A Tangible User Interface Using Untagged Physical Objects for Information Management

Wong Ho Ying
(Master's Program in Computer Science)

Advised by Jiro Tanaka

Submitted to the Graduate School of
Systems and Information Engineering
in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering
at the
University of Tsukuba

March 2011

Abstract

To retrieve digital contents users must reach the link by some searching operation, a task that requires some memory effort and is challenging due to the large amount of contents required for daily work. The conventional solution is keeping bookmarks and shortcuts for relevant contents. An interesting approach is by using Tangible User Interface (TUI), which enables users to interact with computers and information using physical objects.

Tangible User Interface is greatly used for augmented reality system and table-top interaction. Such systems use sensors, readers, large displays and usually designed for specific environments, some of which have complex deployment and not accessible to common users. Conventional TUI systems allow interaction with physical objects which were previously modified to suit the intended purpose. There are researches where additional computer-readable tag identifiers are used to avoid limiting the range of interactive objects. However tagging system also implies previous tag configuration and objects become dependent to such of tags for interaction.

In this work, we present a TUI approach that uses multiple unmodified physical objects surrounding the user to interact with the computer, and to accomplish simple interaction tasks, more specifically accessing information and basic control mechanism for a personal computer. To simplify the deployment, we avoid tagging our interactive object. Each object is identified by its appearance, which is translated in to a set of features, including the color distribution and the approximate shape-size of the object regardless of rotation. Our system Tangible Linker uses the proposed interaction method to support quick access to digital contents using physical objects on user's real desktop. Objects are capable to interact between them to filter out contents with related information. Additionally interactive objects can serve as control point for simple commands on currently opened windows on the desktop environment. In this work, results of evaluation on our object recognition method and user experience using Tangible Linker for information accessing are presented.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

1.1.1 Information management

Personal information management describes the collection, storage, organization and retrieval of digital objects such as files, emails, bookmarks, etc, by an individual in their personal computing environment [1]. In such of environment, to retrieve digital contents users must reach the link by means of some searching operation, a task that requires some memory effort and is challenging due to the large amount of contents required for daily work. The conventional solution is keeping bookmarks and shortcuts for relevant contents which is based on the concept of WIMP (Windows, Icons, Menu, Pointing device) interaction paradigm. This paradigm had gained a remarkable degree of commonality over decades; however, as the electronic devices had been evolving out of the usual computer hardware system, researchers have been searching for new interaction paradigm to accomplish different computer system. Dam [2] called Post-WIMP user interfaces to those that do not use menus, forms or toolbars but relies on other operation methods that match to human nature abilities. Tangible User Interface [3, 4] is one of such methods, which make possible the manipulation of the digital world in physical domain.

1.1.2 Tangible User Interface

Tangible User Interface (TUI) corresponds to the user interface in which a person interacts with digital information through the physical environment. It provides physical form to digital information and computation by embodying it into physical objects. It can be used to represent digital information, mechanisms for interactive control, even the digital state of a system.

Previous researchers [5] had defined three types of physical objects to represent digital information: *container*, *tokens* and *tools*. They refer as *token-based interactions* where physical objects (the tokens) are used to access stored information, the nature of which is physically reflected in the token in some way. Tokens-based interactive system does not digitally store the information into the token but it acts as trigger to show the information through a display device. Additionally, the combination of tokens can be used to access information associated with all the tokens simultaneously. It might require some technique of key generation to associate the tokens in question.

1.1.3 From the Physical to the Digital

One of the design problems of Tangible User Interface system is how to map the physical objects and their virtual counterparts. It refers to the method used to identify the object itself to the system and enable user to access the associated information or operations. This identification consists in some special characteristic attached on the object which is readable by the system. Ailisto *et al.* [6] in his work Physical Selection classified the task into three alternative computer-readable identifiers:

1. Visual code: such as common barcode [7, 8], QR code [9](two-dimensional bar code), 2D pattern markers [10](simple black square tracking code), etc., where detection is achieved using laser scanner, and the latest one by camera using computer vision technologies (such as ARToolKit Library).

2. Electromagnetic technologies: Radio Frequency Identification (RFID) tag is greatly employed by researcher for TUI interaction due to its small size and cheap price. Its detection consists in proximate vicinity of a tag to the reader.

3. Infrared technologies: IR transceivers are capable to send and receive infrared data, which is dependent on the distance, power and alignment of both devices.

Using each one of these options for user interface implies previous tag configuration work such as the generation and registration of such identifier (tag) and its attachment to the physical object. Additional reader devices capable to translate this identifier in to it equivalent digital form is also required.

1.2 Problem

TUI systems involve mostly computational systems that use sensors, readers, large displays and usually designed for specific environments [7–11], some of which have complex deployment and most of them are under research. Moreover, TUI has also become popular for tabletop system[12–14]. On the other hand, conventional TUI systems allow interaction with physical objects which were previously modified to suit the intended purpose. This modification implies some kind of alteration of the object appearance or attachment to an additional part to identify itself in the system. Usually this task corresponds to the designer or developer and it is invisible to user. End users do not have control over what objects to use and what information to be mapped.

With the goal to provide higher usability on TUI, one of the purposes of this research is to make TUI available to accomplish common computational task such as accessing digital information in commercial personal computer (desktop and laptop). Our systemTangible Linker works in such of systems and aims users to access information using surrounding physical objects. Moreover, end users have significant control over the set of objects to be used and choose the domain of information for such purpose.

1.3 Proposal

In this work, we focus on TUI appliance for embodiment of digital information and interactive control device. We present a TUI approach that uses multiple unmodified physical objects surrounding the user to interact with the computer, and to accomplish simple interaction tasks, more specifically accessing information and basic control mechanism for a personal computer. To simplify the deployment, we avoid tagging our interactive object but instead, we use computer vision for object detection and recognition so no object modifications or tag setup are required. Each object is identified by its appearance, which is translated in to a set of features, including the color distribution and the approximate shape-size of the object regardless of rotation.

We implemented a system called Tangible Linker that supports quick access to digital contents using physical objects on user's real desktop. Our system allows users to map different kinds of digital items, regardless their sources, with physical objects and to access directly the desired contents without requiring any search operation. Additionally the combination of multiple objects allows filtering of information associated to them using object's profile as query criteria. Finally interactive objects can serve as control point for simple commands on currently opened windows on the desktop environment. To switch between control point and information access function is achieved by hand gesture.

1.4 Contributions

The contributions of this work can be summarized as follows:

1. A method for recognition of untagged physical objects over the existing color-based recognition methods. Beside color histogram, we use in addition the covariance matrix on the histogram plus the object image dimension.

2. A tangible interaction method for information accessing and control mechanism. Also a multiple object interaction method for information filtering is presented.

3. A Tangible User Interface to accomplish common task on personal computer such as information management using untagged physical objects.

1.5 Organization

This thesis is organized as follows. In Chapter 2 we present the related works. In Chapter 3 we outline the details about our proposal. In Chapter 4 we introduce our system Tangible Linker which put on practice above interaction method, while in Chapter 5 we describe our object recognition implementation details. Evaluation and the results are discussed in Chapter 6. Finally, Chapter 7 presents the conclusions and future works.

Chapter 2

Related Work

2.1 Tangible User Interface

Ishii's Tangible User Interfaces[3][4] and physical browsing by Valkkynen *et al.* [15] describe the possibility of interaction between human and digital information through physical objects. In Ishii's Tangible bits[3][4], they seek to give a physical form to digital information, which is more manipulable and perceptible by human natural abilities. Valkkynen *et al.* [16] proposed to map digital information onto physical objects in human environment under an ambient intelligence scenario.

2.2 Physical Embodiment of Digital Information

An early attempt to merge the physical and digital world was DigitalDesktop by Wellner [11]. They augment the digital working space to a physical desktop where physical paper serves as electronic documents and the interaction with papers is by means of bare fingers. Based on the above ideas, Siio in his work IconSticker *et al.* [7] presents a metaphor very close to this concept. Icon-Sticker is a paper representation of a digital content. It consists on transferring the icons from computer screen to paper, so they can be handled in the real world and used to access directly to the digital content. An icon first is converted into its corresponding barcode, which is printed on a sticker. This way the sticker can be attached to any physical object. To access the icon, user scans the barcode on the sticker with a barcode scanner.

Online resources recovery also had attracted the attention of researchers. WebSticker by Ljungstrand *et al.* [8] and AURA [17] both propose the usage of barcode to represent an online information. WebSticker [8] shares the same analogy with IconSticker [7] although instead of icon they manage web bookmarks, while AURA by Bush *et al.* [17] takes advantage of already existing barcode on surrounding objects. They use handheld device with barcode reading function to capture the input and display related information.

Beside barcode, several researches adopted other type of visual code such 2D barcode that is a small pattern marker greatly used for augmented reality system. CyberCode [9] is a visual tagging system based on this 2D barcode technology. CyberCode tags can be recognized by the low-cost CMOS or CCD cameras found in more and more mobile devices, and it can also be used to determine the 3D position of the tagged object as well as its ID number. Another example is Iwabushi's Natural Storage [10] which uses such of black and white marker to identify user and retrieve files associated to the user.

Above mentioned systems are based on visual coding to identify the physical element. Other than visual code, Want *et al.* [18] proposes to augment everyday objects via embedded RFID tag. Pradhan [19] uses Infrared beacons, RFID or barcode on physical entities in the environment and map them to a web browser. Ailisto [20] takes advantages of the IR transceiver on mobile devices to discover smart physical objects. Sensetable by Patten *et al.* [13] can electronically tracks the position and orientation of multiple wireless objects on a tabletop display surface. Their system uses two adjacent Wacom Intuous sensing tablets for sensing surface and modified mice (pucks) for interactive physical objects which can be sensed by the tablets and sensitive to touching.

There were also attempts to avoid tagging on physical objects for a more natural interaction. EnhancedDesk by Nishi *et al.* [12] can register real objects on user's desktop based on its appearance when a user indicates a region on the desk by making a snapshot gesture with four fingers for tabletop system. Color histogram is used to model the object and pointing gesture is used to trigger the recognition. EnhanceTable also use RGB color histogram to model the object. The size is predefined, limited on mobile phone recognition. TaPuMa by Mistry *et al.* [14] implemented a Tangible Public Map to allow user input searching query using their own belongings or everyday objects to access relevant information. Using Sensetable technology for sensing surface, objects are embedded with sensible tokens for position and orientation detection. Additionally they uses computer vision to detect and track objects. Objects images are compared against a predefined set of objects in the database.

Although many previous researches use physical objects to access information, most of them are token-based interfaces which limit the use of surrounding real objects in tangible interactions to some specific object only. In our system, we use computer vision for object detection and tracking to avoid all kinds of computer readable tags to simplify the interaction and allow any object to become interactive. Also those systems mentioned above consist in some predefined platform and domain of information. Therefore, we intend to return this control to the end users allowing them to define what is more relevant to be represented by the set of objects they chose. Finally, we add another implicit property on the object which is defined by the information associated to it so that can be used for further information searching purpose.

Chapter 3

Tangible Interface for Information Management

In this research, we propose a tangible solution to organize and access digital information using physical objects. In order to enable any surrounding objects becoming interactive with our system at any moment, we avoid any kind of previous configuration or identifier attachment on the object. Instead, objects are detected and recognized as its appearance so users can simply grab the object available in the moment when it is needed.

In contrast to previous research, we do not predefine the domain of information that a user can work with. In our system, users can freely bind and unbind items like Icon's and url in the conventional Graphical User Interface to the selected object. Physical objects do not contain physical memory to store digital data but act like trigger to the digital items. When an object is bound to a digital item, the system automatically extracts the relevant keywords that describe the content and add these to the object's profile. This profile helps to filter related items when multiple objects are detected. We also extended the usage of physical objects in desktop environment by empowering it into control point of applications, using its physical location information mapped in the digital desktop.

3.1 Interaction Technique

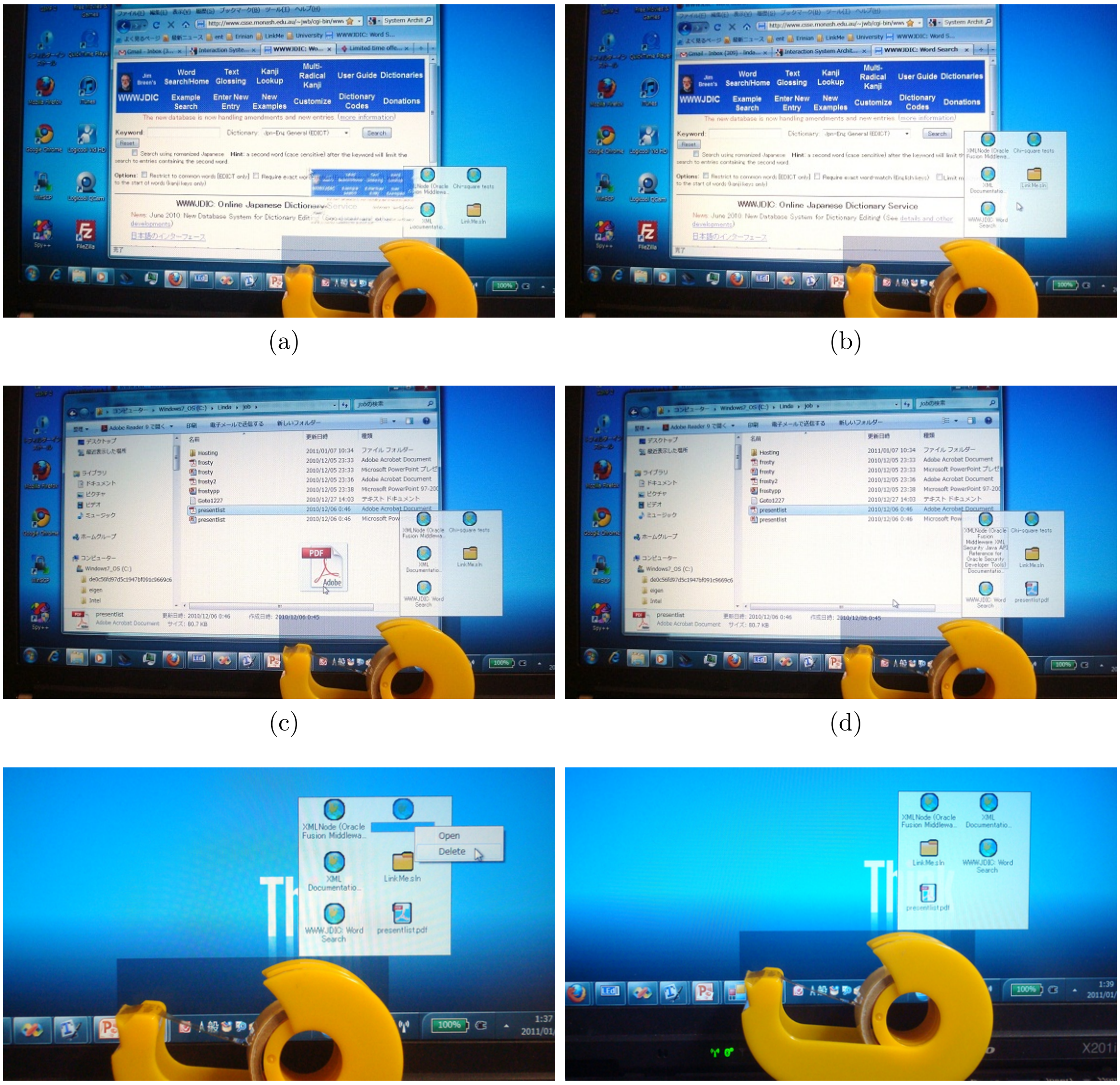
Intuitiveness is our main concern. Consequently our computer vision based object detection system uses the concept of sensing display screen. The reason of such selection is that computer screen is the place where user's attention is focused on, and it is perceived visually without specific indicators or markers. Unlike most tabletop tangible interface system, physical objects cannot be placed on any surface of a desktop screen. Therefore, our sensing surface is limited to the bottom half of a conventional screen. When an object is placed within the sensing area, it is detected and its physical position on the screen is mapped into the digital desktop. Combining with other data, the system determines the digital representation of the object, which is the portion of the digital desktop located behind the physical object. We define it as the drop in area when binding digital items to the object. This way, the object being in the physical space is integrated virtually into the digital environment. Then a corresponding visual feedback to given to the user on the display screen.

3.1.1 Binding and Unbinding Information

Once a physical object became into a digital item on the desktop environment, user can use drag-and-drop operation to bind references with the object. Drag-and drop operation is easy and intuitive for most computer users and it is a native operation supported by the operating system. Derived drag-and-drop controls such as multiple dragging, control button dragging (shift and ctrl button for items selection) is also supported.

Any draggable item in the digital desktop can be bind to the object as reference by dropping it on drop in area which is the digital representation of the object. In Figure 3.1(a) the URL of a website opened in Internet browser tab is being dragged toward the object. After dropped on the drop in area of the object, it is added to the reference list of the object as shown in Figure 3.1(b). In Figure 3.1(c) user binds a file icon from file explorer with the object and Figure 3.1(d) shows the result.

To unbind a content, user can select the item references from the reference list of the object and choose DELETE from the submenu by right-clicking the targets. The operation method is very similar to conventional graphical file browser. In (Figure 3.1(e)) user selected the reference he wants to delete and prompted the submenu to access delete function. If deletion was succeeded, the result is reflected immediately in the reference list as shown in (Figure 3.1(f)).

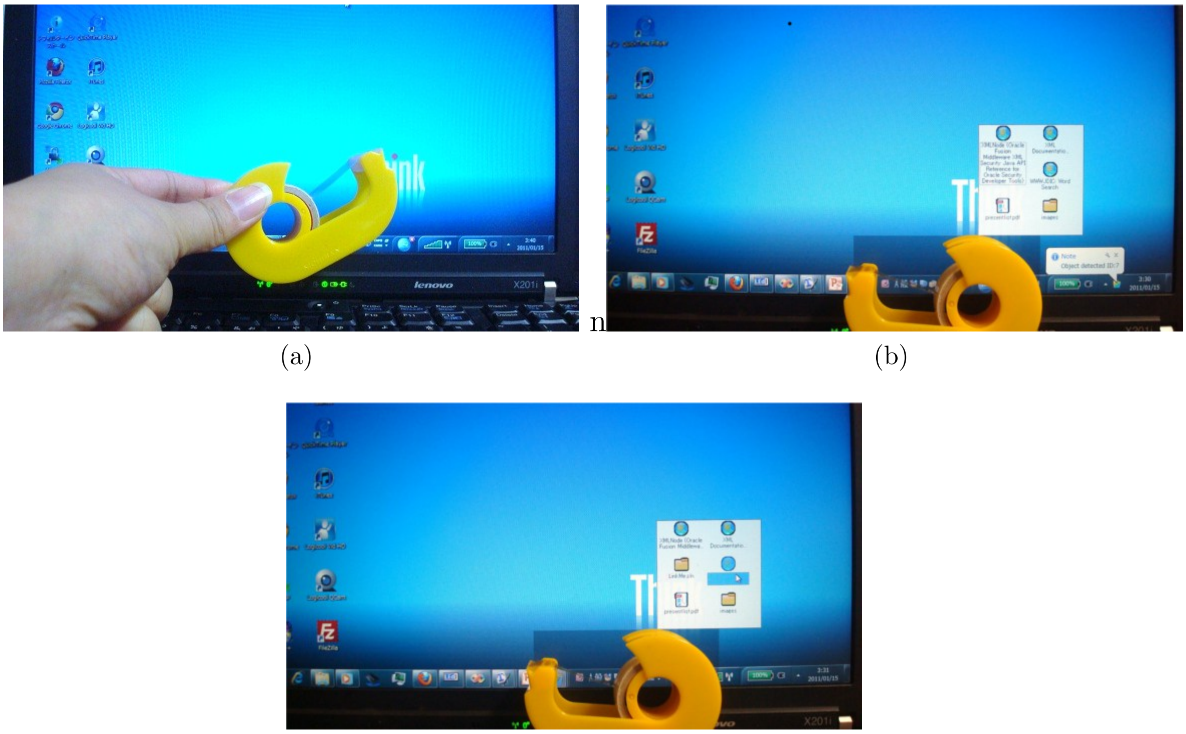


(a) (b) (c) (d) (e) (f)

Figure 3.1: Binding and unbinding information: (a) the URL from an Internet browser tab is being dragged and (b) shows the resulting reference list after binding, (c) a pdf file from file explorer is being dragged and (d) shows the resulting reference list after binding, (e) a reference is being deleted and (f) shows the resulting reference list afterward.

3.1.2 Accessing Information

To retrieve the content in an object, it is simply by placing the object back to the sensing area as shown in Figure 3.2(a). Once it is detected, the associated digital item references are shown in a form widget next to the physical object as shown in Figure 3.2(b). From this reference list user can access the content of the references. In Figure 3.2(c) a reference is selected and to open it, user can either double clicking it or use the submenu to open more than one reference simultaneously. By removing the object from the sensing area, it indicates that user has finished interacting with the object therefore the form closes automatically without requiring any additional operation.



(a) (b)

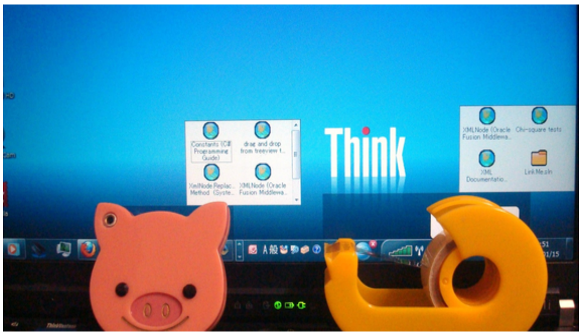
(c)

Figure 3.2: Accessing information using physical object: (a) an object is placed in the sensing area, (b) shows the reference list of the detected object, (c) shows a target reference is being select to open (or delete as well).

3.1.3 Multiple Objects Interaction

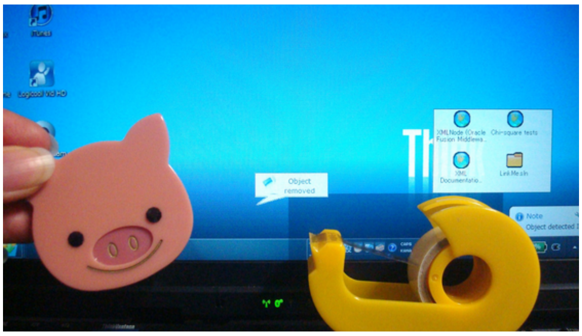
We use the metaphor of physical proximity of two objects to indicate interaction between 2 objects. When 2 objects are spaced out within a close distance in the sensing area, it triggers the references filtering function and gives back the related items between these two objects.

Figure 3.3 shows the interaction of two objects. Each object has a reference list with the associated digital items. When there is a significant distance between two objects, only their reference list is shown beside them (Figure 3.3(a)). However, when an object is placed next to another one (Figure 3.3(b)), the filtering function is set active and those related references are enlisted in a separate reference list form. When any of these objects is being removed (Figure 3.3(c)), its own reference list as well as the filtered reference list disappear.



(a)

(b)



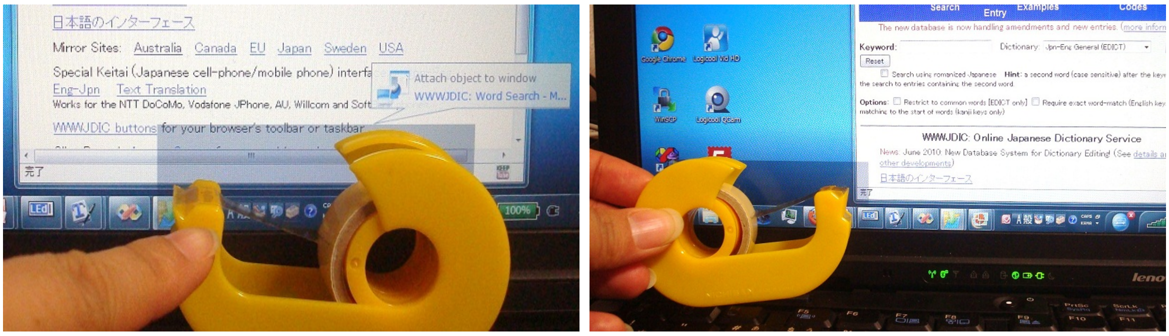
(c)

Figure 3.3: Interaction between two objects: (a) two objects are present simultaneously, (b) one object has been moved next to the other one, (c) one of the object has been removed.

3.1.4 Object Tracking

As the physical position of the object is used for system GUI, by changing this position can trigger control operation on the application which it is associated to the object. Given that our sensing area is limited to the lower portion of the screen, the movement tracking is limited to left and right direction. We use the physical proximity of the object regarding to the digital window to establish such association.

Initially, a detected object is served as trigger to bound digital item. It is possible to switch to control mode, which allows user to submit operative command on currently active windows in the desktop environment. Switch operation is accomplished by causing a short interference on the object image, such as touching the surface of the object (see 4.6(a)), without changing its position and orientation. When it happened, the object is attached to the active window behind, which area is partially occluded by the object with higher z-order (the lowest z-order corresponds to the front most window in window manager). The system then is notifying the user what window was attached. Whenever the user changes the physical position of the object within the sensing area as shown in Figure 4.6(b) where user is dragging the object to the left, an action is performed and reflected on the attached window. To detach an object from attached window, user simply remove the object, or perform the switch operation as shown in 4.6(a).



(a) (b)

Figure 3.4: Control mode and tracking: (a) switch operation, (b) user is dragging an object in the sensing area.

Chapter 4

Tangible Linker

We developed a system to demonstrate the usage of Tangible User Interface for common desktop task named Tangible Linker. Through our system , physical objects aim user to:

1. Organize and access digital information
2. Serve as control point by physically manipulate the object

In our system, the bottom half of display screen is the sensing surface. Nevertheless, it is extended seemingly to the real world when a physical object is placed in front of the screen, which becomes part of the desktop environment acting like an interactive element. The screen portion occluded by the object represents the physical object on the desktop environment. This screen region serves as an addition destination spot for ordinary drag-and-drop operation supported by the operation system. Therefore, when user selected one or multiple items, it is possible to drag-and-drop them into the object-occluded screen region to map digital items to the object. A sound feedback is given when the mapping has been succeeded. Also user can move their mouse pointer to the object-occluded screen region and click on it to display the list of mapped items. Remove physically the object from the sensing area when finished working with the physical object.

To retrieve the associated information, user simply places back the object and the list of mapped digital items will display automatically. Both mapping and retrieving digital information did not imply extra workload to user (barcode generating, printing, browsing, etc.).

4.1 System Architecture

4.1.1 Hardware

In order to keep the deployment as simple as possible, the hardware setting consists in a laptop computer and a web camera. To avoid limiting interactive object to specific configuration, our system uses computer vision approach to detect the presence and extract the features of the target object. Figure 4.1 shows the hardware setup and the some key component of the system. A USB camera is constantly looking at the display screen. Anything other than the computer is interpreted as possible object. To ensure that an object is captured by the camera when it is intended to be, we establish an implicit sensing area which is the bottom half of the display screen. This constraint has two advantages:

1. User convenience: computer screen is the place where user attention is focused on and it is perceived visually without specific indicators or markers. This setup gives the impression of the object being another part of the user interface, seemingly to extend the digital desktop into the physical world and vice versa.

2. Technical reason: it helps to ensure the sensing area and it eases the visual integration of object to desktop environment.

Additionally, a polarized filter is placed before the camera (see Figure 4.1) which aims to eliminate the noises from the screen activities, leaving an images that contains the computer as background and the physical object. The "shadow" projected (occluded area) by the object on the screen is called drop-in area, which is the object digital representation and drop point for drag-and-drop operation. In Figure 4.1 it corresponds to the orange area behind the physical object. The reference list containing the bound digital items is visualized on the display screen next to the drop-in area, that from the user perspective, next to the physical object.

4.1.2 Software

The system was implemented in Microsoft Windows platform. Current version works with 2 drag-and-drop supported applications, which are file explorer and Firefox Internet browser. The object recognition module was implemented in Visual C++

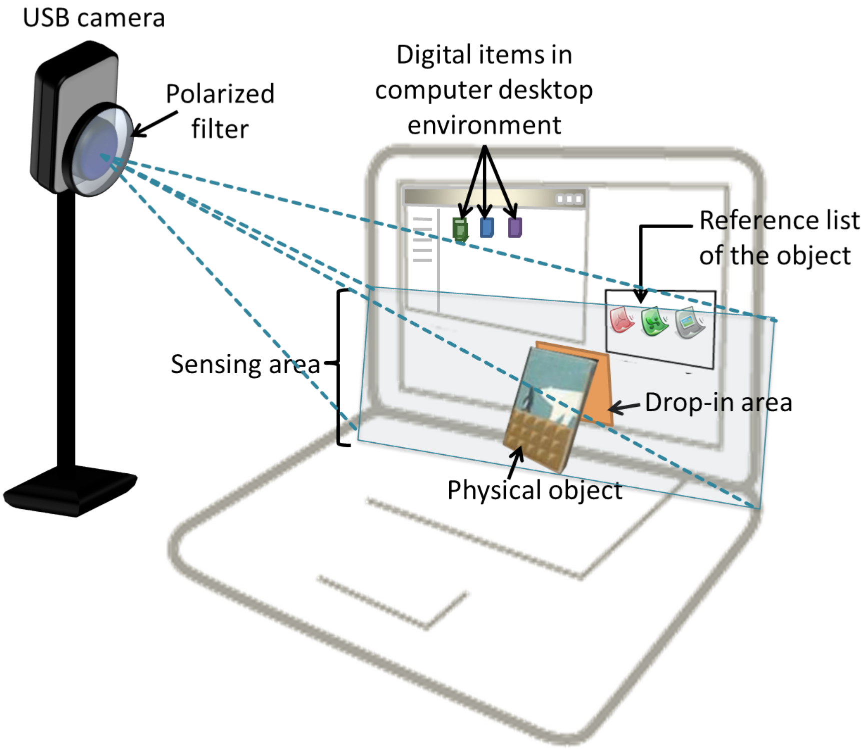


Figure 4.1: Tangible Linker's hardware components

2008 and OpenCV 2.2[21][22] image processing library. The graphical user interface was implemented in Visual C# 2008. Even though in current version, the objects and corresponding database is stored locally, the idea of implementing a networked version making available the database from different workstations and make use of shared data, for example in a cloud computing environment is considered y left for future work.

The entire system consists in two modules: Recognition and Linker. (Figure 4.2) The task of the Recognition module is to process the input image identifying the object found in the sensing area and the state of the object. It extracts the object from an input image, creates the object model and compares it against those entries in the object database. As result, it sends via socket to Linker module the ID of the object along with other data such as screen location, dimension as well as the state of the object. There are three states: 1) object detected: when an object is seen newly, 2) object removed: an object disappeared from the registered location

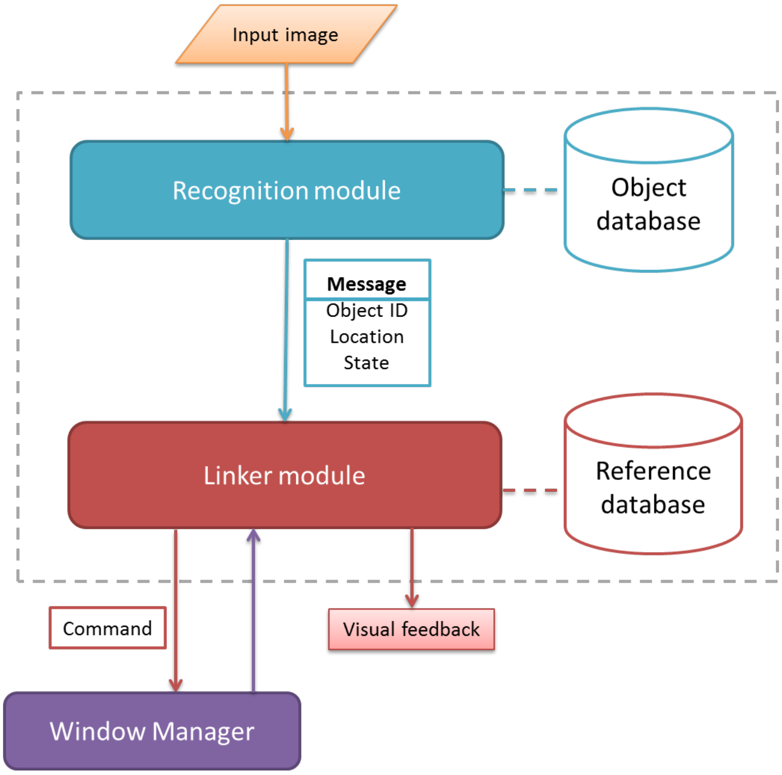


Figure 4.2: Diagram of Tangible Linker architecture

for certain time, and 3) object updated: an object disappeared from the registered location but re-detected within certain time. Linker module then takes the message from the socket and produces the appropriate responses for user. It administrates the reference database, which host the information of the digital items references bind to each registered object as well as their profile. Depending on the state of the object, it retrieves information about the currently active windows from operating system's window manager and submits operative command on the target window. Also it manages all the Graphical User Interface of the system.

The followings describe our system functionality.

4.2 Organize and Access Digital Information

As presented in our previous work [23], Tangible Linker uses a conventional web camera which is facing the computer screen (see Figure 4.3). Any object placed in front of the screen is detected by the system. The system will then try integrating the object into the digital desktop so user can:

1. Map digital information to it using drag-and-drop operation

2. Access information previously associated with the object

Our system allows multiple contents per object, regardless their sources. As there is no physical memory on the objects, the capacity is only limited by the computer hard drive.

Figure 4.3: Setup of Tangible Linker

4.2.1 Use Case

The interaction starts when the user places an object in front of the screen (the sensing area). This setup gives the impression of the object being another part of the graphical user interface, seemingly to extend the digital desktop into the physical world and vice versa. To bind a digital item with an object, the user places the object in front of the screen (Figure 4.4(a)), and drags the item toward the direction of the object and drop it in the drop-in area (Figure 4.4(b)).

Next time, when the same object is placed in the sensing area, the system automatically recovers the list of bound references and display them on the screen next to the object (Figure 4.4(c)). User then can access these references by clicking on the it

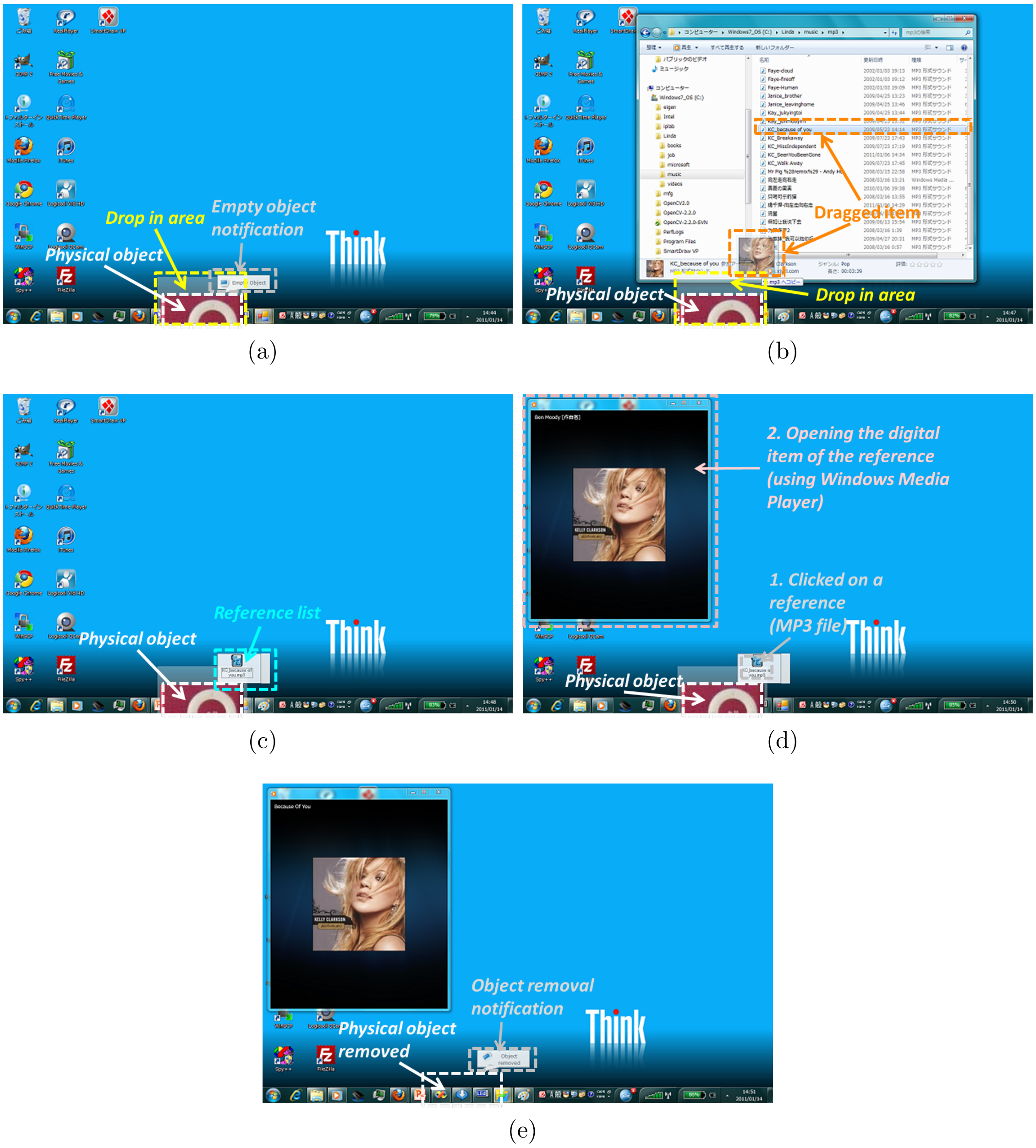


Figure 4.4: General operation of Tangible Linker: (a) an object is placed in the sensing area, (b) user drags 2 music list files toward to object and drop them in the drop-in area, (c) when the object is detected again, the reference list will be shown automatically, (d) user clicks on the reference to open the content, (e) the object is removed and the drop-in area along with the reference list disappear from the screen.

(Figure 4.4(d)). Once the object is no longer needed, remove it from the sensing area and Tangible Linker interface will close automatically (Figure 4.4(e)). Please refer to Appendix A Figure A.1 for more details of the operation flow when object is newly detected.

4.3 Filtering of Digital Information

Physical objects are capable to interact between them through the system. A single object is associated to a subset of digital items. The combination of multiple objects can represent a new subset of information, or serve as a filter to screen out relevant information. In Tangible Linker, filtering is performed on the profile of the object. Object profile describes the content of the associated items. Whenever a digital item is bound to an object, the system will extract the keywords related to the content of the item. These keywords are added to the item description and included in the object profile. The method used for keywords extraction varies regarding to the digital item type. There are 2 types of item:

1. URL path string for Internet resources: we use Yahoo! Query Language (YQL)[24] to extract the keywords from the content of the resource pointed by the URL string. YQL is an expressive SQL-like language that allows user to query, filter, and join data across Web services. To access the YQL Web Service, we use a web client in C# can call HTTP GET, passing the YQL statement as a URL parameter. When it processes a query, the YQL Web Service accesses a data source on the Internet, transforms the data, and returns the results in XML. We then read the result XML extracting the list of keywords according to the criteria of this Web Service.

2. File/folder path string for local file system resources: although it is possible submitting a plain text into the URL parameter to access YQL Web Service so that can return the list of relevant keywords that identify the text, we found that a limit number of words usually results a null list. For that reason, we use instead an open source library called Semantic Library [25][26] to extract the keywords from the name and path of the file. The library receives the input text and decomposes it into a predefined number of possible keywords

order by relevance. Currently, keyword extraction is limited to textual content. Multimedia content such as picture, sound and video are not considered for keywords check.

Each item is described by its own keywords. These keywords are compared to the profile keywords list of the object. The object profile then is updated inserting those new keywords which are not in the object profile.

Every time an object is newly detected or updated, Tangible Linker check whether there is another object already being used in the system and compare their location. If the distance is below a threshold, their profile keywords are compared to find the common keywords. Those items described by these common keywords correspond to the intersection of all the items from both objects, which are given back to the user in a separated reference list.Figure 4.5 shows the result of items filtering function on two objects. The pig-shape mirror has two URL items where two of them contain information about XML. On the other hand, the tape dispenser also has one URL items and a folder item about XML. When these objects were placed together, system found the entry "XML"as common keywords, therefore these items are included in the common reference list.

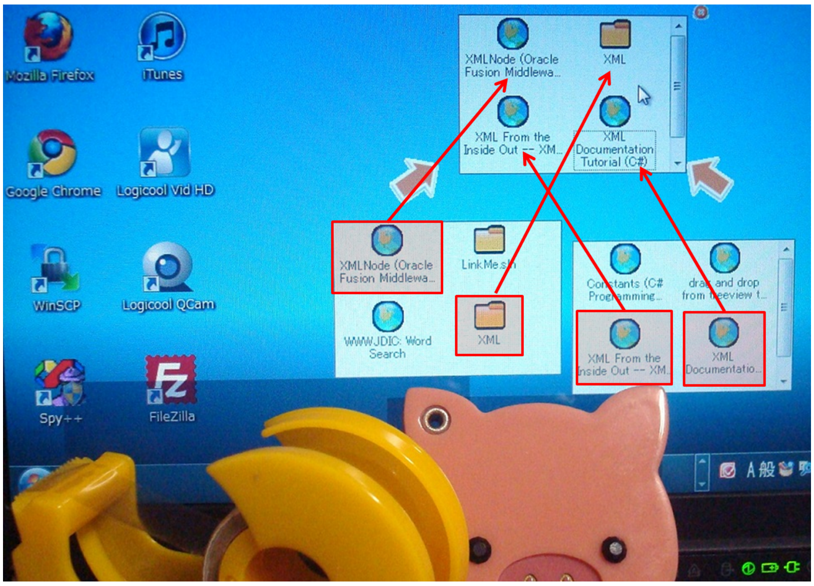
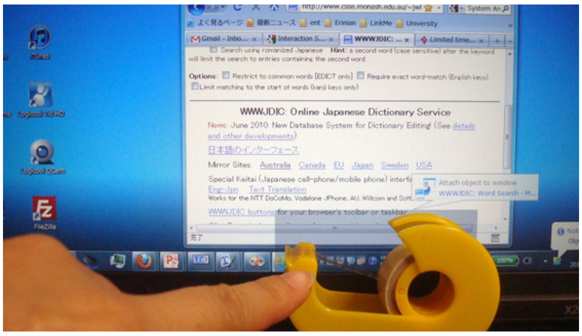


Figure 4.5: Filtering of related items. The red boxes enclose those related items by the keyword "XML".

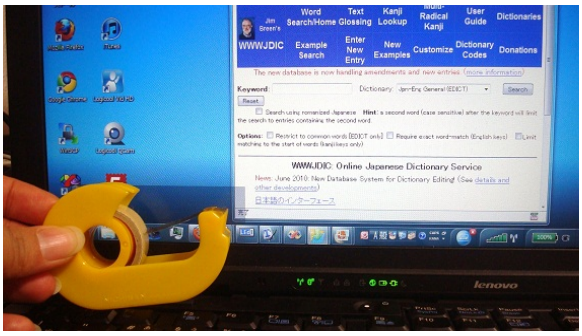
4.4 Control Point Over Active Windows

Physical objects can be attached to currently active windows in the digital desktop and submit control operative on this specific window. It is executed when Linker received an object updated notification from Recognition module (Please refer to Appendix A Figure A.2 for more details of the operation procedure on object updated). By touching the object for no more than 5 seconds, user is attaching it to the window that is partial occluded by the drop-in area of the object. As only one window can be attached to an object, in case where object's drop-in area is causing occlusion on more than one window simultaneously, the one with the highest z-order is chosen. If succeeded, the reference list of the object is being hidden (if shown) and a message is prompted up notifying the switch to control mode along the name of the window attached to the object. Then user can move the object across the horizontal axis of the sensing area to submit control command. The difference between the new location and the latest one is compiled into the corresponding action over the attached window. These actions are variable according to the application. The default actions is simulating the page up (when object is moved to left) and page down (when object is moved to left) key to move around within the visualization area of the window. Special case is considered for sound playback application such as Window Media Player where the actions are volume up and down respectively. To detach the object from window, user can touch again the object, or simply remove the object from the sensing area. In both cases, the corresponding notification message is shown.

In Figure 4.6(a) user is switching to control mode by touching the object. There is an active window of Internet browser that overlap with the drop-in-area of the object, thereby the object is attached to it and a small notification message is shown. Then user is dragging physically the object along the sensing area, causing different action on the Internet browser. In Figure 4.6(b) user dragged the object to the left, causing the website go page up and in 4.6(c) user dragged the object to the right, causing the website go page down. To detach an object from attached window, user simply remove the object, or perform the switch operation as shown in 4.6(d).

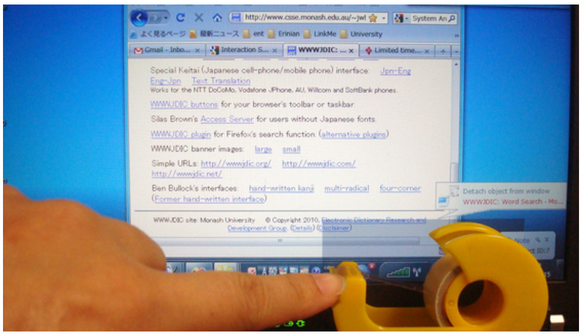


(a)



(b)

(c)



(d)

Figure 4.6: Control mode and tracking: (a) an object is being attached to the window behind, (b) user is dragging an object to the left, (c) user is dragging an object to the right, and (d) user is detaching the object from attached window.

Chapter 5

Object Recognition

In this chapter we are describing our approach to make available physical objects for interaction.

The core of our system consists in interacting with physical object. Recognizing objects on demand by computer vision techniques triggers several problems regarding to image processing. First, given an input image, determine whether there is an object placed in the sensing area. Second, the proper representation of the detected object in the system must be computed, and third, whether the detected object matches any other in the system database. Our Recognition module takes an input image and classifies its pixels into either background pixels or foreground pixel. The resultant set of foreground pixels is considered as raw representation of the object and it is transformed into a representation used by the system to model the object. Finally, this model is used to compare the input object with those in the database to find a match, if any.

When multiple objects are involved for interaction, it is necessary to identify all of them properly. For computer vision based recognition, the closeness and overlapping between objects might cause problem for recognition given that they might be treat as a single big object. Therefore, we decompose this big-object into their correspond objects comparing the big-object contour with those objects which were detected previously and marked active. The remained unmatched portion of the big-object corresponds to the newly placed object.

5.1 Background Subtraction

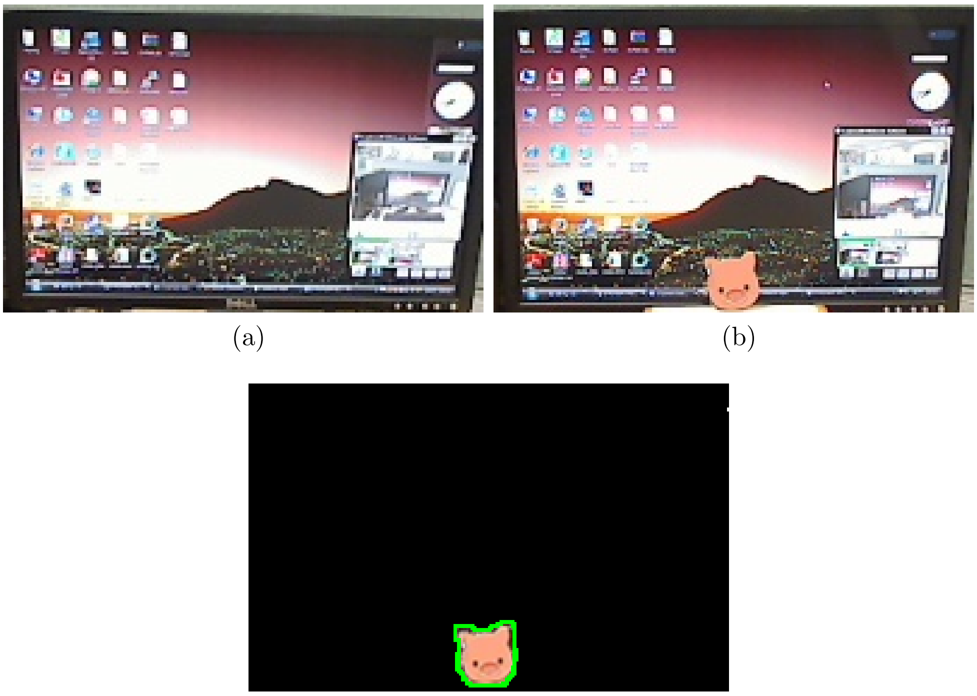
Given the constant activities on the computer screen and others environment noise (i.e., user typing), variation of the background is presented even the camera is capturing from a static position. Therefore, the background must be learned in real time, so the result from comparison between input images with the computed background is relevant.

Modeling background is the basis of a background extraction algorithm. There are several methods to model non stationary background. Adaptive Gaussian Mixture (AGM) [27] is widely used and it is robust under most situations. However, this technique is memory consuming and high computing time since it needs to maintain and update a Gaussian distribution for each pixel.

Instead, we choose Approximated Median Filter (AMF) [28] to model the background. As a recursive technique, it has the advantage of being simple; no buffer for previous frames is required because it recursively updates a single background model based on each input frame. The modeling algorithm is based on the estimation of median value of each pixel, which is incremented by one if the input pixel is larger than the estimate and decreased by in the contrary case. This estimate eventually converges to a value that is the median for each pixel. We present results of background subtraction to produce the foreground objects in Figure 5.1, including the initial background image (Figure 5.1(a)), an object on the capturing area (Figure 5.1(b)) and the background removed producing the actual foreground objects' images (Figure 5.1(c)).

5.2 Object Detection

For every input image, we compare it with the background image looking of differences. Each difference found is then translated into contour. We filter those contours with its area smaller then a predefined threshold, remaining those with area big enough to be candidate physical objects.



(a) (b)

(c)

Figure 5.1: Background subtraction: (a) background image, (b) input image, (c) background removed.

5.3 Objects Comparison

Human vision perceives an object by its physical configuration. However, this information is not so intuitive for the computer. To transform the physical appearance of the object into digital form, we use 3 types of properties to model an object:

1. Average and covariance matrix of the color distribution (in HSV color space)
2. Color histograms for Hue and Saturation channels of the object image (in HSV color space)
3. Object image bounding box geometry, including the relative dimension and side proportion

The first two properties determine the color distribution of the object and the last one determines the approximate shape and size of the object regardless of rotation and scale.

Detected objects are kept in a database including their features, and using an object likelihood function, a newly detected object is compared with those already known for recognition. Given the mentioned model, we compute the distance (difference) between 2 objects using the following metrics:

1. Mahalanobis distance[22, 29] ($d_{\mathcal{M}}$): Distance between the normal color distributions of two objects i and j. This distance is:

$$d_{\mathcal{M}}(i,j) = 1 - \sqrt{(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T \left(\frac{\boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_j}{2} \right)^{-1} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)} \tag{5.1}$$

   where $\boldsymbol{\mu}$ is the color mean vector and $\boldsymbol{\Sigma}$ is the color covariance matrix, both in HSV color space, for the given object.

2. Color Histogram distances ($d_{\mathcal{H}}$ and $d_{\mathcal{S}}$): 2 independent distances between two objects' color histograms using OpenCV implementation of Bhattacharyya's [22, 30] histogram comparison. $d_{\mathcal{H}}$ is:

$$d_{\mathcal{H}}(i,j) = 1 - \sqrt{1 - \sum_{k}^{N} \left(\sqrt{\boldsymbol{H}_i(k) \cdot \boldsymbol{H}_j(k)} \right)} \tag{5.2}$$

   where $\boldsymbol{H}_i$ and $\boldsymbol{H}_j$ are two normalized histograms with N bins, for the Hue channel of objects i and j respectively.

   The distance $d_{\mathcal{S}}$ for saturation is defined similarly, as:

$$d_{\mathcal{S}}(i,j) = 1 - \sqrt{1 - \sum_{k}^{N} \left(\sqrt{\boldsymbol{S}_i(k) \cdot \boldsymbol{S}_j(k)} \right)} \tag{5.3}$$

   where $\boldsymbol{S}_i$ and $\boldsymbol{S}_j$ are two normalized histograms, for the Saturation channel of objects i and j respectively

3. Object bounding box distances ($d_{\mathcal{A}}$ and $d_{\mathcal{B}}$): Compares the likelihood between the bounding boxes of two objects based on the area (number of pixels), width and height of each object.

The area ratio between 2 objects corresponds to:

$$d_{\mathbb{A}}(i) = \frac{\min(A_i, A_j)}{\max(A_i, A_j)} \tag{5.4}$$

where A is the area of an object, i corresponds to the current input object and j is an entry in database.

The bounding box size ratio $d_{\mathbb{B}}$ obtained from the width and height is defined by first computing the size ratios for each object, as:

$$\begin{aligned} R_i &= \left(\frac{\max(w_i, h_j)}{\min(w_i, h_j)}\right) \\ R_j &= \left(\frac{\max(w_j, h_j)}{\min(w_j, h_j)}\right) \end{aligned} \tag{5.5}$$

where w and h are the width and height of the minimum bounding box containing the pixels of the object i and j, respectively.

We compute then the minimum and maximum for each of these ratios in Eq. 5.5 as:

$$\begin{aligned} R_{min}(i, j) &= \min(R_i, R_j) \\ R_{max}(i, j) &= \max(R_i, R_j) \end{aligned} \tag{5.6}$$

The Ratio R between maximum and minimum sizes gives an indication about object shape which is independent of scale and rotation: no matter the position and orientation, if an object is long then its ratio R will be larger when compared to a compact (rather round) object.

Finally, using Eq. 5.6 we define the distance $d_{\mathbb{B}}$ as:

$$d_{\mathbb{B}}(i, j) = \frac{R_{min}(i, j)}{R_{max}(i, j)} \tag{5.7}$$

where i and j are the objects being compared. $d_{\mathbb{B}}$ gives a distance between the shapes of the objects using the ratio r, if the shapes are similar this distance will be small.

Each feature used in the calculation of these distances has different impact on the overall object comparison. For example, the color distribution (the Hue histogram in particular) has the greatest influence over the others in our recognition system. Therefore, each distance value is assigned a corresponding weight. The value of each weight was calculated by repetitive experiments with different objects.

Combining these 5 metrics, we define the likelihood between the actual object i and an object in the system database j as the weighted sum:

$$\mathcal{D}(i,j) = \sum_{n=1}^{N} \Big(\omega_n m_n(i,j) \Big) \tag{5.8}$$

where $m_n(i,j)$ corresponds to the n-th metric function (i.e., $m \in \begin{bmatrix} d_{\mathcal{M}} & d_{\mathcal{H}} & d_{\mathcal{S}} & d_{\mathcal{A}} & d_{\mathcal{B}} \end{bmatrix}$), ω_n defines the weight of the n-th metric and it is normalized so that $\sum_n^N \omega_n = 1$. The values for these weights were obtained experimentally, according to the observed importance of each metric in the recognition process.

As already shown in Eqs. 5.1, 5.2, 5.3, 5.4 and 5.7, each of the metric functions m_n in Eq. 5.8 were defined so that their return value satisfy:

$$m_n(i,j) = \begin{cases} = 1 & \text{for a perfect match} \\ < 1 & \text{otherwise} \end{cases} \tag{5.9}$$

We select from the database the k-th object whose likelihood $\mathcal{D}(i,k)$ is the maximum and not under some minimum threshold θ, this is:

$$\mathcal{O}_k = \arg \max_j \Big(\mathcal{D}(i,j) \geq \theta \Big) \tag{5.10}$$

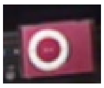
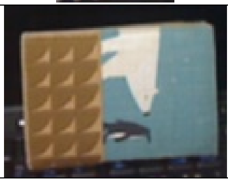
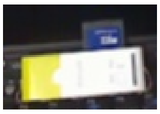
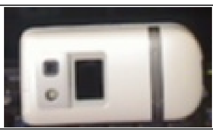
Once an object k is detected and recognized, the user interface would be set active.

5.3.1 Experimental Result

In this experiment we present the results for object detection and recognition with an existing database. We selected a four physical objects: a MP3 player, a mirror, a wallet, an USB memory stick, and a cell phone, as presented in Table5.1.

The width and height values are obtained from the minimum bounding rectangle

Table 5.1: Objects used for the experiment

Object	ID	Description	width	height	area
	1	Red MP3 player	52px	42px	1960px
	2	Pink pocket mirror	63px	63px	3349px
	3	Colorful wallet	119px	92px	9757px
	4	White USB stick	82px	56px	3424px
	5	White cell phone	111px	65px	6707px

enclosing the object's pixels, the area corresponds to the number of pixels inside the object's contour.

As evaluation of our object recognition method, we registered objects 1, 2, 3 and 4 into the database. Then we place the known object 2 in the capturing area, with a different posture (rotation and position) than when first registered. The recognition results are presented in Table5.2.

As expected, the likelihood from object 2 with itself is the maximum and is quite over the defined recognition threshold (currently defined as 0.85).

Next, we evaluate recognition of new (unknown) objects. From the objects in Table5.1, we registered objects 1, 2, 3 and 4 into the database and then present the unregistered object 5 for recognition. The results of comparing object 5 with the database are presented in Table5.3.

The closest object in this case, in terms of $\mathcal{D}$, is object 4, however its likelihood value 0.6689 is below the defined recognition threshold (0.85), therefore we consider the given target object as a new object and add it to the database.

Table 5.2: Recognizing a known object

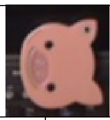
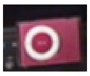
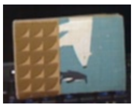
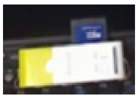
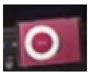
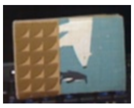
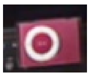
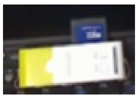
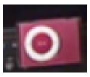
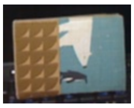
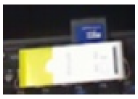
Target						
Source	$d_{\mathcal{M}}$	$d_{\mathcal{H}}$	$d_{\mathcal{S}}$	$d_{\mathcal{A}}$	$d_{\mathcal{B}}$	$\mathcal{D}$
	0.700483	0.588213	0.577307	0.831173	0.587706	0.643659
	0.923989	0.928225	0.946437	0.983936	0.986215	**0.946345**
	0.612787	0.355782	0.541775	0.760311	0.341806	0.503185
	0.464661	0.516006	0.481702	0.677266	0.974007	0.565682

Table 5.3: Comparison with an unknown object

Target						
Source	$d_{\mathcal{M}}$	$d_{\mathcal{H}}$	$d_{\mathcal{S}}$	$d_{\mathcal{A}}$	$d_{\mathcal{B}}$	$\mathcal{D}$
	0.376552	0.480759	0.475147	0.758587	0.287209	0.501082
	0.421656	0.501876	0.401348	0.606724	0.504694	0.487428
	0.543371	0.580591	0.459681	0.786310	0.689118	0.593189
	0.443755	0.661524	0.677051	0.911580	0.490842	0.668940

Finally, we present the complete confusion matrix in Table 5.4, here we present the likelihood $\mathcal{D}$ between all the objects.

Table 5.4: Confusion Matrix

target / source					
	**0.933398**	0.606716	0.538874	0.585548	0.501082
	0.604394	**0.906684**	0.495156	0.597515	0.487428
	0.570735	0.520482	**0.900626**	0.601022	0.593189
	0.535870	0.590346	0.548326	**0.857192**	0.668940
	0.521591	0.486692	0.586875	0.661996	**0.916163**

As expected, recognition of a known object (values on the diagonal) present the highest values, all over the defined threshold. Comparison of objects with similar properties will also have relatively high likelihood values, for example objects with similar histograms: 1 with 2 (likelihood of about 0.60), and 4 with 5 (likelihood of about 0.66); objects with similar size ratio: 3 with 5 (likelihood of about 0.58), etc.

5.4 Multiple Object Recognition

In our system we can detect and recognize multiple physical objects on the scene (capturing area) by analyzing all the contours extracted after background subtraction. We provide an option to the user so that he/she can stop or reactivate new objects recognition, which gives him/her the possibility to specify at once the subset of physical objects to work with and, if it turns necessary, to add new objects by reactivation.

Results for two cases of multiple object recognition are presented in Figure 5.2. In Figure 5.2(a) and 5.2(d) we show the physical objects used, previously learned by the system in the given order (top to bottom) they were placed and learned. In Figure 5.2(b) and 5.2(e) we present the actual frames obtained from the camera with all the objects placed simultaneously. In Figure 5.2(c) and 5.2(f) we present the recognition results, the colored frame around the object image and the number on top correspond to the recognized object on the database.

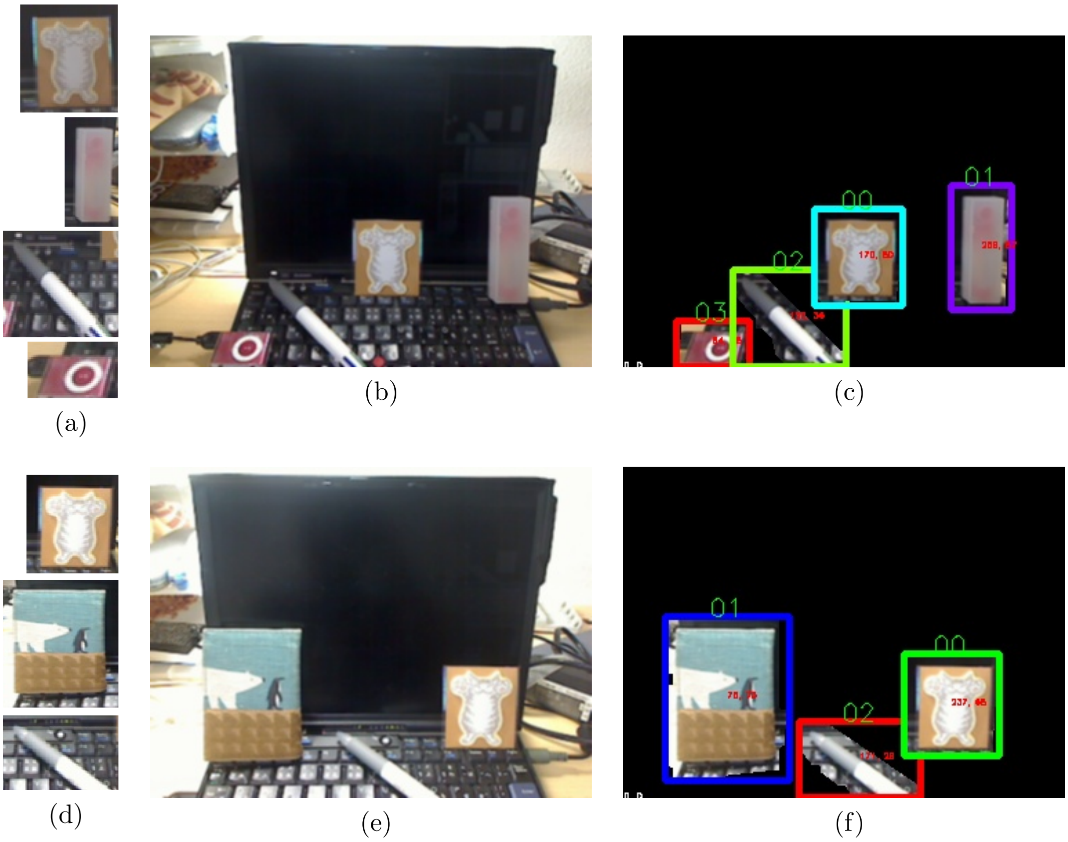


Figure 5.2: Results for multiple object recognition: (a) and (d) correspond to the known physical objects, (b) and (e) actual camera frames with all the objects, and (c) and (f) recognition results for all the objects.

5.5 Separation for Multiple Objects

As part of the object detection step, the contour images from the objects are dilated in order to close holes (gaps) that appear due to noise, poor illumination and other conditions. Thus, the object contour not only encloses the pixels from the object but also a small part of the background. Additionally, when an object is placed on the scene (for example on top of the computer keyboard as in Figs. 5.2(c) and 5.2(f)) it causes some reflections and illumination changes on the surface (shadows), therefore a small fraction of the background pixels around the physical object form part of the object's pixels.

Now, when two or more objects are placed on the scene, it is possible that their individual contours get merged due to their respective dilations, reflections and shadows, particularly when they are physically separated by some small distance and of course when they are overlapped. An example is presented in Figure 5.3 where to different objects (included as insets) are very close (overlapped) and the individual objects are not recognized. Moreover, if the system is running in the mode of new object recognition, then this merged object is treated incorrectly as a new object.

Figure 5.3: Two different objects merged together affecting recognition.

To solve this problem, we include a step of object separation which in turn is used to implement object tracking. If the individual physical objects are known beforehand (i.e., the training function described above), we already have the images for each of the objects our database, as well as other information such as center, bounding box, metrics for object comparison, etc.

In an object merging situation, the individual contours are connected forming a larger contour. To detect merging, we compare each newly-extracted-contour C_i with each of the active objects in the database (i.e., those recognized in the previous frame) O_j. If C_i area is larger than O_j area and C_i bounding box overlaps O_j bounding box, we flag O_j as merged.

This merging detection produces a list of all the objects merged with this newly-extracted-contour C_i, regardless of the visibility of the object (i.e., if the object O_k is partially or totally occluded by other objects). Of course, this merging detection step is effective if the system was trained for some objects and is not in the mode of recognizing new objects.

For each of the known objects O_k merged in C_i, we now analyze its visibility. The object O_k image from the previous frame is compared with the image of the contour C_i using the Continuously Adaptive Mean Shift method (CAMSHIFT) by Bradski [31]. The CAMSHIFT method, based on the Mean Shift algorithm, tries to locate the position of an object by iteratively moving (shifting) and resizing a search window over the image so that this window's center is closest to the object mean. We use the OpenCV implementation of CAMSHIFT, and the inputs of this method are a backprojection image and the initial position and size of the search window. The backprojection corresponds to the re-application of a histogram on an image, so that the histogram bins function as look-up table for the pixel colors.

In our implementation, once we know the object O_k is merged in the contour, we use O_k HSV histogram feature to compute the back projection on C_i HSV image. As for the initial search window, we use O_k bounding box.

By computing the backprojection of the object's histogram on the contour's image, we remove several of the pixels that do not correspond to the object, therefore we can detect whether the object is visible or not.

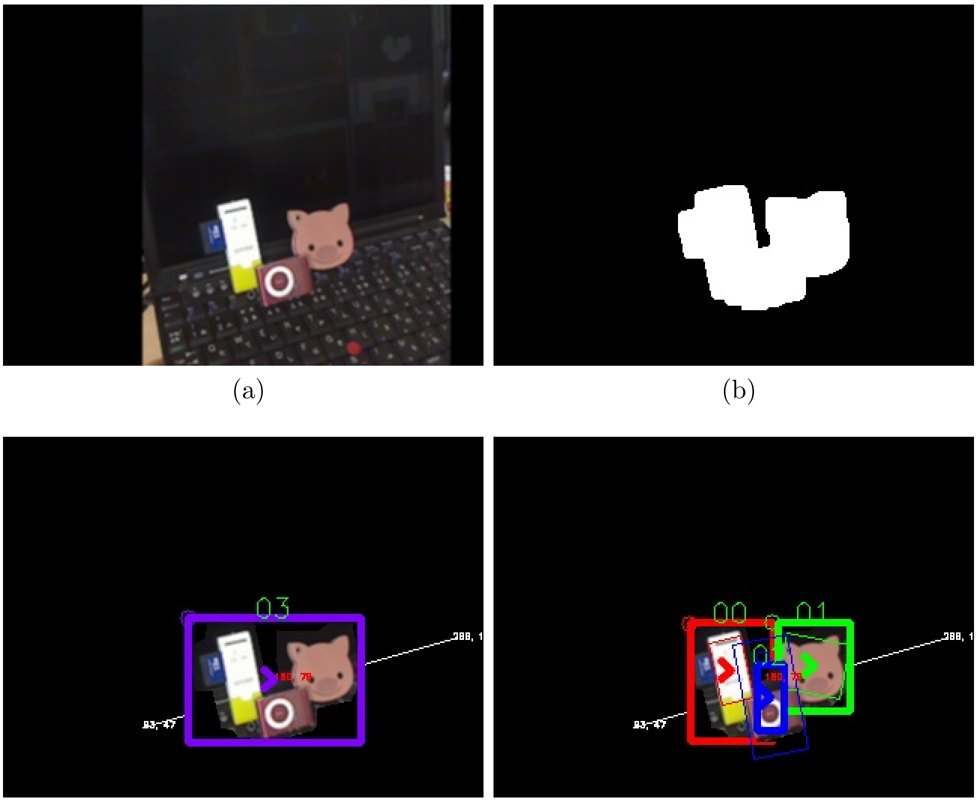
As part of the CAMSHIFT results, we obtain the window location and the area of the object's connected components. If the size of this window and the area are above some minimum acceptance thresholds (defined as a fraction of the object bounding box) then we mark the object as recognized.

By iterating over all the objects merged in C_i using our method, we achieve separation.

Results of multiple object separation are presented in Figure 5.4.

5.6 Object Tracking

The method for separation of merged objects described above, is used as the base for object tracking since we keep updating the object's position. If a user wishes to move an object currently placed on the scene, he/she grabs the object with his/her

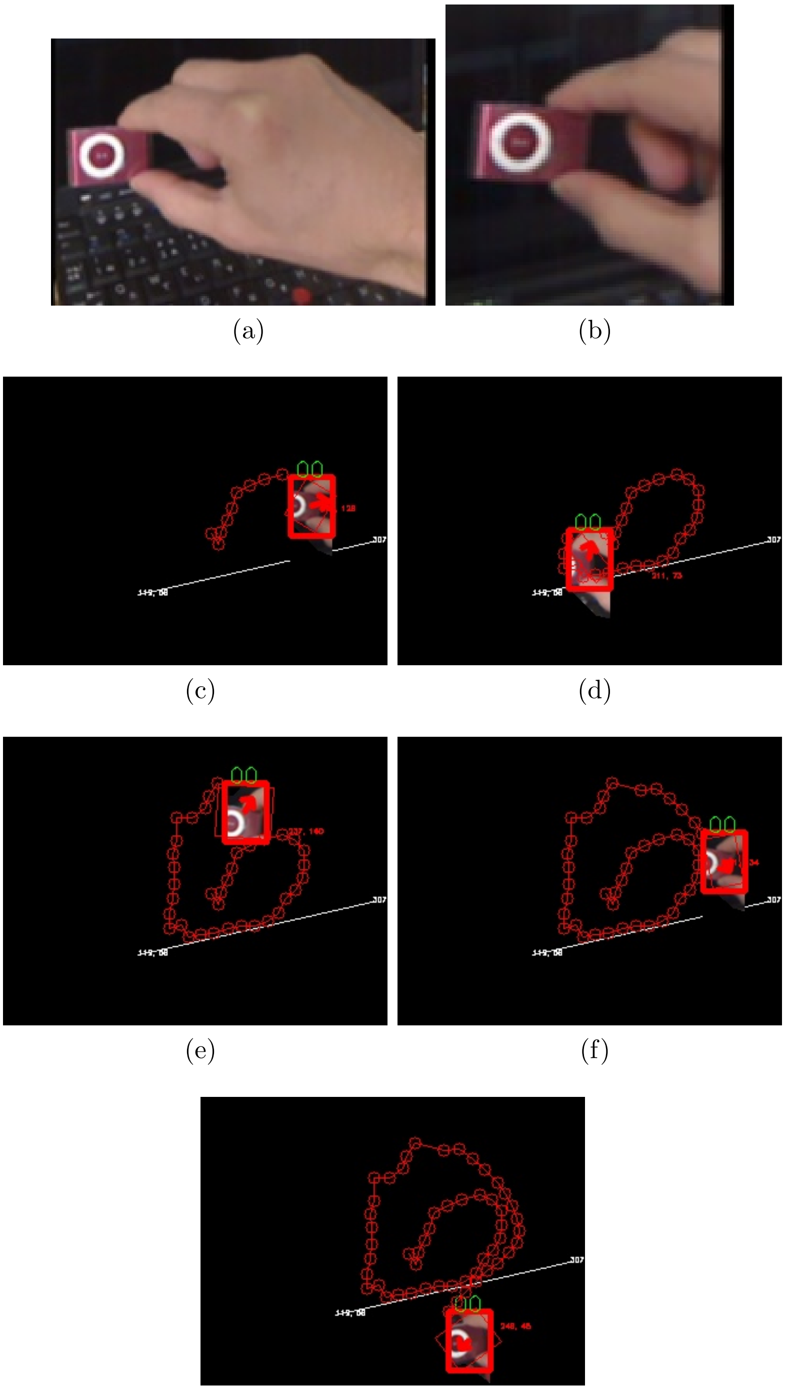


(a) (b) (c) (d)

Figure 5.4: Multiple objects separation results: (a) raw image, (b) merged contour, (c) failure case, and (d) successful separation

fingers and move it around. However in our system placing the hand or fingers in the capturing area means placing a new object(s), therefore if the hand is grasping a known object this creates a merging situation.

Results of tracking are presented in Figure 5.5.



(a) (b)

(c) (d)

(e) (f)

(g)

Figure 5.5: Tracking results: (a) raw image, (b) detail of object with hand, (c) to (g) tracking the object moving in a spiral shape.

Chapter 6

Evaluation

In this chapter we present the evaluation of our system Tangible Linker.

Figure 6.1: Objects used to test the accuracy of Recognition module

Before we carried out the user evaluation, we tested our Recognition module for accuracy of detection. In this test, we collected 20 objects varied in color and shape and registered them into the object database for trials. We present the set of objects used for the evaluation in Figure 6.1. Each trial consists in placing an object in the sensing area in certain position (left, center and right side of the sensing area) and angle (rotation of 90 grades to left and right) and see whether the object was correctly identified.

First we tried to detect each object individually. We did eight trials per object for single object detection. Then we tested multiple object detection where two and three objects were in the sensing area simultaneously and each one must be detected individually. For two objects detection we did two trials per object and for three objects detection, three trials per object. In Table 6.1 we present detection rates for this general evaluation.

Table 6.1: General evaluation of object recognition

Test	True Positive	False Positive	False Negative	True Negative
Single object	87.5%	0.42%	12.5%	99.4%
Two objects	92.5%	0.28%	7.25%	99.72%
Three objects	83.3%	0.29%	16.7%	99.71%

From Table 6.1 it can be observed that the recognition modules worked good for all the three scenarios. Even some objects are very similar in color and shape, it did not cause significant problem to the detection. We can observe that the False Positive Rate is very low, which means that when an object could not be detected correctly, the system simply treated it as a new object but did not confuse with others in the database. Another observation is that the position where the object was placed affected significantly the result.

Recognition issues can be fixed by including a correction mechanism, where user can manually select the correct ID for the object. This way we can construct a mixture model from more than one image from an object and use supervised learning to improve existing features.

6.1 User Evaluation

The purpose of our experiment is to evaluate the usage of visual perception when retrieving information, study user's classification habit and find out end user's opinion about using any physical object to access digital information using our system.

6.1.1 Experiment Setup

A total of 14 participants joined the experiment, ages ranged from 21 to 37 years old. 11 participants are expert computer users and the remaining 3 have basic operation skills. First they were presented with a pool of 35 files of various types (documents, images files, URL, etc.) and topics (math, computer science, travel, restaurants, services, etc.). Using Tangible Linker, they had to classify the information under any criteria most suitable to them to be recovered afterward.

After they finished the classification, we asked the users to retrieve 12 particular contents using Tangible Linker through the physical objects they used. At the end, they had to fill a questionnaire about their opinion of the experience and the system. Ten minutes training was given before the experiment. Most of the participants showed excitement about using any available physical object to interact with the computer. Each participant decided the subset of physical objects to use in the experiment and the respective mapping of contents.

6.1.2 Results

Regarding the first task (classification), for the set of 35 digital contents, our participants used in average 5.6 objects to classify them. Table 6.2 shows the number of objects used by the participants. A total of 79 different objects were used in the whole experiment. The average digital contents number stored in each physical object is 6.3, with a maximum of 13 contents in one object. Although the total number of digital contents was relatively small, users preferred a small number of contents per object to efficiently complete the task.

Table 6.2: Number of objects used for experiment

N. of Users	N. of Objects
1	3
1	4
1	5
10	6
1	7
Total objs.	79
Usage avg.	5.6

Regarding the second task (retrieval), each of the 14 participants was requested to retrieve 12 different digital contents stored during the first task. We observed that users were very efficient during contents recovery. Even though some participants showed concern about forgetting where they have stored some specific information, during the retrieval phase they tried to associate the requested information according to their own visual perception of the object, and for most of the times it was a correct retrieval (a hit). As presented in Figure 6.2, 144 out of 168 attempts (86%) they were able to recover the requested information at once. Only once a participant had to try 5 objects and once for 6 objects to retrieve the requested information.

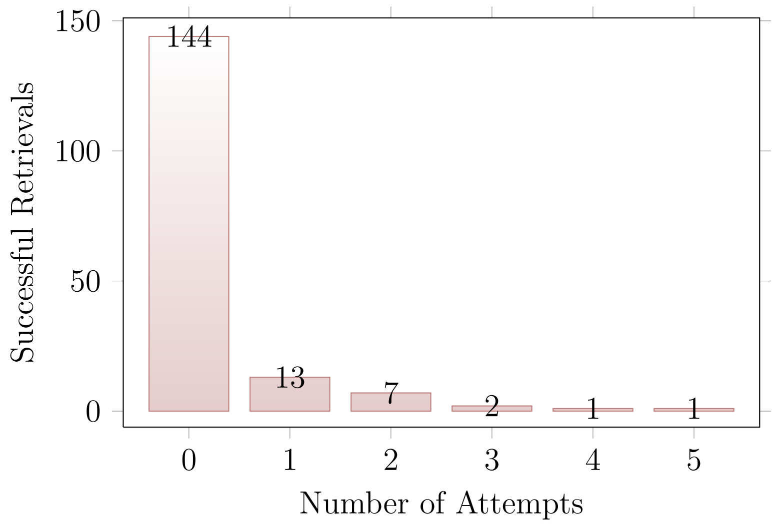


Figure 6.2: Number of attempts to successful retrieve information.

Figure 6.3 shows the general opinion about the system. We received positive feedback with the exception of two cases (mark 5 and 2). The lowest mark belonged to a participant who considered the system should run over the Network, enabling to retrieve information stored in an object from any computer. Some participants argued that the experiment task did not reflect their ordinary working scenario and found the tool not suitable for the task. However they expressed interest in using it for other purposes, for example to classify their files by job, others suggested using it to manage photo albums and music lists. Finally, 11 out of 14 participants showed interest in using Tangible Linker for daily activities.

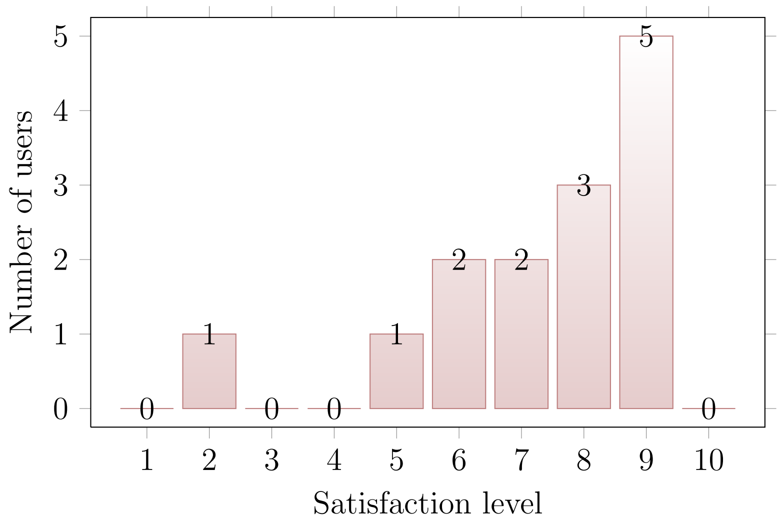


Figure 6.3: General Opinion of Tangible Linker.

6.2 Analysis

Regarding classification habits, 6 users selected file extension for classification, 7 by topic and one user did not use any pattern, while 3 of them associated the same content to more than one object as shown in Table 6.3.

Table 6.3: Classification method preference

Type	N. of Users
By file extension	6
By topic	7
Mixed	1
Crossover (same content in more than one object)	3

Users organized contents following some logical (not random) classification method. Given the classification habits, we believe it is possible to design a system that, given some physical object, it is able to make predictions on the type of digital contents that should be stored on it and make recommendations to the user. For that, semantic information about the contents in an object (type, name, title, etc.) can be collected to create a profile for the object.

The object recognition module worked well for most of the cases. However, due to the 3D nature of physical objects, when the user expose the object showing a face very different from that when the object was registered (e.g., a Rubik's cube), the recognition system may fail. Also in cases of objects which shape varies between usages (e.g., one participant chose a keychain with some keys). Besides the improvement mentioned before, we must consider also a better way to detect object with dynamic shape.

Chapter 7

Conclusions and Future Work

This thesis described a new tangible interaction method based on untagged physical objects for personal computer system. An alternative solution to WIMP is proposed for retrieval of information without involving conventional searching procedure in graphical user interface. In our system, objects require no modifications and no previous setup so any object around the user space can be integrated with the digital desktop when needed. We implemented the system Tangible Linker to show how surrounding objects can be augmented to represent digital information also a control point over an active window in the digital desktop environment. By using our system people are capable to relate easily information with physical objects by visual perception. The experiment showed that users are eager to explore new interactive environments, for example our tangible interaction method, in their daily activities, using ordinary objects to ease their work. We believe this is another step towards a true ubiquitous environment where seamless interaction with any physical object is possible.

As future works, we will extend our system to be available in a networked environment where users can share information simply by exchanging physical objects as well as access to shared data, for example in a cloud computing environment. Although our object detection and recognition system is simple and works well for several cases, improvement is required in order to make our recognition system tolerant of variation on physical objects aspect (refletive surface, changeabe shape and size, etc.). We consider supervised machine learning for a more realisitc object model.

Appendix A

Tangible Linker Operation Procedure

In this section we describe the procedure of operation in Linker module on incoming message from Recognition module regarding to the state of the object. There are three possible states: object newly detected, updated or removed.

The Figure A.1 describes the procedure when object state is newly detected. This procedure is executed every time an object is placed in the sensing area. If the object has never been used in the system, Linker inserts a new entry in the database creating a XML file with the object ID only. For visual feedback, a notification of empty object is shown to user.

In case the object was recognized as an existing entry, Linker will load the XML file of the object and check for valid items in current context (whether the item exists in current file system) to be displayed. If there is another object already in the sensing area, Linker will check for the proximity of the recently detected object with the other one and search for common keywords in both objects profile. If the list of common keywords is not empty, those items described by these common keywords are included in an additional reference list to be display between these objects.

The Figure A.2 describes the procedure when object state is updated. An object updated means that it was detected recently but it disappeared for a very short time and before Recognition module considered it removed, it was detected again. Here Linker will check the location information of the object. If an object has not been

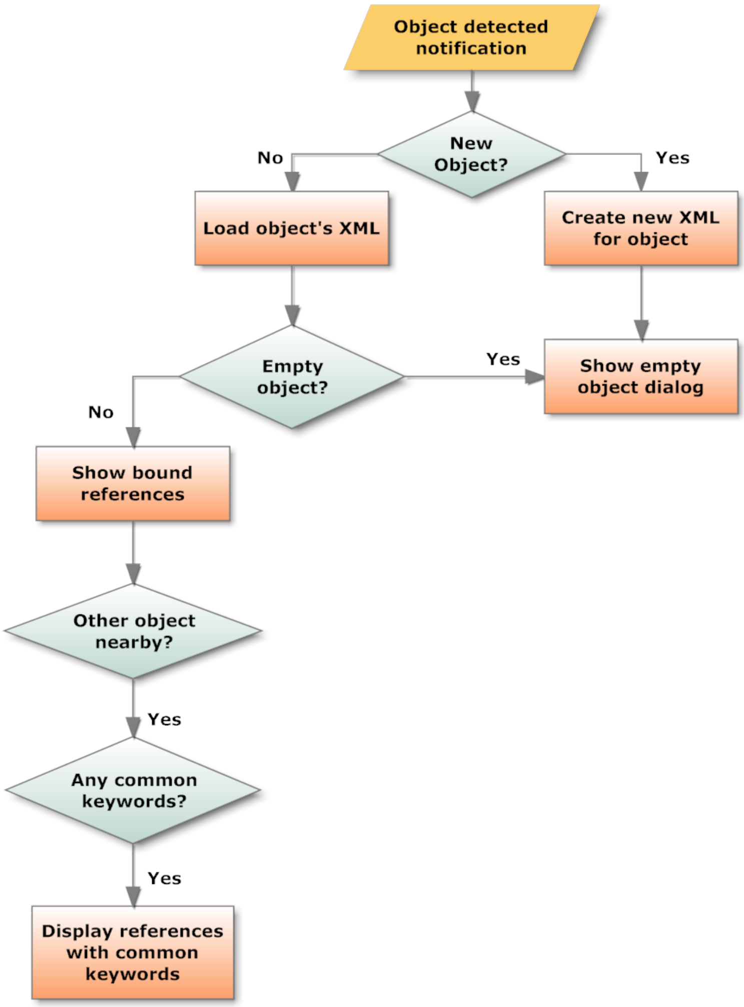


Figure A.1: Flow chart on object detected message

moved, then the new location is the same with the previous one, and it is interpreted as a mode switch operation. Otherwise, Linker will get the attached window information and obtain the command from the difference of the current location and the previous one. In case that an object does not have attached window, Linker proceeds to search for nearby object and filter out related items if any (as described previously).

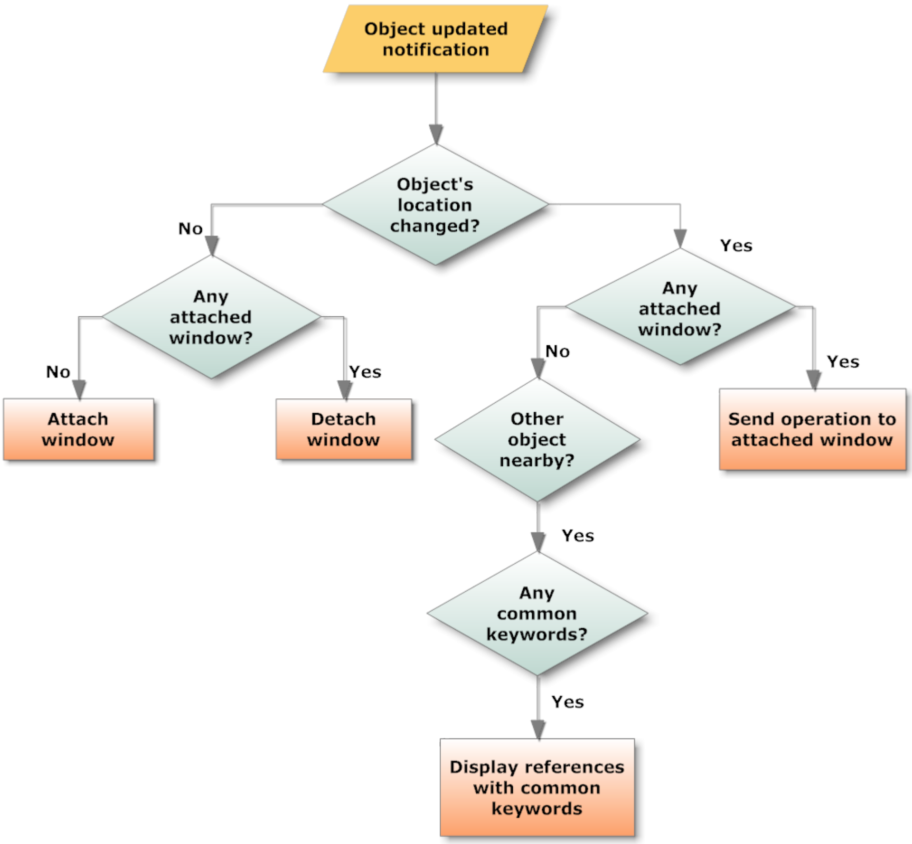


Figure A.2: Flow chart on object updated message

Acknowledgements

I would like to express my profound gratitude to my advisor Prof. Jiro Tanaka for his encouragement, sound advice and good teaching, and for giving me the chance to join IPLab. Also my gratitude to professors Prof. Misue and Prof. Shizuki for their invaluable insights through several presentations and seminars.

I must thank most especially to Dr. Shin Takahashi whose expertise, understanding and patience, added considerably to my graduate experience. My deepest gratitude to Dr. Takahashi for his supervision, advice, and guidance from the very early stage of this research as well as giving me extraordinary experiences throughout the work. I would have been lost without him.

I also wish to thank warmly my colleagues in IPLab who, through our philosophical debates, exchanges of knowledge, skills, and venting of frustration during my graduate program, which helped enrich the experience.

I would also like to thank my family for the support they provided me through my entire life and in particular, I must acknowledge my husband and best friend, Alexander, without whose love, encouragement and editing assistance, I would not have finished this thesis.

In conclusion, I express my gratitude to University of Tsukuba who had provided me the opportunity to complete my graduate studies in this prestigious institution.

Bibliography

[1] M. Lansdale, "The psychology of personal information management," *Applied Ergonomics*, vol. 19, no. 1, pp. 55–66, 1988.

[2] A. Van Dam, "Post-wimp user interfaces," *Communications of the ACM*, vol. 40, pp. 63–67, February 1997.

[3] H. Ishii and B. Ullmer, "Tangible bits: towards seamless interfaces between people, bits and atoms," in *SIGCHI conference on Human factors in computing systems*, (Atlanta, United States), pp. 234–241, March 1997.

[4] H. Ishii, "Tangible bits: beyond pixels," in *2nd International Conference on Tangible and Embedded Interaction(TEI 2008)*, (Bonn, Germany), pp. XV–XXV, February 2008.

[5] L. E. Holmquist, J. Redström, and P. Ljungstrand, "Token-based access to digital information," in *In Proceeding of the 1st International Symposium on Handheld and Ubiquitous Computing*, (Karlsruhe, Germany), pp. 234–245, September 1999.

[6] H. Ailisto, J. Plomp, L. Pohjanheimo, and E. Strömmer, "A physical selection paradigm for ubiquitous computing," *1st European symposium on ambient intelligence (EUSAI 2003), Lecture Notes in Computer Science*, vol. 2875, pp. 372–383, 2003.

[7] I. Siio and Y. Mima, "Iconstickers: converting computer icons into real paper icons," in *8th International Conference on Human-Computer Interaction: Ergonomics and User Interfaces*, (Munich, Germany), pp. 271–275, August 1999.

[8] P. Ljungstrand, J. Redström, and L. E. Holmquist, "Webstickers: using physical tokens to access, manage and share bookmarks to the web," in *Designing augmented reality environments (DARE 2000)*, (Elsinore, Denmark), pp. 23–31, April 2000.

[9] J. Rekimoto and Y. Ayatsuka, "Cybercode: Designing augmented reality environments with visual tags," in *Designing augmented reality environments (DARE 2000)*, (Elsinore, Denmark), pp. 1–10, April 2000.

[10] S. Iwabuchi, B. Shizuki, K. Misue, and J. Tanaka, "Natural storage in human body," in *9th International Conference on Knowledge-Based Intelligent Information and Engineering Systems*, vol. 4, (Melborune, Australia), pp. 430–436, September 2005.

[11] P. Wellner, "The digitaldesk calculator: Tangible manipulation on a desk top display," in *4th annual ACM symposium on User interface software and technology (UIST 1991)*, (South California, United States), pp. 27–33, 1991.

[12] T. Nishi, Y. Sato, and H. Koike, "Interactive object registration and recognition for augmented desk interface," in *IFIP conference on human-computer interface (Interact 2001)*, (Tokyo, Japan), pp. 240–246, July 2001.

[13] J. Patten, H. Ishii, J. Hines, and G. Pangaro, "Sensetable: a wireless object tracking platform for tangible user interfaces," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '01, (New York, NY, USA), pp. 253–260, ACM, 2001.

[14] P. Mistry, T. Kuroki, and C. Chang, "Tapuma: tangible public map for information acquirement through the things we carry," in *Proceedings of the 1st international conference on Ambient media and systems*, Ambi-Sys '08, (ICST, Brussels, Belgium, Belgium), pp. 12:1–12:5, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.

[15] P. Valkkynen, T. Tuomisto, and I. Korhonen, "Suggestion for visualizing physical hyperlinks," in *Pervasive Mobile Interaction Devices (PERMID 2006)*, (Dublin, Ireland), pp. 245–254, May 2006.

[16] P. Valkkynen and T. Tuomisto, "Physical browsing research," in *Pervasive Mobile Interaction Devices (PERMID 2005)*, (Munich, Germany), pp. 35–38, May 2005.

[17] A. J. Bernheim Bush, T. Combs Turner, M. A. Smith, and N. Gupta, "Scanning objects in the wild: assessing an object triggers information system," September 2005.

[18] R. Want, K. Fishkin, A. Gujar, and B. L. Harrison, "Bridging physical and virtual worlds with electronic tags," in *In Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*, (Pennsylvania, United States), pp. 370–377, 1999.

[19] S. Pradhan, C. Brignone, J.-H. Cui, A. McReynolds, and M. T. Smith, "Websigns: Hyperlinking physical locations to the web," *Computer*, vol. 34, pp. 42–48, August 2001.

[20] H. Ailisto, L. Pohjanheimo, P. V. lkkynen Esko Strömmer, T. Tuomisto, and I. Korhonen, "Bridging the physical and virtual worlds by local connectivity-based physical selection," *Personal and Ubiquitous Computing*, vol. 10, pp. 333–344, September 2006.

[21] Willow Garage, "Opencv." http://www.willowgarage.com/pages/software/opencv, cited december 2010.

[22] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*. Cambridge, MA: O'Reilly, 2008.

[23] H. Y. Wong, S. Takahashi, and J. Tanaka, "Tangible linker: organizing and accesing digital contents using untagged tangible objects," in *9th International Conference Asia Pacific Computer Human Interaction (APCHI)*, (Bali, Indonesia), pp. 693–698, August 2010.

[24] Yahoo! Developer Network, "Yahoo! query language." http://developer.yahoo.com/yql/, cited June 2010.

[25] T. Jensen, "Keyword extraction in c# with word co-occurrence algorithm." http://www.tsjensen.com/blog/2010/03/14/Keyword+Extraction+In+C+With+Word+Cooccurrence+Algorithm.aspx, cited June 2010.

[26] Y. Matsuo and M. Ishizuka, "Keyword extraction from a single document using word co-occurrence statistical information," *International Journal on Artficial Intelligence Tools*, vol. 13, pp. 157–170, 2004.

[27] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99)*, vol. 2, p. 2246, 1999.

[28] S.-C. S. Cheung and C. Kamath, "Robust techniques for background subtraction in urban traffic video," in *Proceedings of the SPIE Electronic Imaging: Video Communications and Image Processing*, vol. 5308, (California, United States), pp. 881–892, 2004.

[29] P. C. Mahalanobis, "On the generalized distance in statistics," in *Proceedings of the National Institute of Science, Calcutta*, vol. 12, pp. 49–55, 1936.

[30] A. Bhattacharyya, "On a measure of divergence between two statistical populations defined by probability distributions," *Bulletin of the Calcutta Mathematical Society*, vol. 35, pp. 99–109, 1943.

[31] G. R. Bradski, "Computer Vision Face Tracking For Use in a Perceptual User Interface," *Intel Technology Journal*, vol. 1, no. Q2, pp. 1–15, 1998.