of handheld projector-camera systems in the [63]. The handheld projector-camera system included onboard computing, a tilt-sensor and network access.

Fig. 20. (left) The iLamps Project developed a handheld projector-camera system, (right) Detecting circular fiducial markers to augment a wall scene with projection

The projector-camera system was calibrated to determine the optical parameters and relative locations and orientations of projector and camera. Projection-based object adaptive displays were then demonstrated using circular fiducial markers to allow the system to calculate the projector position. With projector pose known relative to an object (in this case the black rectangle with a fiducial marker in Figure 21), a projection can be registered with the object so it is overlaid on its surfaces.

Due to recent technology developments in micro displays and cheap, long-life LED and LASER light sources and the reduction in the size of projectors, enabled projectors to be carried in a pocket or embedded in mobile devices. And furthermore, the ubiquity of those devices are allowing focusing on collaborative interaction techniques using multiple handheld projectors [64].

However, multi-projection can face some main issues such as overlapping, light pollution, Privacy, and brightness inequality.

A common use of camera and position sensors to get the projected space proprieties and detect nearby projections in order to avoid overlapping projections [65]. Equipping projected UIs devices with steerable mirror can greatly extend their sensing and projection abilities. The camera can also detect human faces, and enable the projector to darken the face regions in the projection, so as to avoid accidentally shining passer-bys in the eyes.

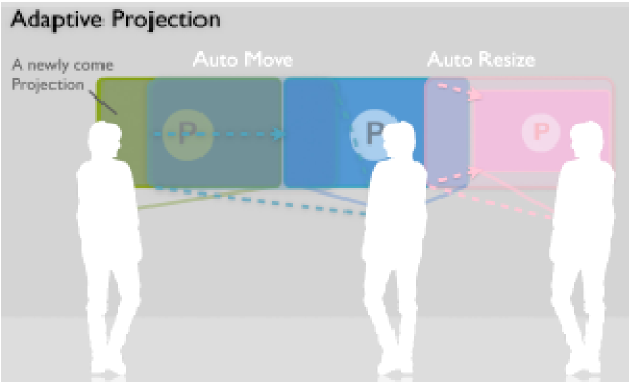


Fig. 21. Adaptable projection [66]

In addition, a rule system can be proposed to control and fix the most of those issues people should be able to register a location or an object as projectable or unprojectable properties. And the projected UI devices should be able to check the information online to inform the user about the projecting rules in that particular place. Furthermore, an individual can register himself/herself to be un-projectable. As a result, nearby projectors will not be allowed to project on the personal space of the individual. [66]

2.3. Projector based user interface in UUI

When Weiser wrote about ubiquitous computing as an alternative to personal computing using the desktop metaphor about 20 years ago [2], he speculated that many displays with integrated and networked processors in various sizes (tabs, pads and boards, in a centimeter, decimeter, and meter scale respectively) would be a core element of the distributed infrastructure, and the way users will make use of it. Looking back, we realize that the visual aspects and graphical human computer interaction play a smaller role in pervasive computing than originally assumed. This may be attributed to the fact that displays are not so cheap to include in pervasive devices: For one, graphical displays themselves are still rather expensive parts. Then, having a display in the design can pose additional constraints on the selected processor such as additional interface lines or an included LCD controller (that are likely to make the processor more expensive), or it may even require additional chips to be included on the board. Another reason why displays are in many cases avoided is, that they consume quite a lot of energy (especially if back-lit). Finally, the displays require a minimum size in order to be informative and readable which may conflict with the design goal to have a small and lightweight device. In consequence it seems that a single small screen display device such as a mobile phone or PDA is at the centre of many pervasive applications. [67]

Projectors are currently undergoing a transformation as they evolve from simple output devices to portable, environment-aware devices with communication ability. An enhanced projector can determine and respond to the geometric context of the display surface, and when used alone or in a cluster can create an ad-hoc self configurable display. Information display is such a prevailing part of everyday life that new and more flexible ways to project data are likely to have significant impact [68].

3. CAMERA PROJECTOR BASED USER INTERFACE AS UUI

As shown before (section II.C.1), ubiquitous user interfaces are regulated by a number of specified characteristics, and as presented, the camera projector based user interface have enough potential to be one of the user interfaces that can answer to all those characteristics.

TABLE I. UUI CHARACTERISTICS AND THE GRANTING PART OF THE CPUIS.

UUI Characteristic	Computer Vision	Projection
Bliss	X	
Distraction	X	X
Cognitive Flow	X	X
No Manuals	X	
Transparency	X	X
Modelessness	X	
Fearlessness Of Interaction	X	X
Notifications		X
Calming	X	
Defaults	X	X

The computer vision part of the camera projector based user interfaces makes them natural UIs, the context awareness using camera sensors is one of the highest, due to the advancement and the ubiquity of cameras in now days life, also, the fact that most of the human interaction is visible acts, the CPUIs used in ubiquitous oriented way can lead to high level of social awareness and serve the privacy constraint of UUI, added to that projection systems, by their adaptability, allows readiness to be integrated in almost all environment types and scenarios without being as intrusive as special devices (Smartphone’s or widescreens) which giving CPUIs a friendly social impact and experience.

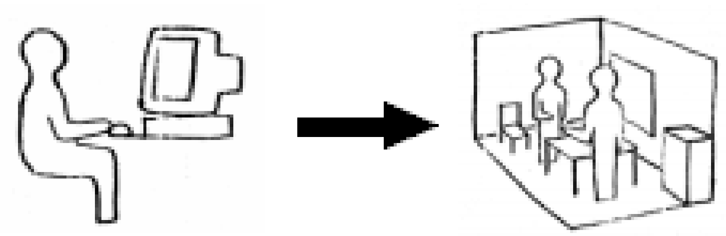


Fig. 22. From GUI to Tangible User Interfaces [46]

All that makes CPUIs a very suitable user interface to ubiquity.

4. CONCLUSION

Camera projector user interfaces are meant to extend passive projection of augmented reality to an interactive system, with navigation and update of the augmentation data. Blurring the boundaries between the physical and digital worlds by making the everyday environment into a self-describing wireless data source, a display surface, and a medium for interaction.

We can imagine an unconstrained environment in the future containing many smart objects. In this environment new objects and users can arrive, move around and interact and leave. If we assume projector-camera systems are installed ubiquitously in this environment offering a display service, the smart objects can request use of the projection capability to obtain a display on its surface and solve its output problem [45].

But the challenge still big, added to the challenges of ubiquitous computing, it leaves no doubt that any try to create a CPUI for ubicomp demands a significantly complex approach. The Chapter III demonstrates some of related approaches.

CHAPTER III. STATE OF ART (POSITIONING OF THE SEARCH PROBLEMATIC)

To surpass the PC era success, Ubiquitous User interfaces require an analogous unified interaction framework. As WIMP and desktop metaphor are the familiar subset of Graphic user interfaces and successful interfaces, there a need to shape the principles and the guidelines of future Ubiquitous interfaces [69].

This section presents projects that have application issues, principles and/or applied technologies in common with the present work.

1. IBM’S STEERABLE INTERFACE EDML FRAMEWORK [70]

EDML framework [70] is a framework steerable projector-camera systems to project onto objects and surfaces in an Everywhere Display. Its architecture comprises three layers:

- The lower services level containing the actual hardware dependant implementation,
- The middle integration layer which abstracts and synchronizes the hardware,
- The high level is the application layer.

Using a 3D world modeling tool the lower levels provided applications for explicit user modeling of displays and provided user localization and geometric reasoning for use in applications such as the display following the user.

The integration layer gives the main classes of the API to build applications with the framework,

- Event management,
- Geometric distortion correction,
- Handling of interactive content to be rendered on the virtual displays.

The EDML framework is designed to characterize steerable interfaces as shown in the six following qualities:

- Moveable output interface: the ability to move video and audio around spatially,
- Moveable input interface: such as steerable cameras and directional microphones.
- Adaptation to user context: the user’s location and orientation.

- Adaptation to environment: reasoning about the geometry and properties of surfaces, adapting to dynamic conditions such as occlusions and ambient noise.
- Device-free interaction: using multi-modal input techniques for interaction.
- And natural interaction: intuitive and usable interaction, sensitive to user context.

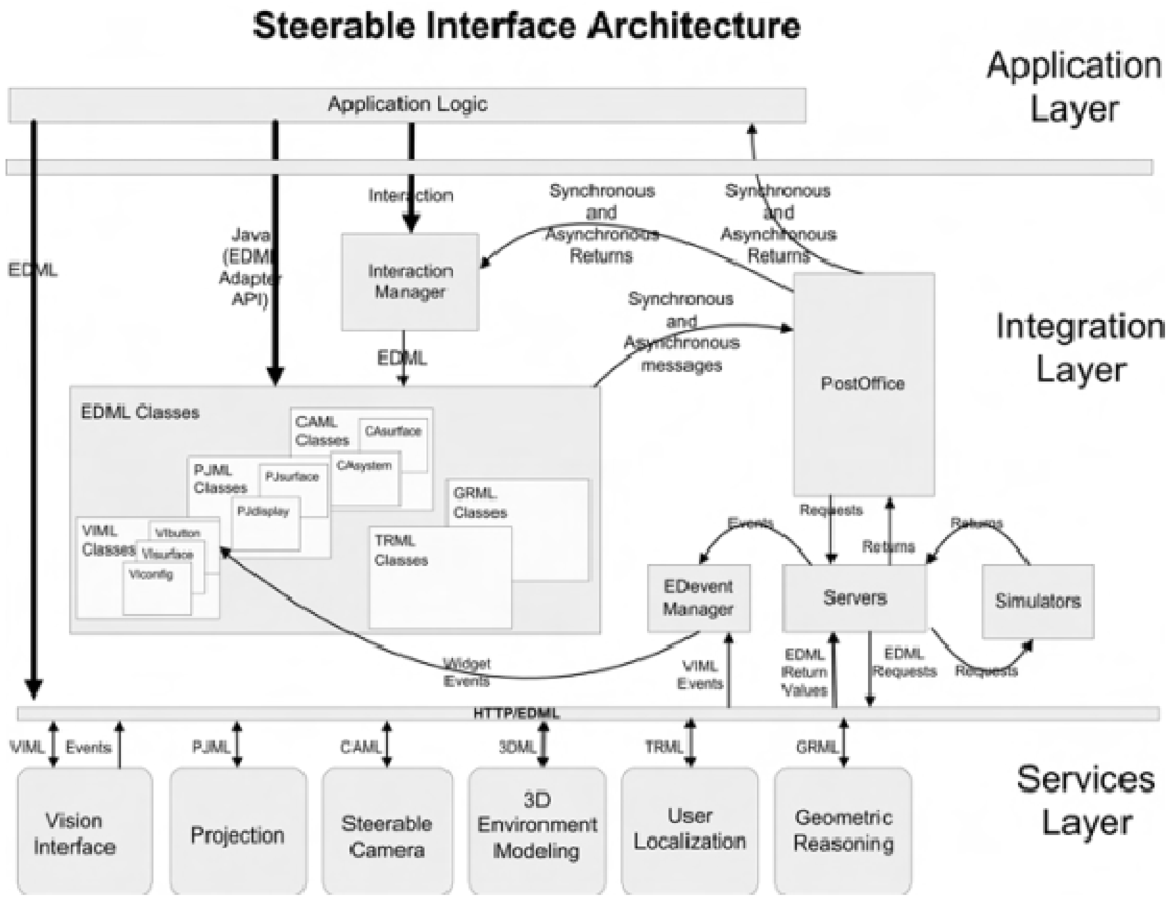


Fig. 23. IBM's Steerable Interface EDML Framework [70].

While supporting a distributed architecture, the framework has a number of limitations. Despite its dynamic occlusion detection of the displays, by using vision-based head tracking, it still limited to known, static environments and needs to be pre-calibrated, in fact, it is reliant on the user to explicitly model their world and pre-calibrate displays. Therefore, the system cannot detect any non-human occlusion of pre-calibrated display locations such as furniture for example.

Although the characteristics Pingali et al. propose are a good way to describe steerable interfaces, they do not directly address the Ubiquitous Computing vision of the future, which proposes computing embedded into everyday objects. It is conceded that “special purpose” devices for interaction could be accommodated; however, there is no support for projection on mobile smart objects in their interaction paradigm.

The framework also does not address any methods for multiple projectors to work cooperatively, assuming only a single projector in any environment.

2. DISTRIBUTED WEARABLE AUGMENTED REALITY FRAMEWORK FOR RAPID PROTOTYPING SOFTWARE INFRASTRUCTURE [71]

After analyzing some of AR frameworks, [71] propose a generic functional decomposition of UAR user interfaces based on their requirements and inspired to show the relevant subsystems and components within them. It is important to note that the subsystems (Input Devices, Media Analysis, Interaction Management, Media Design and Output Devices) are general purpose and generic, however the components within them are just examples. Similarly, Dwarf is one possible implementation of a framework enabling UAR user interfaces. Other implementations adhering to the functional decomposition in fig. would be possible.

The subsystems are:

- The Input Devices subsystem contains input devices that are used to receive commands from the user. Each of these devices offers an individual input modality to be evaluated by the multimodal user interface.
- The Output Devices subsystem renders the signal on the specified output devices. For multimedia based systems several output devices are used at the same time.
- Media Analysis is the process of turning physical user input into abstract tokens. Separate classes, such as gesture analysis, speech analysis, and tangible input analysis deal with the specific properties of different input modalities of the input devices.
- Media Design subsystem contains the software components that present content to the user over any of the cognitive channels, e.g. visual and aural.

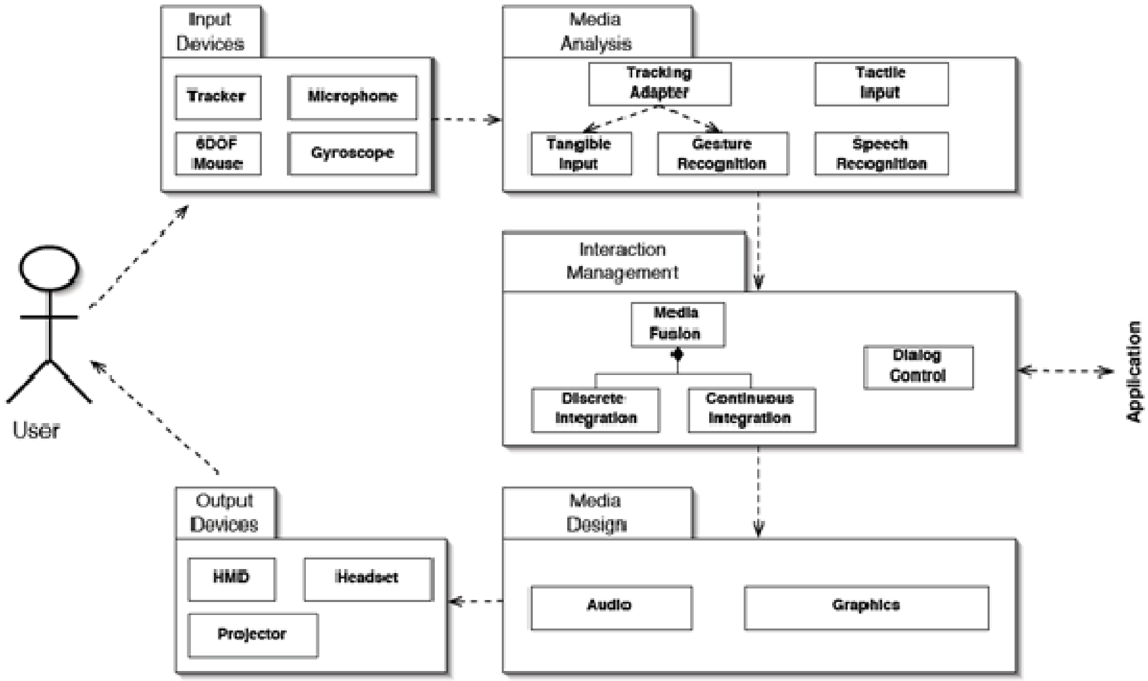


Fig. 24. A generic functional decomposition of Ubiquitous Augmented Reality User Interfaces [71].

- The Interaction Management subsystem determines which output is presented to the user. Current challenges for interaction management are performance, flexibility, adaptivity, usability and an efficient error management. The Media Fusion component takes the tokens of several input channels and infers user intention from them. Two different ways for combining different input channels are considered: Continuous Integration, and Discrete Integration. Finally, the Dialog Control component selects the presentation medium and what to present over it.

Authors of [70] explained several architectural principles and components that make up the user interface framework within Dwarf. In this approach, UI components are arranged in three layers. Most data flows linearly from the Media Analysis layer, which contains input components to the Interaction Management layer where the tokens are interpreted. From there the data flow continues to Media Design layer where the output components reside.

DWARF’s user interface framework (Distributed Wearable Augmented Reality Framework) was designed as a research platform combining wearable systems with ubiquitous environments. Its component model and architecture can be used in several different research areas.

Dwarf is suitable for building highly dynamic, flexible system arrangements within which mobile users, who carry mobile sensors and devices, can be connected on demand to stationarily available, ubiquitous resources that are provided within intelligent environments of the future.

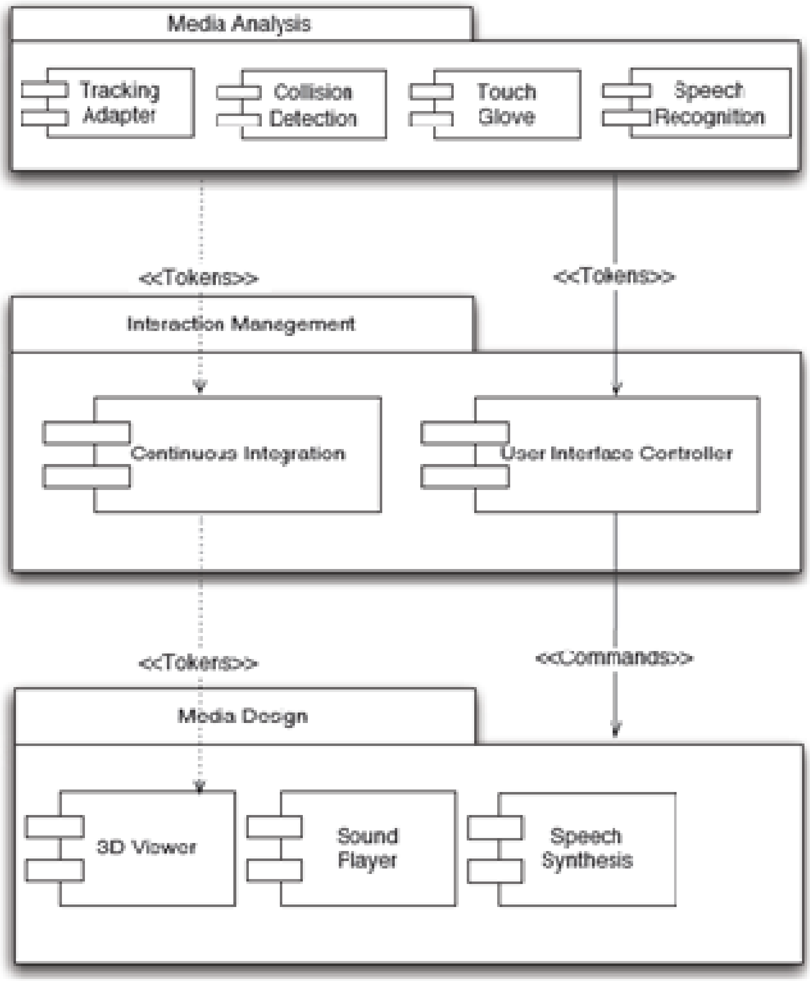


Fig. 25. Functional decomposition of DWARF specific user interface components [71].

Dwarf describes different contributing system parts as separate components that are able to connect with one another across a dynamically configurable peer-to-peer network of distributed processes. Whenever new components are in reach, they get connected automatically into the network of communicating components. The connectivity structure of components is not fixed at startup time. In fact, it can be changed arbitrarily at runtime.

3. Molyneaux and Gellersen's Distributed Framework [72]

This paper presents the theory behind how to consider interaction for projected interfaces with an architecture design and a proof of concept implementation using an augmented photograph album, which merges TUI and GUI to present interactive interfaces on the surfaces of smart tangible objects. David Molyneaux and Hans Gellersen proposed an architecture design and a proof of concept for enabling any smart object to become a tangible user interface.

The projected interface proposed is based on the Cooperative Augmentation concept, which doesn't stores the information required in the environment but in the smart objects themselves.

By abstracting the detection and projection process to services (Fig. 26) and adding a discovery mechanism to smart objects the system enables the use to any type of smart object and any projector or camera hardware.

Main parts of the framework are:

Database Server and Object Proxies: A single world model maintained on the network, supporting services and applications on top of its model, updated when the 3D model (object), sensor states or appearance of the object changes, and manages and tries to minimize the traffic.

Smart Objects: Supposed to describe themselves, their capabilities and sensors, they interact with the system using their states. Those pre-known stets, set by the sensors one or more sensors together, the state changes only when the operations evaluate to within the set ranges.

The framework supports seven separate input modalities supported:

Direct interaction, sensed by camera:

- Manipulation of object location and orientation,
- Interactive projected user interfaces (interaction sensed via a camera and finger tracking).

Direct interaction, sensed by object:

- Direct manipulation of object (e.g. Shaking detected by an embedded accelerometer sensor),
- Manipulation of object morphology (e.g. Opening or closing a book sensed by an object light sensor),
- And manipulation of physical interaction components (e.g. Interaction with buttons or dials on its surfaces).

Indirect interaction that can be sensed or used by object: (1)Manipulation of physical environment remote to object (e.g. switching the light on in the room), (2) and interaction with other smart objects in the environment (e.g. Bringing another object closer).

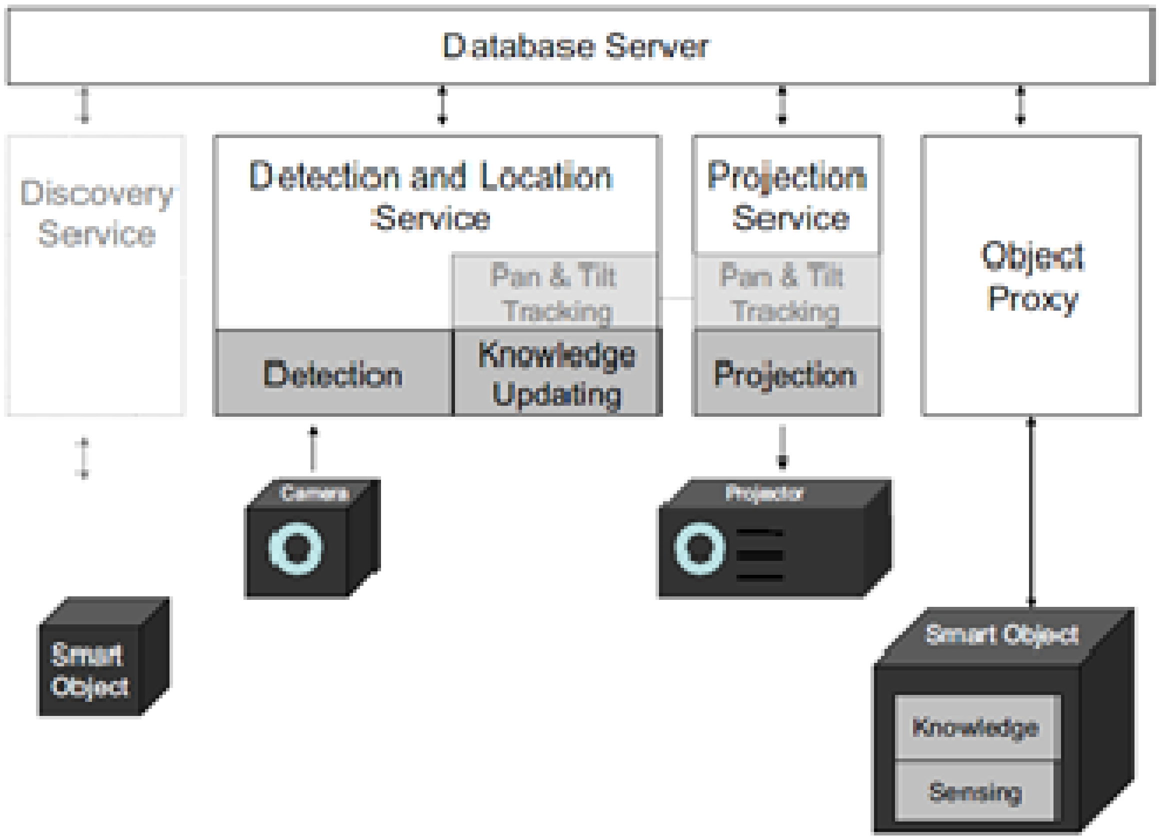


Fig. 26. D.Molyneaux and H.Gellersen's Distributed System Architecture Overview [72].

Detection: A detection and tracking service exists for each camera in the environment. Composed of the two core components Detection and Knowledge updating, and optionally "Pan and tilt tracking".

Authors of [72] designed a multi-cue detection system using algorithms from 4 cues of object appearance: color, texture, shape and features. By combining the results from complementary cues it allows detection of a larger range of objects with high reliability.

Detection methods based on suitability for detecting a particular object are ranked based on three aspects:

- When an object is moving we rank algorithms with shorter average runtimes higher. When an object is static we rank algorithms with a higher detection performance,
- Object's context, by looking at the background model and the object's movement sensing capabilities. (i.e. If the object is same color with background) color based methods are excluded,
- And in objects stored knowledge of their detection performance with scaling and rotation.

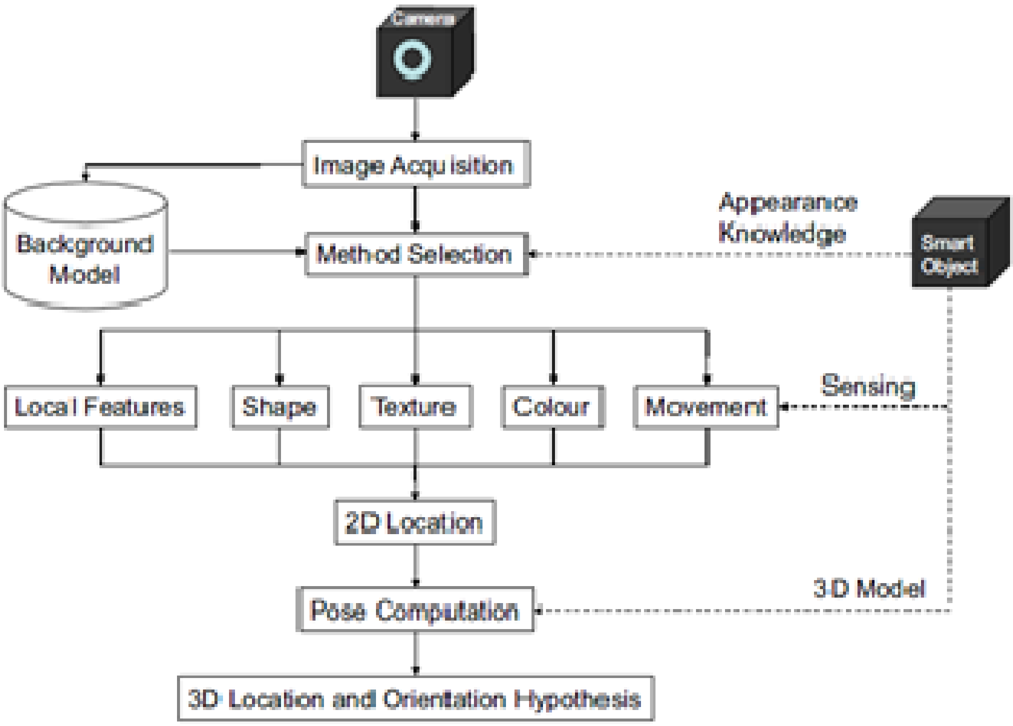


Fig. 27. D.Molyneaux and H.Gellersen‘s Detection Method Selection [72].

After object detection, the framework proceeds to pose detection, from matched local feature correspondences or by fitting the 3D model to edges detected in the 2D image.

Projection: As for detection, a projection service exists for each projector in the environment.

Color and display correction have been added to support various surfaces.

By this framework gave a global approach to augmented environment based on camera projector system, but it’s highly based on smart objet, which are imperatively in need captors to use the interface, requests have to be done from the smart object and not by the user alone.

Also, the infrastructure proposed, despite using steerable projector and cameras, it still a fixed infrastructure which make it loose at mobility point.

The human to human interaction haven’t been addressed even when the techniques involved can serve that propose.

4. REAL-TIME FRAMEWORK FOR VISION BASED HAND DETECTION AND RECOGNITION SYSTEM [73]

Geethapriya et al. proposed a real-time framework for vision based hand detection and recognition system. The keypoints extracted using RASIM technique from every image is clustered to map them into a bag-of-words vector. This could be finally given as input to the multiconlitron training classifier to recognize the hand gestures. Detection of the hand is based on skin's color can be used as a significant image feature to detect and track human hands, after elimination of face and skin-like objects. The system has two stages, offline training stage and online testing stage.

In the training pre-processing stage, the bag-of-words model and the training classifier are built from the training images that contain only the hand gestures from different people in varying scale, orientation and illumination conditions without any background.

The image processing pass by the following process: Scale Invariant Feature Transform (SIFT) is applied to the training image for finding reference object template keypoints. The prediction filters are saved for each keypoint and ring to be used for matching process.

The keypoints of each of the training image is given to the KHM clustering model to build fixed dimensional bag-of-words vector before providing it as input to the multiconlitron classifier. The bag-of-words vectors are grouped and the same hand gestures are given the same class number. The training model is like that built.

In the testing stage, for each frame captured from the webcam, is applied skin detection after the face subtraction to detect hand gesture. The detected hand gesture is saved as a small image. The keypoints are extracted from the small image and given to the cluster model to map into bag-of-words vector, which is given as input to the multiconlitron training classifier model to recognize the hand gestures.

The hand gestures will be detected and recognized. The results will be shown with public image dataset containing hand gestures.

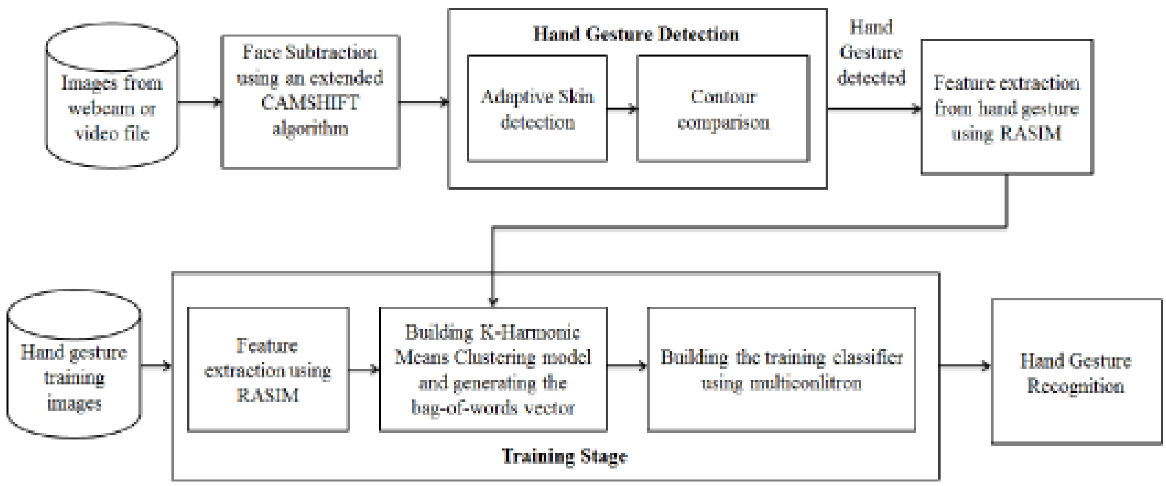


Fig. 28. A Real-time Hand Gesture Detection and Recognition Framework [73]

5. HAND GESTURE BASED USER INTERFACE FOR COMPUTER USING A CAMERA AND PROJECTOR[76]

This framework proposes a hand gesture based human computer interaction system comprising of a webcam and a pocket projector.

The system comprises a camera, a pocket projector and a mobile computing device. The projector projects the display on plain surfaces.

a camera detects and tracks the fingertips using various digital image processing and computer vision techniques. User can interact with the projected screen using his fingertips which are tracked in air by the camera using ‘Camshift’ tracker. A robust method has been developed to detect and recognize single stroke gestures traced with fingertips, which are then translated into actions.

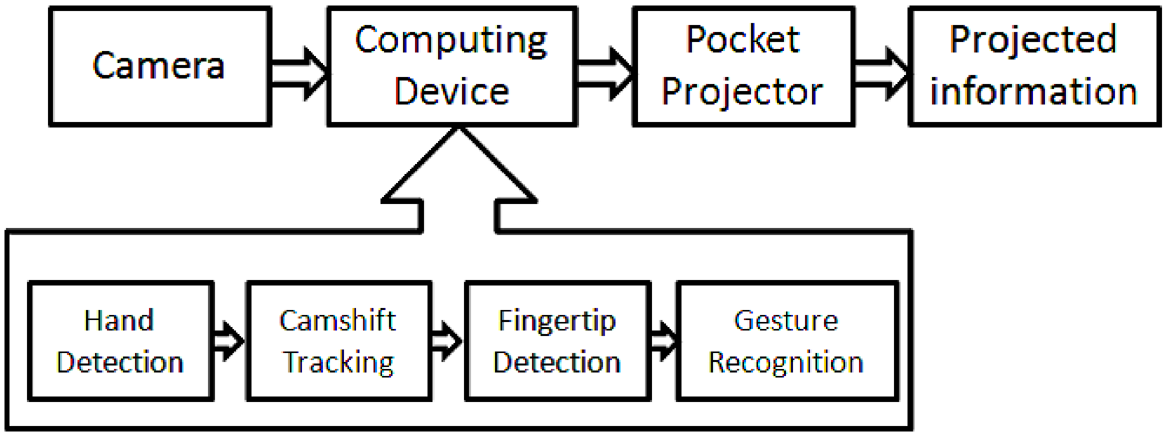


Fig. 29. Diagram of gesture-based user interface system

6. SYNTHESIS

Theses frameworks all present a contribution for ubiquitous Camera projector based user interfaces design. We summarized the main characteristics in the table.

We concluded the main characteristics that such framework should provide to reach:

- Robustness: is the quality describing systems sturdy, durable and able to handle any conditions and answer with sufficient comebacks. Distribution: Ubiquitous Systems support distributed user interfaces.
- Distribution: Ubiquitous Systems support distributed user interfaces that must be independent from the infrastructure as much as possible.

- Flexibility: interaction capabilities should not be limited. It should be adaptable to various cases of use in different surroundings.
- Efficiency: is the ability to achieve desired results in optimized manner; taking advantage from the existing resources.

Moreover, it should consider and support amount of input and output devices and interaction capabilities.

TABLE II. STATE OF ART RECAPETULATIF.

Framework	**Main features**	**Camera type**	**Projector type**	**Positives**	**Negatives**
IBM’s Steerable Interface EDML Framework. (Levas et al., 2003).	Framework for steerable projector-camera systems to project onto objects and surfaces in an Everywhere Display infrastructure.	Steerable camera	Steerable projector	(1) High agility, (2) multimodal input (3) dynamic adaptation to environments, (4) and sensitivity to user context.	(1)Not directly address the Ubiquitous Computing (2) no support for projection on mobile smart objects in their interaction paradigm. (3) and not address any methods for multiple projectors to work cooperatively.
Distributed Wearable Augmented Reality Framework for rapid prototyping software infrastructure (Sandor et al., 2005).	A platform combining wearable systems with ubiquitous environments. Based on Distributed Wearable Augmented Reality interfaces.	Not specified	Not specified	(1) rapid prototyping, (2) suitable for building highly dynamic, flexible system arrangements.	(1) it considers mainly augmentation, (2) do not consider intelligent objects interactions

Molyneaux and Gellersen’s Distributed Framework (Molyneaux et al., 2009).	Framework for enabling any smart object to become a tangible user interface. Based on the Cooperative Augmentation concept, which doesn’t stores the information required in the environment but in the smart objects themselves.	Fixed camera, and steerable camera	Fixed, and steerable projector	(1) Abstraction of the detection and projection process to services, (2) Any projector or camera hardware,	(1) Highly based on smart objet, (2) The human to human interaction haven’t been addressed (3) fixed infrastructure.
Real-time framework for vision based hand detection and recognition system (Geethapriya et al., 2012).	framework for vision based hand detection and recognition system	Not specified	Not included	(1) real-time hand detection and recognition, (2) high recognition accuracy and robustness.	(1) Performances not compared with other systems, (2) Not directly address the Ubiquitous Interfaces.
Hand gesture based user interface for computer using a camera and projector (Shah et al., 2011).	This framework proposes a hand gesture based human computer interaction.	Webcam	Pocket projector	(1) robustness of the gesture recognition algorithm.	(1) The performance of fingertips detection and gesture recognition depends on the contour the hand extracted, (2) limited number of gestures supported.

CHAPTER IV. A FRAMEWORK FOR A CAMERA PROJECTOR UBIQUITOUS USER INTERFACE

1. FRAMEWORK DESCRIPTION

After analyzing some frameworks involving the camera-projection systems, and extracting main advantages and issue, we were inspired to show the relevant subsystems and components that makes those advantages and tried to give solutions to the issues. The main contribution of this paper is to develop a more efficient framework that considers the characteristics of Ubiquitous User Interfaces, and provides very high flexibility that answers the needs enumerated in the previous section.

So we propose a framework for a distributed infrastructure for multi cameras and projectors based ubiquitous interface that allows taking advantage of other kinds of captors and interactions system to enhance the efficiency of the detection and display.

We also introduced in it a door for collaboration between systems as a way to prevent form multi projector and multi user's issues, to maximize the flexibility.

Our framework includes seven main parts (presented next),and it's designed to allow multi input type in addition to camera captors in order to take full advantage of a ubiquitous environment (where computing is integrated everywhere in this environment), and multi output types to cover the gaps that projectors can suffer from.

Those inputs are mainly used by the system to catch the user's environment, actions and all the objects surrounding them. And by user we mean anyone who enters the UIs field of activity (defined depending the case of use). Our proposition takes account of that the objects can interact also with the system and can be computed (smart) or not.

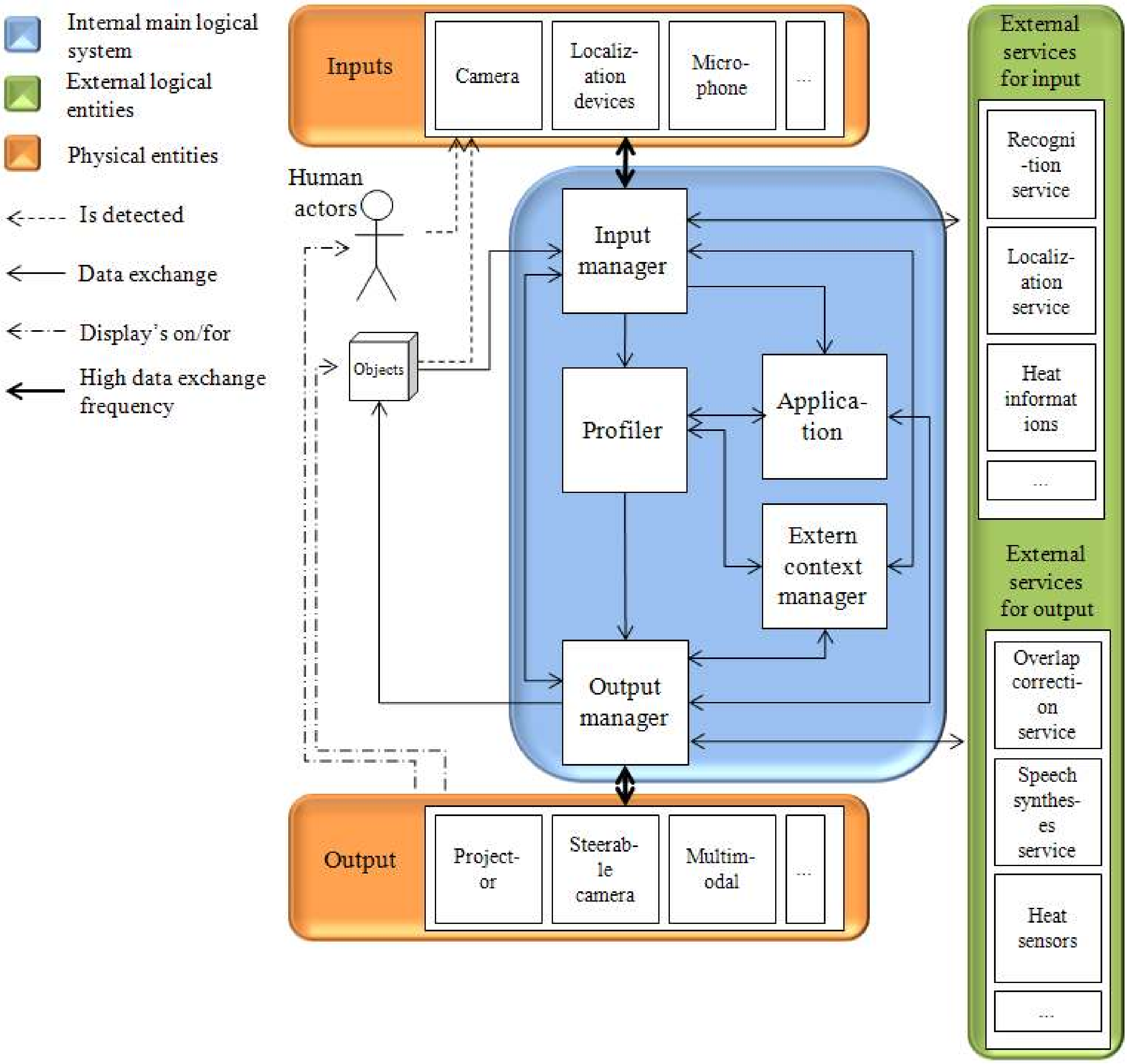


Fig. 30. The framework proposed for a camera projector based ubiquitous user interface

1.1. The input manager

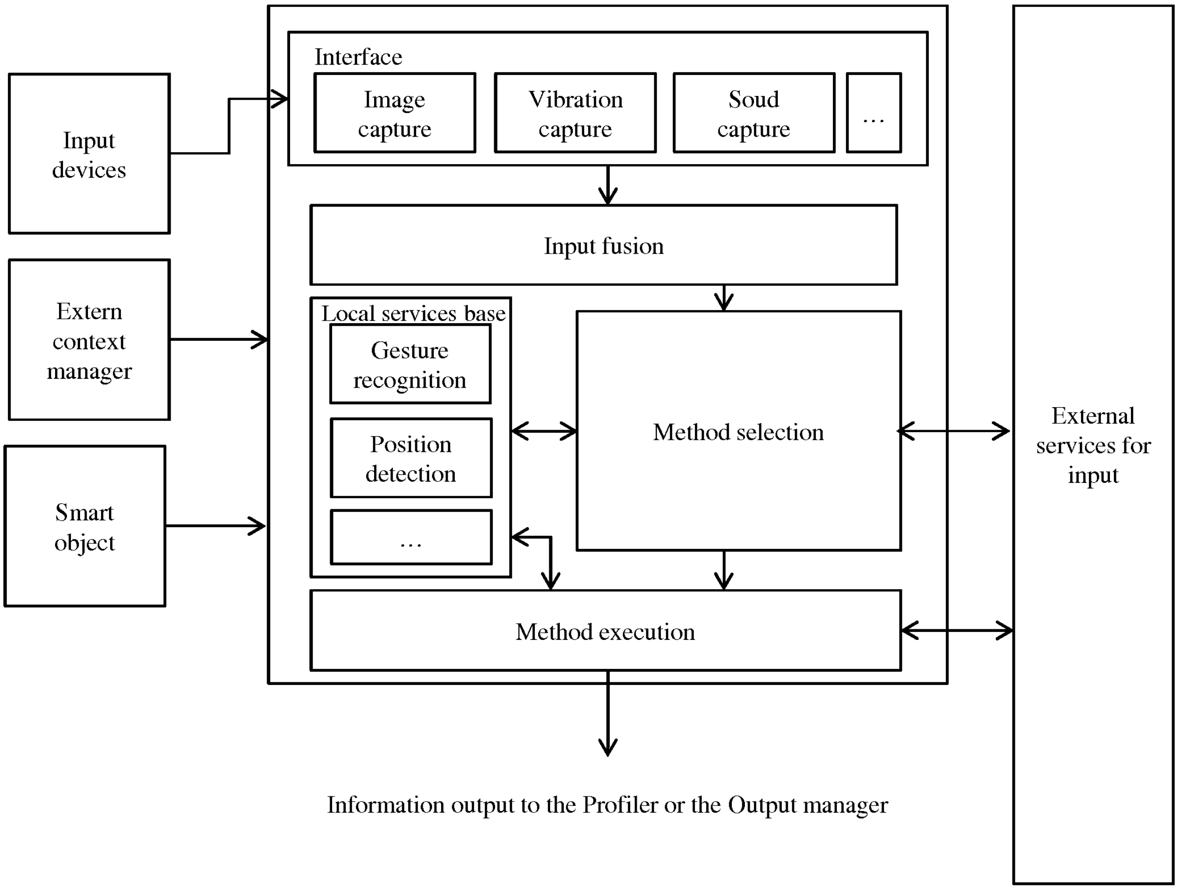


Fig. 31. The input manager

It aims to collect the captured information from input devices and fusions them to get a coherent input information. Those collected information consist of context and commends from input interaction from users and smart objects. After making semantic information reliable for other parts of the system to be used.

Is the only part related to the input hardware, it proceeds by:

- Gathering the information from input devices (images, sounds, heat values, Etc.) using the interface.
- Merging those information and those collected from the external context manager and sorting them by use reliability, for example images and GPS data for localization and later for recognition,
- Method selection, after getting coherent data, it is needed to extract the events and commands from it. For that, many process may be applied, i.e. gesture recognition, in this step the manager should decide which method to use from the available methods

from “local services base”, or from “the external services for input” in the network or internet. A method catalog can be proposed to incorporate to the “local services base”, where the services are divided and stored by selected characteristics (speed of execution, precision, color based etc.).

- The execution provides the only necessary resulting data in a specific format for the other modules; globally it is in this part where high level execution conflicts can be managed.

1.2. The external services for input and the external services for output

Are external modules to the system, containing methods that can be used by respectively by “the input manager” and “the output manager”, it can contain two main types of services:

- Method services, services that are used to treat the information provided to them and returns significant information to the using module, i.e. “The external services for input” module can contain methods that treat optical flow for advanced motion semantization like dance detection or crowd movement prediction that demands a relatively high computational capacities which this module can find in outer infrastructures all depending the service.
- Informative services, provides additional information that can help in the execution step, Such as weather, temperature, daytime etc.

As for “the local services base” of “the input manager” a method catalog, or more specifically, a service catalog can be included to this module to sort the different services.

1.3. The profiler

Due to the need of transparency in UUIs, asking users to authentify themselves continuously or periodically as for classical UIs isn’t adapted, so we proposed to integrate a module that has for main task the pre-auto-authentication step based on visual aspects and the additional data relayed by the “input manager” relatively to its incorporated database.

So, it proceeds updating the database which contains information about the objects and the users by setting a profile for each of those entities. That helps identify the elements detected by the “input manager” objects or users. Its principal tasks are:

- Creating new profiles for newly detected elements, and attributing the correspondent specifications for the new profiles.

- Updating and storing the history of the elements, and the in-between relations to “the internal database”.
- Providing the necessary profile information to “the application”, “output manager”, and “external context manager” modules.

Used in its best way the profiler that can allow social awareness by exploiting the relations within the database, and then, lead to an efficiently privacy allowing system.

1.4. The application

Is the main part that the users are interested to, takes advantage form the information provided by the other main modules and provides the information that is needed to be outputted to the user to “the output manager”. The information can be device restriction free or specified to be prompted via a specified device.

1.5. The output manager

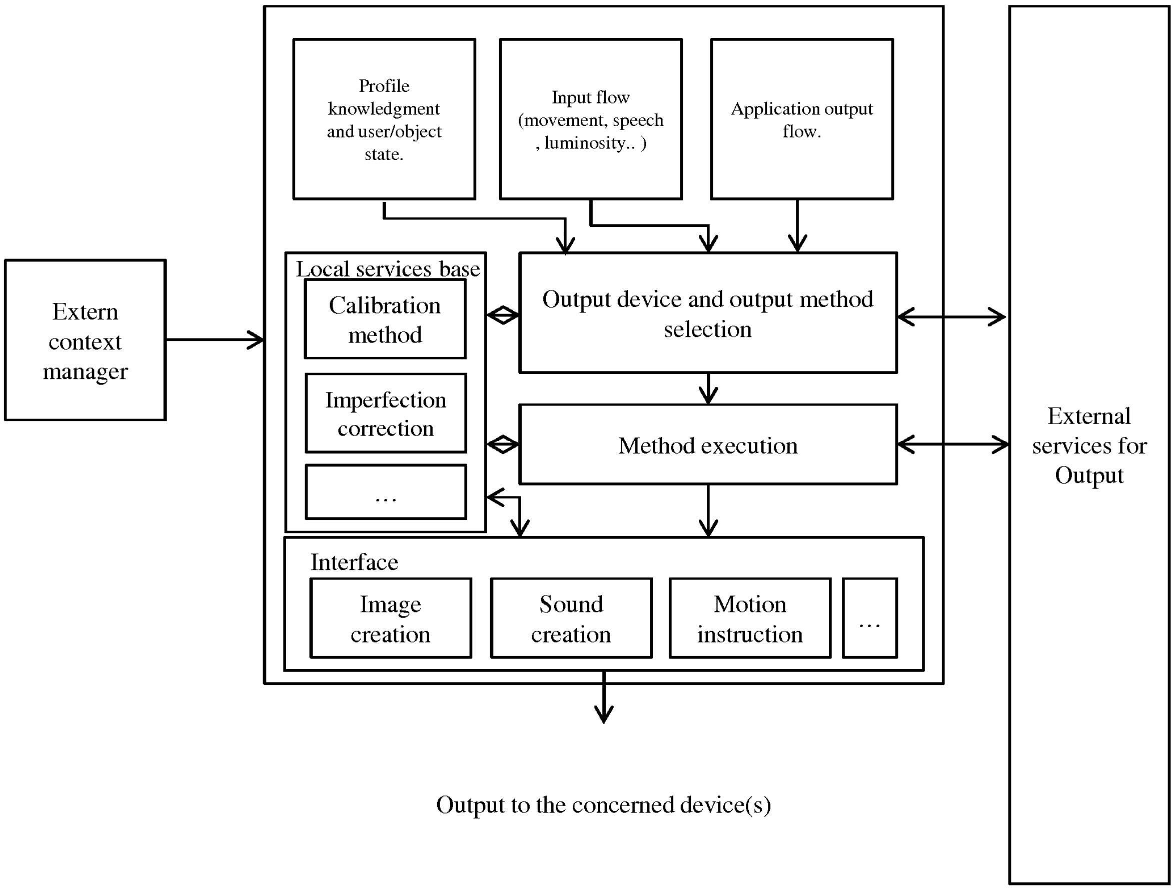


Fig. 32. The output manager

Responsible of the way the information is relayed to the user and output devices management, proceeding by a similar way to "the input manager's" approach,

- Gathering the information from the other modules,
- Selecting the adequate device and method to output with, based on the information collected and needed to be prompted; it's related to an internal "local services base" and if needed an "external service for output" module.
- Executing the selected method and providing a ready to output data.

In this module there is no information fusion because the incoming data are divided efficiently:

- The profile acknowledgment and user or object states are used to know the direction or the source to output with.

- The input flow helps to know additional information about the environment (noisy, projectable spaces etc.), also can be used for low level interaction detection (without needing to go through the application).
- The application output flow is the information which is mainly aimed to prompt out can be device restriction free or specified to be prompted via a specified device.

1.6. External context manager

External context manager takes the responsibility of communications with other systems and exchanging with them useful information especially about context, to allow fixing multi systems issues [66] (which are frequent in ubiquitous systems) by opening a collaboration door. Gathers form and informs “the input manager”, “the profiler”, and “the output manager” with related information:

- Gathers form “the input manager” information about the environment (heat, noise, etc.) and the user (number of users, direction, in movement or not etc.); and informs it about other systems data to get fusioned in with other in-system input data.
- Gathers form “the profiler” information about the identity of the user and the objects (object’s uses, user’s grade etc.) and informs “the profiler” about the data correspondent data gathered from other systems to enhance its “local profiles database”.
- Relays “the output managers” of the systems to help selecting the adequate output method (collaborative, others respectful, etc.).

2. CONCLUSION

In this chapter, we presented a framework for design and implementation of Ubiquitous User interfaces based on cameras and projectors. This architecture respects Ubiquitous requirements and shows many capabilities. It supports distributed interfaces, smart objects and many modalities. It demonstrates efficiency and flexibility.

CHAPTER V. VALIDATION OF THE SEARCH PROPOSITION

1. INTRODUCTION

Due to the lake of the possibility of implementation and deployment in the current time, we proceeded to scenario representation involving our solution to validate it.

As presented, our framework can answer many cases of use. Learning environments are ones that have a high factor of instability in terms of devices and uses, so it is the application case that we chose to illustrate in this chapter.

2. PRESENTATION OF THE SCENARIO

Education has experienced major changes in recent years, the development of digital information transfer, storage and communication methods all involved in learning process enhancement. Ubiquitous learning environment is a situation or setting of pervasive and omnipresent education. Education is happening all around the student. Data source is present in the embedded objects. U-learning is extended from m-learning wich is extended from e-learning. [77,78] After the initial impact of computers and their applications in education, the introduction of e-learning and m-learning represented the constant transformations that were occurring in education. Now, the assimilation of ubiquitous computing in education marks another great step forward, It is conveyed that it, allows students to access education flexibly, calmly and seamlessly. U-learning has the potential to revolutionize education and remove many of the physical constraints of traditional learning. and offer great innovation in the delivery of education, allowing for adaptation and customization to student needs [78].

More specifically the scenario that we chose involves u-learning in a classical class room, with classical in-room users teacher and students, and external (out of the room) actors, to illustrate the ubiquity of the user interface. Those users are supposed to be interacting simultaneously and continuously with the interface, getting specified output for each user and use. In fact, we expect a selection of an adequate space (wall) to serve as a projection board and additional display spaces for additional information, such as, notification space on the teacher’s desk, annotation and interactive space on student’s tables, etc.

The user interface is used with adequate applications, more frequently a course managing application but the class can also need many other applications i.e. geographical, modeling applications (diagrams, 3D, chemical.. etc.), calculus etc.. Each of those applications needs an adequate interaction and display way involving all the classroom’s elements in the most transparent, assuring way and adapted to each users state's (rotation way, position etc.) and grade i.e. only teachers can change the course projected on the board.

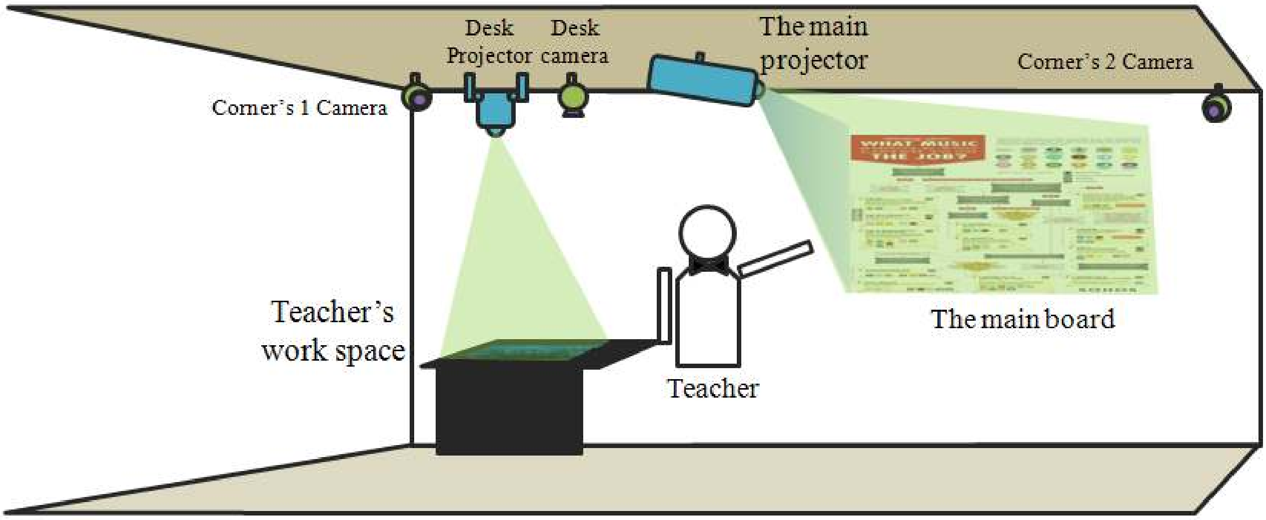


Fig. 33. The class room from student's point of view

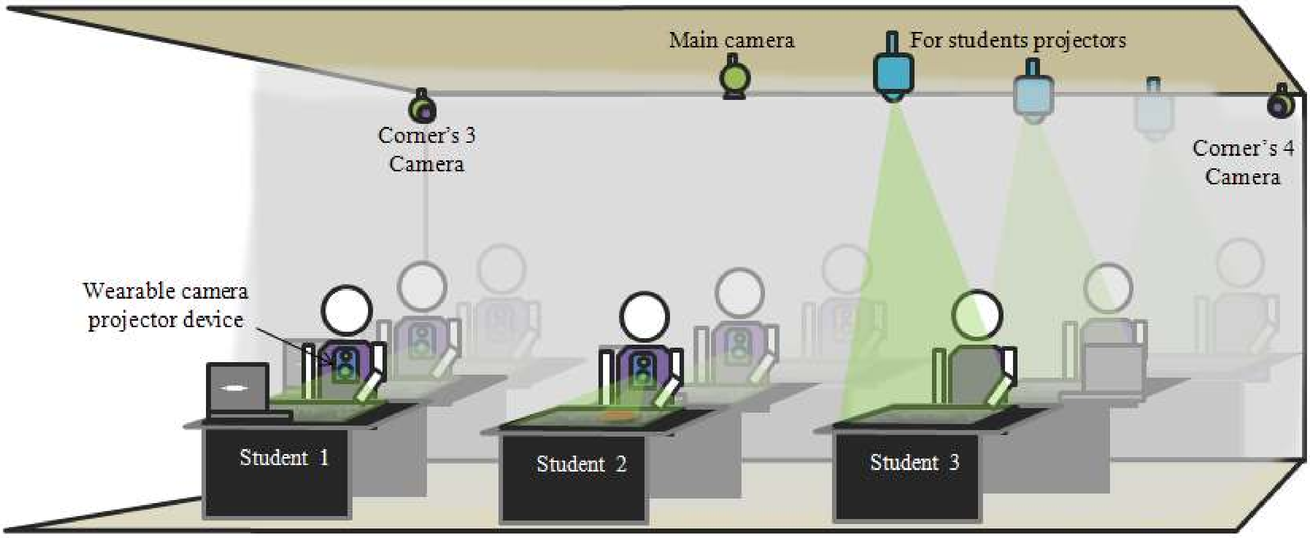


Fig. 34. The class room from teacher's point of view

As an enhancement of the user interface's capacities, we imagined involving user's personal devices in the interaction process, as inputs or outputs. So for example if students have wearable camera projector devices we will use the capacities of those devices to gather students actions and display on their tables to enhance the quality instead of using the devices embedded on the class room. Other personal devices such as Smartphone and laptops will be involved in the interaction.

Collaborative actions are continuously made within the class, form the basic "send a question to the teacher's notification space" to multi-users modeling. We can imagine students and the teacher trying to construct a global model, each student works in a specific part of the model on the space projected on their tables, and the totality of the model is displayed on the board space allowing the teacher to see, help and correct the students in their demarches.

External actors can participate to the class, additional teachers and experts, or considering inapt to be present students (infirm, isolated, etc.) the user interface will get the necessary information and connect them the classroom system, granting them as much as possible interaction capacities to match the in room user's capacities. So for example, the user interface has to be aware that those external users cannot see the projected board so another way to display the on board information is selected, using those external user's own systems.

The u-learning (as for all ubicomp sub-domains) has to be continuous in time and aware of the needs of the users. The system has to show to the users adapted information in and only in adequate places and times considering the user context. So for example notifications about the course will stop when the students are out for lunch and resume as soon as they enter the pedagogic building. Also all inadequate notifications (i.e. information about other courses) will be disabled when the class begins.

3. PRESENTATION OF THE ELEMENTS OF THE SCENARIO

As presented the chosen scenario involves many elements. In this section we present some clarifications about those elements:

3.1. The human actors

Or the users, the center of the system's focus. Capable of interaction. They are the sources of most of passive and active commands. In this scenario case, users are more specifically:

3.1.1. The main Teacher

Responsible of the course. Allowed to use almost all the element of the user interface, such as, manipulating the main board, opening and closing automatically the class room windows, manipulating student's workspaces (on their tables) to help them, implicating external teachers and allowing external students to intend the course remotely.

He can also have personal spaces on their desk allowing them to write their personal notes and receive student's questions; he can chose some of those questions to display to the main board to answer it etc.

3.1.2. Students

presenting the most of the system user's population, coming to the class to learn and understand things exposed by the teacher, and using the interactions allowed by the system such as being able to write and draw notes to their personal workspaces for a later consultation, or working collaboratively with their comrades on group projects for example.

3.1.3. External teachers

Connected remotely to the class room's u-learning system. They are teachers and experts that the main course teacher can as use their help to provide a qualitative teaching. Those external teachers have to see all the reliable information displayed on the class room (boards, students etc.). So their own devices have to be used to display those information.

3.1.4. External students

Similar to the external teachers case, absent physically from the class room. Connected to the class room remotely. They have to get access to all the necessary features that in-room students have (hearing and viewing the teacher, seeing the content of the board and writing on it (virtually) and participating to the cause). Their own devices will be used for that.

As presented all those actors still have to get adequate information considering each one's context and condition, as an example the display may be brighter for the users who have vision difficulties etc.

3.2. Non human elements:

3.2.1. Virtual elements

The board

Displays the main course information, meant to be seen by all the users. In this scenario it will be projected on the front wall of the class room. For the external teachers and students the board will be added to the personal display (for example if it is a computer screen the student will have two windows one for the board and one other for the personal workspace).

The personal workspace

An element little more interactive than a virtual textbook. For each user, a personal workspace is attributed, allowing them to make actions to enhance their experience.

Every workspace has to answer multiple needs and be adapted to its user, some examples:

- Users can make notes, store them, exchanging them,
- They can also drawing diagrams or 3D modes,
- Students can send questions to the teacher,
- Teachers have to be notified and see student's questions on their workspace,
- Students have to be able to see the progress of their comrades in group projects.

The ideal display way for that space is to be projected by user's devices on their places but can also be displayed using projectors specially placed on the class room (on the top roof) as a part of the class room's system.

3.2.2. Real world elements

The environment

The surrounding elements of the users, in a global point of view, in our case we have the for the system implemented in the classroom: the class room itself, and more generally the building. And for external users: the user's room or office.

The objects

All the in room elements that the user interface can use to display information or to understand the users more. We can cite: Walls, chairs, closets, black board, student's tables, pens, windows, etc.

Some or all of those objects can be Smart. Containing sensors and communicating with the class room's system, they provide the user interface with additional information and details about themselves and the users.

Devices

We can divide them to three main categories:

Class room's devices: the devices that are embedded to the class room, mainly used as input and output peripherals. We cite:

- Cameras: we assume that we have on camera placed in each of the four top corners of the class room in a way that they can cover all the elements on the surface. Additional camera is placed over the teacher's desk to get in more precession his movement's and gestures, and one other better quality camera on the center of the class room placed on top of the students that will have similar work to the teacher's cam. No need for special kind of camera on our scenario but depth cameras can be added to enhance the movement detection.
- Projectors: To display the board, and all the graphical elements of the user interface. A main projector placed to display the board. Another one is placed over the rostrum or the space where the teacher is meant to move a lot, it's used to display the teacher's workspace; we chose to make it steerable so it can follow the teacher an display some shortcuts on his hand for example.
- A microphone: placed on the middle of the class room capturing all the sounds in it.
- A wireless ID card reader (using near field communication as an example) is placed near the room's entry.
- Additional devices can be added, we chose to add a heat sensor, and mortised windows.

User's connected systems: meant to interact with the class room's system, more specifically we take it as wearable camera projector systems; some students can be equipped with that. Those devices will help the class room's system. By communicating data with it to select which projector for example to use to show information to the users or which cam is more adapted to be used for displacement and gesture detection.

Additional user's devices: such as laptops, Smartphone or smart watches. Those can or not be relayed to the system to help the u-learning process, we consider that the laptops (or tablets) will be relayed to the system.

4. APPLICATION OF OUR FRAMEWORK ON THE SCENARIO

Now we will show how a basic implementation of our framework is supposed to answer all the cited scenario needs, step by step.

4.1. Implementation aspect

We assume that the class room will be connected to a main classroom computer which is connected to the global datacenter of the establishing which will provide additional computation power and can contain the external services.

4.2. Environment and adaptation to the palace:

First of all the system once implemented in the class room (camera placed on the corners, projectors on the tops, etc) have no perception of the environment. So an adaptation process will run, in other words:

- Projectors calibration.
- Camera position acknowledgment: the system will need to know which are the frontal cameras and which are the back cameras, a simple way to do that is by using the main projector, the cameras that can "see" the board projected are back cams the ones that do not are the front cams and the main cam.

We assume that for those two acts only the input manager and the output manager modules will be involved:

(1) The input manager and the output manager consults the system state (it will be set to not ready),

(2) They start a readiness service (available in the local service base): for the two cases cited it will be in need of sequential execution, fist the camera position, then the projector calibration,

(3) The service of camera position detection uses adequate as a way to do that, as an example we will use the projector:

a. The service requests the input manager to turn off the main display and to send him the images from the cameras while projectors are off.

b. The input manager requests the output manager to off the main display.

c. The output manager "looks" what is the main display (for example the one connected to the port 1), and turns it off.

d. The input manager relays to the service the gathered images from the cameras while projectors are off.

e. The service request's the input manager to turn on the main projector and display a white board and capture the images.

f. The input manager relays the "turn on" request to the output manager which will execute it similarly to 'c.'

g. The input manager gathers the pictures a second time and sends it as asked to the service.

h. The service executes a comparison algorithm and detects which image contains a change equal to a white board and send those information's to the input manager.

i. The input managers attribute's the positions to the physical devices and set's the camera position state to ready.

Similar execution process will be used for other cameras detection, and projector calibration, involving the specified service for each need.

Also as a step of readiness process, the detection of the objects within the environment. Chairs, walls, desk, windows, have to be detected and sorted in adequate classes (projectable, movable, related to teachers, related to students etc) all those characteristics are detected by specified services that the input manager invokes. Then he relays those data to the Profilers who will store those elements on his database.

4.3. Passive execution

After all the readiness steppes are done, the system start's working continuously and monitoring the element of the class room executing mainly change detection processes (services in the input manager).

As an example, as soon as the teacher enters the class room, the cameras detects him. The input manager's service informs the input manager that a person has entered the room.

The input manager invokes face recognition service to identify the person's face characteristics and send the result's to the profiler.

The profiler will compare those data with the data in his database and get his profile. This profile is then relayed to the course application.

The application then check's the course planning and get's the slides that the teacher is mean to teach that day and send with the workspace information related to that teacher (notes, preferences etc.) to the output manager and informs him about which is the information needed to be prompt on the board and which is the workspace.

The output manager after getting that display information, check's the input information such as luminosity, or if there is something on the desk or the front wall (i.e. sunlight) etc. And send those information to a correction and adaptation service to process the display needed and to give better information to send to the projector. The output manager's selects the and informs the main projector and the desk projector to display respectively the board (slides) and the workspace.

Other examples of passive execution can be found:

- Start student's workspaces when they sit on their tables
- Auto-close the windows when it's rainy outside,
- Brighter the projection when it's too luminous in the room,
- Move the projected space when the teacher put's his briefcase on the desk.
- Etc.

4.4. Active execution

Active execution is meant to answer the users' commands. Some typical cases:

- When the students want to intend the classroom:
    (1) They present their digital id card to the door which will communicate their id to the main user interface system (as another way of identification than computer vision).
    (2) The systems start tracking them until they sit down on their tables and then tries to communicate with their wearable camera a projector device.
    (3) This external context manager of the class room's system and the student's device's begins a collaboration process and construct's a relay(via Bluetooth for example) between the class system and the device's

element's so the class room's can get what the device's get. And the device can get information's from the class room.

- When the students manipulate their workspaces:

  The system will try to figure out which camera to use to get the best result: if it's arm movement the camera placed on the top of students will be used. If its finger movement the camera embedded in the user's device will be used, it's a service of the class room's input manager who will help to take that decision.

- Etc.

The external actors will have to communicate through the external context manager to get the needed information and participate to the two processes. In addition the user's laptops and Smartphone's are considered as smart objects and will communicate directly with the input manager and the output manager of the class room and will be used as input/output devices with the help of adequate services.

5. Conclusion

In this chapter we exposed the validation of our framework proposed by scenario. A u-learning scenario showing interactions between a system integrating our framework and the class users; it shows this system capabilities and the role of each component to reach traced goals.

This scenario doesn't provide all the system possibilities. Still, it shows the usefulness of each component to reach the ubiquity of the interface.

CONCLUSION AND PERSPECTIVES

This paper is part of the current trend to provide a consistent framework for Ubiquitous user interfaces development. Our work tries to give a generic prototype to facilitate development of Ubiquitous interfaces particularly ones based on camera-projectors. This framework is a new approach considering Ubiquitous Systems requirements; Ubiquitous interfaces characteristics and interactions forms.

In the first chapter, we gave an overview to Ubiquitous Computing and Ubiquitous interfaces. We gave the fundamental aspects about the subject. We started by defining ubiquitous computing and current research challenges. After that we described ubiquitous interfaces characteristics and its main classes which are suitable to realize interfaces for applications in environments with lot of computation, input, and output.

The second chapter was centered on Camera projector interfaces. It started by a presentation of computer vision and projection separately before combining them as ubiquitous interface. This chapter makes in evidence these technologies in their classic use and highlights their usefulness in Ubiquitous interfaces.

After that, the third chapter sums up some works proposing frameworks considering camera-projector systems; they treated many aspects and gave many approaches, which we considered to make our own framework.

Our main contribution is in the fourth chapter; we considered strengths of analyzed works and tried to avoid their weaknesses to make a generic framework based on cameras as main input and projectors as main output. We made use of confluence between multimodal, surface, tangible and AR-based user interfaces to get the ubiquity. This framework considers natural interactions, smart objects, context awareness and distribution. In this chapter, we described the main components of our framework, their roles and functioning. After that, we gave a typical scenario which shown this framework consistence, and the importance of each component.

Robustness of this framework can depend on technologies and services limitations. That can affect recognition accuracy that affect all the interaction. The main amelioration could planned is to enhance services accuracy and duration.

Moreover, it is needed to evaluate this work on real applications. Scenario given is not enough to prove the efficiency of the approach. An implementation still in need before a total validation of the proposal.

Another perspective is to extend this framework to all kinds of interfaces. It will not be a simple task to cover all interfaces variants, but it seems to be less and less far to be achieved. Developing each component to support more possibilities looks like a promising way to reach this goal. The study of security aspects can also bring enhancements to the framework.

For us, This work was very important to accomplish our educational path. It allowed to us to understand and apply the methodology of academic research. We were introduced to a new subject, Ubiquitous Computing, which is a very outsized multidisciplinary domain. We got the opportunity to discover the research environment and its community.

To conclude, Ubiquitous Computing and Ubiquitous Computing Interfaces have now reached their maturity, but due to their complexity and diversity they still contain a lot of research opportunities and challenges.

REFERENCES

[1] Want, Roy. "An introduction to ubiquitous computing." Ubiquitous Computing Fundamentals (2010): 1-36.

[2] Weiser, Mark. "The computer for the 21st century." Scientific american 265.3 (1991): 94-104.

[3] Weiser, Mark. "Some computer science issues in ubiquitous computing."Communications of the ACM 36.7 (1993): 75-84.

[4] Quigley, A. J. "From GUI to UUI: interfaces for ubiquitous computing." Ubiquitous Computing Fundamentals (2010): 1-36.

[5] Dix, A., Finlay, J. E., Abowd, G. D., & Beale, R. "Human-computer interaction." (2004).

[6] Höök, Kristina. "Steps to take before intelligent user interfaces become real."Interacting with computers 12.4 (2000): 409-426.

[7] Kurkovsky, Stan. "Pervasive computing: Past, present and future." Information and Communications Technology, 2007. ICICT 2007. ITI 5th International Conference on. IEEE, 2007.

[8] Bell, Genevieve, and Paul Dourish. "Yesterday's tomorrows: notes on ubiquitous computing's dominant vision." Personal and Ubiquitous Computing11.2 (2007): 133-143.

[9] Friedewald, Michael, and Oliver Raabe. "Ubiquitous computing: An overview of technology impacts." Telematics and Informatics 28.2 (2011): 55-65.

[10] Aarts, E., and J. L. Encarnação. Into ambient intelligence. Springer Verlag,Berlin, Heidelberg, New York (2006): 1-16.

[11] Bardram, Jakob; Friday, Adrian."Ubiquitous Computing Systems." Ubiquitous Computing Fundamentals (2010): 37-94.

[12] Mühlhäuser, Max, and IrynaGurevych. "Introduction to Ubiquitous Computing", Handbook of Research on Ubiquitous Computing Technology for Real Time Enterprises. Information Science Reference, (2008) : 1-20.

[13] Ndiwalana, Ali, et al. "Ubiquitous computing: By the people, for the people."CHI'03 Extended Abstracts on Human Factors in Computing Systems. ACM, 2003.

[14] Banavar, Guruduth, and Abraham Bernstein."Software infrastructure and design challenges for ubiquitous computing applications." Communications of the ACM45.12 (2002): 92-96.

[15] Abowd, Gregory D., and Elizabeth D. Mynatt. "Charting past, present, and future research in ubiquitous computing." ACM Transactions on Computer-Human Interaction (TOCHI) 7.1 (2000): 29-58.

[16] Prekop, Paul, and Mark Burnett. "Activities, context and ubiquitous computing."Computer Communications 26.11 (2003): 1168-1176.

[17] Camp, L. J., and K. H. Connelly. "Beyond consent: privacy in ubicomp." Digital privacy: Theory, technologies, and practices (2007): 327-47.

[18] Wasinger, Rainer, and Wolfgang Wahlster. "Multi-modal human–environment interaction." True Visions. Springer Berlin Heidelberg, 2006. 291-306.

[19] Bhuiyan, Moniruzzaman, and Rich Picking. "Gesture-controlled user interfaces, what have we done and what’s next?." Proceedings of the Fifth Collaborative Research Symposium on Security, E-Learning, Internet and Networking (SEIN 2009), Darmstadt, Germany. 2009.

[20] Mueller, Florian, Stefan Agamanolis, and Rosalind Picard. "Exertion interfaces: sports over a distance for social bonding and fun." Proceedings of the SIGCHI conference on Human factors in computing systems. ACM, 2003.

[21] Ishii, Hiroshi, and Brygg Ullmer. "Tangible bits: towards seamless interfaces between people, bits and atoms." Proceedings of the ACM SIGCHI Conference on Human factors in computing systems. ACM, 1997.

[22] Shaer, Orit, and Eva Hornecker. "Tangible user interfaces: past, present, and future directions." Foundations and Trends in Human-Computer Interaction 3.1–2 (2010): 1-137.

[23] Underkoffler, John, and Hiroshi Ishii. "Urp: a luminous-tangible workbench for urban planning and design." Proceedings of the SIGCHI conference on Human factors in computing systems. ACM, 1999.

[24] Gross, Tom. "Ambient interfaces: design challenges and recommendations."Proceedings of the 10th International Conference on Human-Computer Interaction-HCII. 2003.

[25] van Mensvoort, Koert. "Datafountain: Money translated to water." (2005).

[26] Ishii, Hiroshi, et al. "ambientROOM: integrating ambient media with architectural space." CHI 98 Cconference Summary on Human Factors in Computing Systems. ACM, 1998.

[27] Tosas, Martin. Visual articulated hand tracking for interactive surfaces. Diss. University of Nottingham, 2006.

[28] Dietz, Paul, and Darren Leigh. "DiamondTouch: a multi-user touch technology."Proceedings of the 14th annual ACM symposium on User interface software and technology. ACM, 2001.

[29] Zhang, Shuai, et al. "Low-cost interactive whiteboard using the Kinect." Image Analysis and Signal Processing (IASP), 2012 International Conference on. IEEE, 2012.

[30] Höllerer, Tobias, and Steve Feiner. "Mobile augmented reality."Telegeoinformatics: Location-Based Computing and Services. Taylor and Francis Books Ltd., London, UK (2004).

[31] "New augmented reality system shows 3D GPS navigation through your windshield, Digital Trends." Technology News, Product Reviews and App Reviews, Digital Trends. http://www.digitaltrends.com/cars/new-augmented-reality-system-shows-3d-gps-navigation-through-your-windshield/ (accessed May 25, 2013).

[32] Coutaz, Joëlle, and Jean Caelen. "A taxonomy for multimedia and multimodal user interfaces." Proceedings of the 1st ERCIM Workshop on Multimedia HCI. 1991.

[33] Satyanarayanan, Mahadev. "Pervasive computing: Vision and challenges."Personal Communications, IEEE 8.4 (2001): 10-17.

[34] Tang, Lei, Zhiwen Yu, Xingshe Zhou, Hanbo Wang, and Christian Becker. "Supporting rapid design and evaluation of pervasive applications: challenges and solutions." Personal and ubiquitous computing15.3 (2011): 253-269.

[35] Cederman-Haysom, Tim, and M. F. Brereton. "Bridging technical and HCI research: Creating usable ubiquitous computing." Proc. OZCHI 2004 (2004).

[36] Rogers, Yvonne. "Moving on from weiser's vision of calm computing: Engaging ubicomp experiences." UbiComp 2006: Ubiquitous Computing. Springer Berlin Heidelberg, 2006. 404-421.

[37] Saffer, Dan. Designing for interaction: creating smart applications and clever devices. New Riders Pub, 2010.

[38] Edwards, K. W., V. Bellotti, and A. K. Dey. "Newman, and MW, Stuck in the Middle: The Challenges of User-Centered Design and Evaluation for Infrastructure." Proc. CHI2003. AMC Press.

[39] Picking, Rich, et al. "Simplicity, consistency, universality, flexibility and familiarity: the SCUFF principles for developing user interfaces for ambient computer systems." (2010).

[40] Sørensen, Ruth IK. "Rethinking interaction design within ubiquitous computing." (2010).

[41] Yuan, Yuan, Xubo Yang, and Shuangjiu Xiao. "A Framework for Tangible User Interfaces within Projector-based Mixed Reality." *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*. IEEE, 2007.

[42] Harrison, Chris, Hrvoje Benko, and Andrew D. Wilson. "OmniTouch: wearable multitouch interaction everywhere." *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, 2011.

[43] Raskar, Ramesh, et al. "The office of the future: A unified approach to image-based modeling and spatially immersive displays." *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. ACM, 1998.

[44] Zhang, Shuai, et al. "Low-cost interactive whiteboard using the Kinect." *Image Analysis and Signal Processing (IASP), 2012 International Conference on*. IEEE, 2012.

[45] Molyneaux, David, et al. *Smart Object, not Smart Environment:Cooperative Augmentation of Smart ObjectsUsing Projector-Camera Systems.* Thesis, Lancaster University. 2008.

[46] Molyneaux, David, et al. "Cooperative augmentation of smart objects with projector-camera systems." *UbiComp 2007: Ubiquitous Computing*. Springer Berlin Heidelberg, 2007. 501-518.

[47] Ishii, Hiroshi, and Brygg Ullmer. "Tangible bits: towards seamless interfaces between people, bits and atoms." *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*. ACM, 1997.

[48] Wu, Ying. "An Introduction to Computer Vision." *EECS 432-Advanced Computer Vision Notes Series 1* http://www.eecs.northwestern.edu/~yingwu/teaching/EECS432/Notes/intro.pdf Last visited on 21.02 (2003).

[49] Kropatsch, Walter G. "History of Computer Vision A Personal Perspective." Institute of Computer Aided Automation, Vienna University of Technology http://www. icg. tugraz. at/Members/hgrabner/historyOfCV/kropatsch. pdf-Last visited on 05.03 (2011).

[50] Boyer, K. L., and A. C. Kak. "Color-encoded structured light for rapid active ranging." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 1 (1987): 14-28.

[51] Iddan, G. J., and G. Yahav. "3D Imaging in the studio (and elsewhere)." *Proc. SPIE*. Vol. 4298. No. 1. 2001.

[52] Rubin, Peter. "Exclusive First Look at Xbox One | Gadget Lab | Wired.com." *wired.com* . N.p., n.d. Web. 22 May 2013. <http://www.wired.com/gadgetlab/2013/05/xbox-one/>.

[53] Velten, Andreas, et al. "Recovering three-dimensional shape around a corner using ultrafast time-of-flight imaging." *Nature Communications* 3 (2012): 745.

[54] Raskar, Ramesh, et al. "Non-photorealistic camera: depth edge detection and stylized rendering using multi-flash imaging." *ACM Transactions on Graphics (TOG)*. Vol. 23. No. 3. ACM, 2004.

[55] Reeves, Byron, and Clifford Nass. *How people treat computers, television, and new media like real people and places*. CSLI Publications and Cambridge university press, 1996.

[56] "Nikon | News | Digital Compact Camera Nikon COOLPIX S1000pj." *Nikon | Home*. N.p., n.d. Web. 31 May 2013. <http://www.nikon.com/news/2009/0804_coolpixs1000pj_03.htm>.

[57] Pinhanez, Claudio, et al. "Applications of steerable projector-camera systems." *Proceedings of the IEEE International Workshop on Projector-Camera Systems at ICCV 2003*. 2003.

[58] Pingali, Gopal, et al. "Steerable interfaces for pervasive computing spaces." Pervasive Computing and Communications, 2003.(PerCom 2003). Proceedings of the First IEEE International Conference on. IEEE, 2003.

[59] Pinhanez, Claudio. "The everywhere displays projector: A device to create ubiquitous graphical interfaces." *Ubicomp 2001: Ubiquitous Computing*. Springer Berlin Heidelberg, 2001.

[60] Borkowski, Stanislaw, Olivier Riff, and James L. Crowley. "Projecting rectified images in an augmented environment." *International Workshop on Projector-Camera Systems (ProCams2003)*. 2003.

[61] Raskar, Ramesh, et al. "RFIG lamps: interacting with a self-describing world via photosensing wireless tags and projectors." *ACM Transactions on Graphics (TOG)*. Vol. 23. No. 3. ACM, 2004.

[62] Rekimoto, Jun, and Masanori Saitoh. "Augmented surfaces: a spatially continuous work space for hybrid computing environments." *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*. ACM, 1999.

[63] Raskar, Ramesh, et al. "iLamps: geometrically aware and self-configuring projectors." *ACM SIGGRAPH 2006 Courses*. ACM, 2006.

[64] Cao, Xiang, et al. "ShapeTouch: Leveraging contact shape on interactive surfaces." Horizontal Interactive Human Computer Systems, 2008. TABLETOP 2008. 3rd IEEE International Workshop on. IEEE, 2008.

[65] Cao, Xiang, Clifton Forlines, and Ravin Balakrishnan. "Multi-user interaction using handheld projectors." *Proceedings of the 20th annual ACM symposium on User interface software and technology*. ACM, 2007.

[66] Ko, Ju-Chun, Li-Wei Chan, and Yi-Ping Hung. "Public issues on projected user interface." Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems. ACM, 2010.

[67] Rapp, Stefan. "Spotlight Navigation and LumEnActive: Ubiquitous projection user interfaces." Pervasive, 2010.

[68] "MERL – Projector." *MERL – Mitsubishi Electric Research Laboratories*. N.p., n.d. Web. 31 May 2013. <http://www.merl.com/projects/Projector/>.

[69] Hong, Jason I., Scott Lederer, and Mark W. Newman. "Towards a Unified Interaction Framework for Ubicomp User Interfaces." Ubicomp 2002 conference workshop on Concepts and Models for Ubiquitous Computing, Goteborg, Sweden. 2002.

[70] Levas, A., Pinhanez, C., Pingali, G., Kjeldsen, R., Podlaseck, M., & Sukaviriya, N. "An Architecture and Framework for Steerable Interface Systems." UbiComp 2003: Ubiquitous Computing. Springer Berlin Heidelberg, 2003.

[71] Sandor, Christian, and Gudrun Klinker. "A rapid prototyping software infrastructure for user interfaces in ubiquitous augmented reality." Personal and Ubiquitous Computing 9.3 (2005): 169-185.

[72] Molyneaux, David, and Hans Gellersen. "Projected interfaces: enabling serendipitous interaction with smart tangible objects." Proceedings of the 3rd International Conference on Tangible and Embedded Interaction. ACM, 2009.

[73] Geethapriya, J., and A. Srinivasan. "A New Framework for Real-time Hand Gesture Detection and Recognition." in International Conference on Information and Network Technology (ICINT 2012), IPCSIT vol. 37 (2012)

[74] Han, Jefferson Y. "Low-cost multi-touch sensing through frustrated total internal reflection." *Proceedings of the 18th annual ACM symposium on User interface software and technology*. ACM, 2005.

[75] Xu, Huichuan, et al. "User interface by virtual shadow projection." *SICE-ICASE, 2006. International Joint Conference*. IEEE, 2006.

[76] Linder, Natan, and Pattie Maes. "LuminAR: portable robotic augmented reality interface design and prototype." Adjunct proceedings of the 23nd annual ACM symposium on User interface software and technology. ACM, 2010.

[77] Hwang, Gwo-Jen, Chin-Chung Tsai, and Stephen JH Yang."Criteria, strategies and research issues of context-aware ubiquitous learning." *Educational Technology & Society* 11.2 (2008): 81-91.

[78] Jones, Vicki, and Jun H. Jo. "Ubiquitous learning environment: An adaptive teaching system using ubiquitous technology." *Beyond the comfort zone: Proceedings of the 21st ASCILITE Conference*. 2004.