

WEBCAM-BASED MARKER TRACKING APPLICATION

LIM DE JIANG

This thesis submitted in fulfillment of the requirements
for the award of the degree of the Bachelor of Computer Science
(Network Engineering)

Faculty of Systems Computer & Software Engineering
Universiti Malaysia Pahang

MAY 2011

PERPUSTAKAAN UNIVERSITI MALAYSIA PAHANG	
No. Perolehan 068653	No. Panggitan TA 1656 L56 2011 13 Bc.
Tarikh 30 NOV 2012	

ABSTRACT

Touchless technology is still considered a new field in information technology due to its effectiveness in real world application. This technology started in development in the late 80s. Although this technology has been around for about 20 years, the application of this technology is not widely. This is primarily because it is only suitable for interaction between human and computer for entertainment purposes. The example we can see is the implementation of this technology is in Xbox 360 or Microsoft Kinect, which is mainly targeted in entertainment. In order to implement this technology besides for entertainment, Webcam-based Marker Tracking Application has been developed to show its usage in writing recognition. It offers an interactive and fun way of handwriting recognition by using marker tracking. In addition, the low cost hardware and software requirement of this technology makes it can used by anyone. Object-oriented analysis and design method has been used in the development of this project. The way this method decompose each function into object makes it easy for problem solving. This is because we can focus on the solution on one object before continues to the next. Finally, I deeply believes Touchless can be implemented in other field besides than entertainment, probably it can be implemented in education for children learning purposes.

Keywords: webcam, marker tracking, human computer interaction, handwriting recognition

ABSTRAK

Teknologi Touchless masih dipertimbangkan sebagai satu bidang yang baru dalam teknologi maklumat disebabkan keberkesanannya dalam penggunaan dunia nyata. Teknologi ini mula dibangunkan pada tahun lewat 1980-an. Walaupun ianya sudah wujud lebih kurang dalam 20 tahun, penggunaan teknologi tidak luas. Hal ini disebabkan ianya hanya sesuai untuk interaksi antara manusia dengan computer dalam bidang hiburan. Contoh implementasi teknologi ini yang kita boleh nampak adalah Xbox 360 atau Microsoft Kinect, di mana sasaran utamanya dalam hiburan. Webcam-based Marker Tracking Application telah dibangunkan untuk menunjukkan kegunannya dalam pengecaman tulisan tangan selain daripada implementasi dalam hiburan. Ianya menawarkan kaedah yang interaktif dan senorok dalam pengecaman tulisan tangan dengan menggunakan pengesanan penanda. Tambahan pula, kos keperluan perkakasan dan perisian yang rendah membolehkan technology ini boleh digunakan oleh semua. Kaedah rekabentuk dan analisis berdasarkan object telah digunakan dalam pembangunan projek ini. Cara kaedah ini memecahkan setiap fungsi kepada objek membolehkan ianya senang untuk penyelesaian masalah. Hal ini disebabkan kita boleh focus kepada penyelesaian satu objek sebelum meneruskan kepada yang seterusnya. Akhirnya, saya sangat percaya yang Touchless boleh digunakan dalam bidang lain selain daripada hiburan. Mungkin ianya boleh digunakan dalam pendidikan untuk tujuan pembelajaran kanak-kanak.

Kata kunci: webcam, pengesanan penanda, interaksi manusia komputer, pengecaman tulisan tangan

TABLE OF CONTENT

CHAPTER	CONTENT	PAGE
	TITLE PAGE	i
	STUDENT'S DECLARATION	ii
	ACKNOWLEDGEMENTS	iii
	ABSTRACT	iv
	ABSTRAK	v
	TABLE OF CONTENT	vi-ix
	LIST OF TABLES	x
	LIST OF FIGURES	xi-xii
	LIST OF APPENDIXES	xiii
	 INTRODUCTION	 1-3
	1.1 Introduction	1-2
	1.2 Problem Statement	2
	1.3 Objectives	2
	1.4 Scope	3
	1.5 Thesis Organization	3

II	LITERATURE REVIEW	4-10
2.1	Introduction	4-5
2.2	Related Work on Webcam based Marker Tracking Application for Human-Computer Interaction	5-7
2.3	System Overview of Webcam based Marker Tracking Application for Human-Computer Interaction	8
2.4	Comparison between Existing System and Current System	8-10
2.5	Conclusion	10
III	METHODOLOGY	11-33
3.1	Introduction	11-12
3.1.1	Topics under methodology	12-14
3.2	Object-Oriented Analysis and Design	14
3.2.1	Object-Oriented Analysis	15
3.2.2	Object-Oriented Object	15-16
3.2.3	Unified Modeling Language	16-22
3.3	Fucntional Requirement	22
3.3.1	Webcam based Marker Tracking Application's Object	23
3.3.2	Webcam based Marker Tracking Application's Attribute	23

3.3.3	Webcam based Marker Tracking Application's Process/Behavior	24
3.3.4	Process/Flowchart of Webcam based Marker Tracking Application	25-27
3.3.5	Diagram Module	27-28
3.3.6	Application Flow	29-30
3.3.7	Iteration Diagram	30-31
3.3.8	Expected Result/Outcome Possibility	31-32
3.4	Hardware and Software Requirement	33
3.5	Conclusion	33
IV	IMPLEMENTATION	34-44
4.1	Introduction	34
4.2	Marker Tracking	34-36
4.2.1	Camera Detection	37
4.2.2	Marker Adding	37-40
4.3	Sketching using Marker	40-44
4.4	Conclusion	44
V	RESULT & DISCUSSION	45-49
5.1	Introduction	45
5.2	Result Analysis	45-47
5.3	Constraint	47
5.3.1	System Constraint	47-48

5.3.2	User Constraint	48
5.4	Advantages & Disadvantages	48
5.4.1	Advantages	48
5.4.2	Disadvantages	49
5.5	Future Research	49
5.6	Conclusion	49
VI	CONCLUSION	49-50
6.1	Introduction	49-50
	REFERENCES	51
	APPENDIXES	

LIST OF TABLES

TABLE	TITLE	PAGE
2.1	System Comparison Table	8-10
3.1	Object Tracking Application's Objects' Attribute	23
3.2	Webcam based Marker Tracking Application's Objects' Process/Behavior	24
3.3	Software Requirement	32
3.4	Hardware Requirement	33

LIST OF FIGURES

FIGURE	TITLE	PAGE
3.1	Business Model Integration	13
3.2	Class Diagram	16
3.3	Component Diagram	17
3.4	Composite Structure Diagram	17
3.5	Deployment Diagram	18
3.6	Object Diagram	18
3.7	Package Diagram	19
3.8	Activity Diagram	19
3.9	State Machine Diagram	20
3.10	Use Case Diagram	20
3.11	Communication Diagram	21
3.12	Interaction Overview Diagram	21
3.13	Sequence Diagram	22
3.14	Webcam based Marker Tracking Application Flow Chart	25
3.15	Module Diagram	28
3.16	Activity Diagram	29
3.17	Marker Sequence Diagram	30
3.18	Writing Recognition Diagram	31
3.19	Expected Result (Application Interface)	31
4.1	Touchlesslib Library	35
4.2	Add Reference	35

4.3	Reference Added	36
4.4	WebCamLib Library	36
4.5	Detect Camera Code	37
4.6	Select Marker	38
4.7	Freeze Frame Code	38
4.8	Mouse Location Code	38
4.9	Mouse Movement Code	39
4.10	Adding Marker Code	39
4.11	Draw Marker Code	39
4.12	Marker Added	40
4.13	Start Sketching	40
4.14	Sketching Code	41
4.15	Stop Sketching	41
4.16	Stop Sketching Code	42
4.17	Save Sketches	42
4.18	Sketches Bitmap	43
4.19	Sketches View	43
4.20	Save Skecthes Code	44
5.1	Marker Tracking	46
5.2	Sketching Using Marker	46
5.3	Display and Save Sketches	47

LIST OF APPENDIXES

APPENDIX	TITLE	PAGE
A	Gantt Chart	53-56
B	Questionnaire	57-60

CHAPTER 1

INTRODUCTION

1.1 Introduction

Over a few years ago, the Touchless is a new concept. It implements human-computer interaction concept, Natural User Interface (NUI) which user can interact with computer without necessarily to use a means of device as a medium such as mouse or keyboard. Nowadays, this concept has become a new technology. However it is still not widely spread among community today as it replaces computer input devices such as mouse and keyboard. For most of the organizations, this is not necessary as it is sufficient for the employee to use keyboard and mouse as computer input devices. They do not have to spend extra and waste cost to implement this new technology.

The idea is by using a camera to capture or monitor the movement of a gesture or structures as marks with a single color tone, different from others. Such monitoring can get the positions and the speed of the marks. For example, we can use our own finger with color tape as marks to move objects in computer such as a map, folder, or we can even draw or write using our fingers.

The ability of Touchless to allow more than one touch point (mark) or known as multi-touch offers more interactive power of the touchscreen. Besides that, this technology does not need the costly hardware and software to experience the multi-touch capabilities. A simple camera that is used to monitor the color marks set by users is sufficient to implement this technology.

Now, Touchless is free and open source to the world. The most common and well-known example of Touchless is Microsoft SDK for .NET Framework. The most recent of the implementation of this technology is X-Box Natal as touchless games.

1.2 Problem Statements

The problem statements of this project are as below:

- This technology is the future means of interaction between human and computer by NUI (Natural User Interface) computing, as claimed by Microsoft's chief research and strategy officer, Craig Mundie. Nowadays, human still bound to keyboard and mouse in order to interact with computers.
- The interaction method or device used between computer and human like PDA, touchscreen or any other device that are expensive compare to a single camera that is used by Touchless technology.
- Touchless technology is majorly implemented in entertainment purposes like Xbox 360, but not in other field such as learning purposes.

1.3 Objective

The objectives of this project are as below:

- To develop a prototype using marker tracking.
- To sketch or write using marker.
- To ensure sketches drawn by using marker can be saved.

1.4 Scope

The scopes of this project are as below:

- Any end user that stand or position themselves within the range of the camera and under excellent lighting condition for the camera to capture or monitor the marker.
- The variation color of the background of the user has to be as less as possible.
- The application that is created to implement Touchless is only able to recognize one alphabet at one time.

1.5 Thesis Organization

This thesis consists of six chapters. Chapter 1 – Introduction, will discuss on the introduction to the Touchless technology, problem statement, objective, scope of this project and thesis organization. Chapter 2 – Literature Review, will discuss about literature review regarding the title of this thesis. Through this chapter, review will be done about existing system and the method, tools, software and technique that are suitable to develop this project. In the other hand, Chapter 3 - Methodology will discuss about the method or the approach used in developing, designing and implementation of this project. Besides that, it will discuss about the software and hardware requirement. In Chapter 4 – Implementation, it will discuss about the documentation about all the processes or procedure involved in the development of this project. It will also show the design of user interface and the source code of the project. Chapter 5 – Result & Discussion will explain about the result or the outcome of this project. It also will discuss about the obstacles faced during the development and future upgrade of this project. The last chapter, Chapter 6 – Conclusion will discuss about the conclusion of the whole project, the methodology and implementation used, possible upgrading and drawback of this project.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Computer technologies are rapidly-evolved technologies. Every day, we can hear, see and read about new computer technologies have been developed and put into practical use. Some of the technologies like devices and applications that do not have traditional mouse and keyboard interface, such as TabletPCs, media-center PCs, kiosks [7], hand-held computers, home appliances, video-games, and wall-sized displays. In these applications and devices, mouse and keyboard may seem to be clumsy and may require more effort than they should be. As an alternate channel of interaction, besides using voice to communicate with computer, hand gestures also being used, or known as Touchless. However, there are some drawbacks. For example, using intrusive hand tracking devices such as data gloves can be cumbersome and expensive. In order to overcome this problem, computer vision is used to capture hand gesture/motion.

Touchless is not a new technology. It is highly expressive and simple to manipulate. It first started by Mike Wassermans as college project at Columbia University. For now, Touchless is still considered as open research problem as it still

has many restrictions and limitations like the color response of captured images is heavily affected due to the environmental lights. The slight changes could make significance changes of the color captured by camera and thus making large error in tracking algorithm. Besides this, the tracking algorithm consumes a lot of computer computing resources as it needs to process a lot of captured images. These two limitations are mentioned in [1]. Touchless is not widely used due to these reasons. It can only be used in applications that can be tolerated with these inaccuracies [1]. However, due to its high potential of becoming next generation navigational computer input devices [7], a great deal of research efforts have been devoted for developing new algorithm and exploring new techniques in order to overcome the limitations of Touchless technology.

2.2 Related Work on Webcam based Marker Tracking Application for Human-Computer Interaction

From the demonstration videos on [2], there are several applications that able to applied object tracking quite well and do not need high image accuracies and able to tolerate with slight environmental lights changes. For example, by using single marker or two markers, we can write, sketch or draw. Another application is navigation of a map or an object by using single marker and by using two markers. By using markers, an object or a map (as demonstrated in [2]) can be panned & tilted, zooming in and out around the application canvas. However, these applications are restricted by using markers and inside the applications windows.

FlowMouse, demonstrated by Andrew [12], uses optical flow technique for object tracking to avoid difficulties such as lighting conditions, skin color and hand shape model development, and segmentation of separating foreground from background. This application contains a switch that can be used to enable or disable the application. The switch allows the application to process only intentional gesture by users and allows “clutching” action for enabling and disabling the application. Even tough the switch avoids problem such as confusion between the motions for positioning and clicking action, it does not allow practical use. The algorithm of

FlowMouse operates by acquisition of grayscale image from the camera. Then the optical flow is computed by using simple Block Matching Algorithm. At each point (x,y) on a regular grid in the image, the integer vector quantity (dx,dy) is determined such that the image patch centered on (x, y) at time $t - 1$ most closely matches the image patch centered on $(x + dx, y + dy)$ at time t . In this calculation, image patches are compared by computing the sum of pixelwise absolute differences (SAD). Low SAD values means there is close match between the two images.

Gesture and motion tracking also can be seen applied at kiosk system. Corey [6] discusses about a 3D interactive kiosk system that consists of gesture tracking, voice recognition, an interaction engine and a graphics module. However, I only focus on gesture tracking. Corey [6] uses skin model for gesture tracking. They develop the skin model based on users' face skin color tone. This is because users have wide range of skin color tone and the difficulties of accurately capturing skin color due to environmental conditions. The system's skin color model uses Hue Saturation based color instead of RGB color space to detect users' hand motion. This is because hue defines dominant color while saturation measures the saturation of hue to white light, as mention by Corey [6]. The system does not contain any skin color model. It will only initialize and maintain the skin color model when the system starts. Then it will find users' face (face detection) and extract skin color information. The system maximizes the uses of face detection to determine whether the users is facing the system or not, as there will be a lot of people walking past the kiosk. For the same reason, a stereo vision device is used to calculate the depth of map of hand areas in every frame of image and remove false hand areas. Then, CamShift algorithm (Continuously Adaptive Mean Shift, face tracking algorithm) is used to track the face areas between the current frame and the next. This algorithm combines the basic of Mean Shift algorithm with an adaptive region-sizing step. This system allows users to move object from left to right, upward to downward and vice-versa, rotating object and scaling object by zooming in and out.

Wilson [11] demonstrates the uses of multiple hypotheses tracking approach for gesture tracking. This approach monitors only the object that is closest to the camera, while ignoring the others, which is usually users' hand when they are facing the camera. In the other hand, the algorithm that they used for motion tracking is

simple image comparison function, the sum of absolute differences (SAD) over square patches in two images. This function detects motion by comparing a patch of current image with a patch of previous image at the same location. In order to recover the depth of two or more stereo images in different viewpoints, binocular disparity is used. This technique is used to match the density or patch of two stereo images. All three algorithms used greatly increases the robustness of gesture and motion tracking. Unlike other applications that depend on fragile appearance model like skin color models or markers, which is greatly affected by environmental conditions or when new users use the application. However, the research is conducted by assumption that the cameras are parallel and users have to be close to the application in order to use it. If the range of users and cameras are to be extended, the position and distance of cameras (baseline) have to be known.

Besides the application like kiosk, gesture and motion tracking also implemented in Augmented Reality (AR). AR refers to a live direct or indirect view of a physical real-world environment whose elements are augmented by virtual computer-generated imagery. This is showed by Lee [3] for the demonstration of a system that track users' bare hand and use it as the reference pattern for AR inspection. The system operates by using a wearable camera. In real time, the camera captures the image of users' hand and the hand regions are segmented. In this process, users' skin model is constructed in one time calibration. Then the hand model is constructed for fingertip detection and tracking. In the segmentation process, the hand region color histogram is learned. The histogram is learned only when the hand and fingertip is in camera's view. Then the segmentation result is used for main hand region tracking. Curvature-based algorithm and ellipse fitting algorithm are used for fingertip detection. Users have to hold their hand up for initial detection. However, due to the reason that false fingertip detection may occur, only the most frequent detected points above the center of users' hand for a number of consecutive frames are selected as final fingertips. While for fingertip tracking, they use fingertip in newly detected frame to match with the previous detected fingertip frame.

2.3 System Overview of Webcam based Marker Tracking Application for Human-Computer Interaction

Object Tracking Application will use the idea by Wilson[12] about his research about FlowMouse, which is his idea that avoids the environmental conditions that affect very much in object tracking. Other journals are quite advanced in the idea of object tracking, which they do not use object tracking. Instead they use gesture tracking that is better in applying human computer interaction concept. Furthermore, gesture tracking is affected very much by environmental conditions as it detects human hand gesture without any markers. Besides that, the algorithm used in computing gesture tracking is more complicated and complex which consumes a lot of processor processing resources.

2.4 Comparison between Existing System and Current System

Table 2.1 System Comparison Table

System	Application	Algorithm Used	Advantages	Disadvantages
FlowMouse by Wilson [12]	- desktop application.	- use optical flow for object tracking. - use Block Matching Algorithm to compute optical flow.	- avoid difficulties such as lighting conditions, skin color and hand shape model development, and segmentation of separating foreground from background.	- users have to enable and disable the application by manual.
3D Interactive Kiosk System by Corey [6]	- kiosk system.	- use skin model (Hue & Saturation) for gesture tracking. - use CamShift Algorithm to detect users'	- users able to move object from left to right, upward to downward and vice-versa, rotating object and scaling	- affected by environmental conditions.

		face, combination of basic of Mean Shift algorithm with an adaptive region-sizing step.	object by zooming in and out.	
GWINDOWS by Wilson [11]	- desktop application	<ul style="list-style-type: none"> - multiple hypotheses tracking approach for gesture tracking. - use simple image comparison function, the sum of absolute differences (SAD) over square patches in two images for motion tracking. - binocular disparity is used to match the density or patch of two stereo images. 	- high robustness in gesture and motion tracking	<ul style="list-style-type: none"> - close range between user and application. - the distance of two parallel camera have to be determined in order to get the actual coordinate for object tracking.
Handy AR by Lee [3]	- desktop application	<ul style="list-style-type: none"> - use color histogram. - for fingertip detection, Curvature-based algorithm and ellipse fitting algorithm are used. 	- fast and accurate gesture tracking.	- restriction for initializing gesture tracking
Current System	- desktop application.	<ul style="list-style-type: none"> - use grayscale for image segmentation. - use simple 	- fast object tracking.	<ul style="list-style-type: none"> - use marker as object detection. - low

		block matching algorithm for motion tracking.		accuracy.
--	--	---	--	-----------

2.5 Conclusion

As computer become more and more integrated in our daily lives, we can expect to find users in a wide variety of contexts where traditional mouse and keyboard interfaces are awkward, too intrusive, or unavailable. The uses of marker in this paper that I proposed is considered troublesome as more and more research conducted by others already use markerless pointing object as input. Users can only use their own hand or finger to interact with the computer. In this way, users can naturally interact with computer as the purpose of human computer interaction, enable human to interact with computer in comfortable and satisfaction. While from the aspect of algorithm used for motion detection or motion tracking, there are many types of algorithm developed by previous researcher and they are widely available in the internet. We only need to determine which type of algorithm suitable to be implemented in our application.

CHAPTER 4

IMPLEMENTATION

4.1 Introduction

The main purpose of this chapter is to document all the processes that are involved in the application development. Generally, this chapter explained about how the application is being designed and developed. Besides that, designated interface and coding being implemented in the application are being included too.

This application consists of two major modules which are marker tracking and sketching using marker. In marker tracking, the processes involved are camera detection and mouse event. While in sketching, the processes involved are marker event and save images as bitmap.

4.2 Marker Tracking

In order for this function to work, two libraries are required to be included. One of the libraries is Touchlesslib.dll. First, locate the library file.

After the library has been added, it will appear at the Reference of the under development.

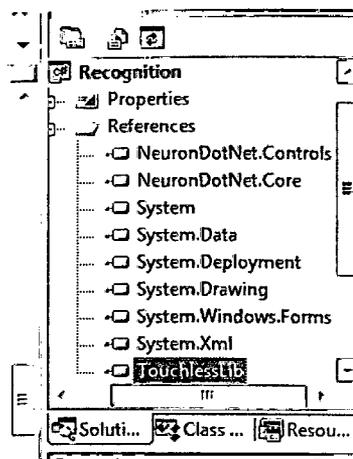


Figure 4.3 Reference added

The next library is WebCamLib.dll. This library does not need to be added as reference in the IDE tool. It just has to be copied to the bin\Debug folder.

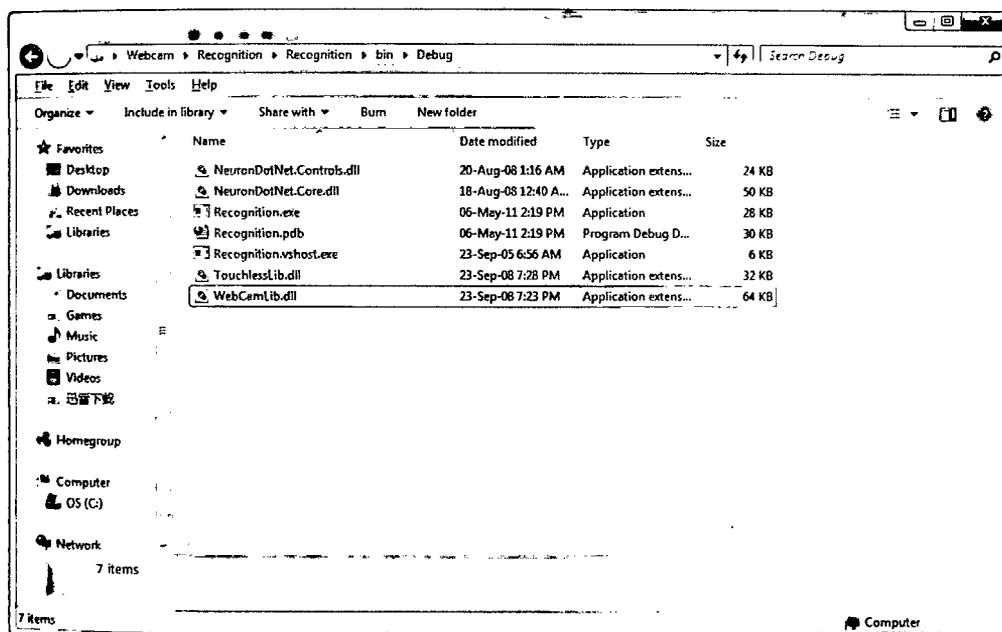


Figure.4.4 WebCamLib Library

4.2.1 Camera Detection

Before the marker is being added to the application, it first has to detect camera. The code that is implemented in this application is to directly use the first camera detected by the computer. Figure 4.5 show the code implemented to detect camera. After that, the image captured will be displayed in a Picture Box. An error handling is also added if there is no camera is detected, it will show an error message to the user.

```
private void Form1_Load(object sender, EventArgs e)
{
    try
    {
        foreach (Camera c in _touch.Cameras)
        {
            if (c != null)
            {
                _touch.CurrentCamera = c; ;
                c.OnImageCaptured += new EventHandler<CameraEventArgs>(c_OnImageCaptured);
                _layout = new Bitmap(c.CaptureWidth, c.CaptureHeight, PixelFormat.Format24bppRgb);
                _layout.MakeTransparent();
                break;
            }
        }
        pictureBox1.Paint += new PaintEventHandler(pictureBox1_Paint);
    }
    catch (Exception)
    {
        MessageBox.Show("Camera not Detected!");
        Environment.Exit(0);
    }
}
```

Figure 4.5 Detect Camera Code

4.2.2 Marker Adding

When user wants to add marker, user can use any object with significant color that differ from background color. After that, use mouse to click on the colored object and move to create a circle. The circle will show the radius of the mouse movement.