

Virtual Touch Screen Using Wiimote

Omkar Pawar¹, Yogesh Jain², Suraj Kanchugar³

KCCEMSR,Thane, India

¹omkarpawarfull@gmail.com, ²yogij811@gmail.com, ³kksuraj27@gmail.com

Abstract- This paper introduces a touch screen technology based on infrared optical. The aim of it is to demonstrate how the existing technology of Wii Remote be used to convert a normal screen to a touch screen. This technology has apparent advantages in the large-size applications, which is simple, low cost, high feasibility. We use C# and the latest programmability features available in Wiimote library files. Using Nintendo Wii Remote we can convert any display screen into a touch screen.

Keywords- Wiimote, light pen, Nintendo

I. INTRODUCTION

Image processing is the domain of this technology. Object tracking, an important part of image processing is implemented in this project. The project consists of Wiimote, also known as wiimote, and light pen as an important components of the project. Wiimote tracks the movement of the infrared light emitted by the light pen and according to it the pointer of the mouse is moved. Using wiimote any display screen can be converted into a touch screen. The project uses C# as the programming language and the Wiimote Library files are also being used. Using the above specified technologies and components the project can convert any display screen into a touch screen.

II. A BRIEF HISTORY OF WIIMOTE

Development of a motion enabled controller began in 2001, coinciding with development of the Wii console. In that year, Nintendo licensed a number of motion-sensing patents from Gyration Inc., a company that produces wireless motion sensing computer mice. Nintendo then commissioned Gyration Inc. to create a one-handed controller for it, which eventually developed the "Gyropod", a more traditional gamepad which allowed its right half to break away for motion-control. At this point, Gyration Inc. brought in a separate design firm Bridge Design to help pitch its concept to Nintendo. Under requirement to "roughly preserve the existing Game Cube button layout", it experimented with different forms "through sketches, models and interviewing various hardcore gamers". By "late 2004, early 2005", however, Nintendo had come up with the Wii Remote's less traditional "wand shape". Nintendo had also decided upon using a motion sensor, infrared pointer, and the layout of the buttons and by the end of 2005 the controller was ready for mass production.

III. SENSING

The Wii Remote has the ability to sense acceleration along three axes through the use of an ADXL330 accelerometer. The Wii Remote also features a PixArt optical sensor, allowing it to determine where the Wii Remote is pointing. Unlike a light gun that senses light from a television screen, the Wii Remote senses light from the console's Sensor Bar (model number RVL-014), which allows consistent usage regardless of a television's type or size. The Sensor Bar is about 20 cm (7.9 in) long and features ten infrared LED'S, five at each end of the bar. The LEDs furthest from the center are pointed slightly outwards, the LEDs closest to the center are pointed slightly inwards, while the rest are pointed straight forward. The Sensor Bar's cable is 353 cm (11 ft 7 in) in length. The bar may be placed above or below the television, and should be centered. If placed above, the sensor should be in line with the front of the television, and if placed below, should be in line with the front of the surface the television is placed on. It is not necessary to point directly at the Sensor Bar, but pointing significantly away from the bar will disrupt position-sensing ability due to the limited viewing angle of the Wii Remote.

Use of the Sensor Bar allows the Wii Remote to be used as an accurate pointing device up to 5 meters (approx. 16 ft) away from the bar. The Wii Remote's image sensor is used to locate the Sensor Bar's points of light in the Wii Remote's field of view. The light emitted from each end of the Sensor Bar is focused onto the image sensor which sees the light as two bright dots separated by a distance "mi" on the image sensor. The second distance "m" between the two clusters of light emitters in the Sensor Bar is a fixed distance. From these two distances m and mi, the Wii CPU calculates the distance between the Wii Remote and the Sensor Bar using triangulation. In addition, rotation of the Wii Remote with respect to the ground can also be calculated from the relative angle of the two dots of light on the image sensor.

IV. USING WIIMOTE LIBRARY

Brian Peek demonstrates how to connect to and use the Nintendo Wiimote from C# and VB.NET. The final output is an easy-to-use managed API for the Wiimote that can be used in any managed application.

Software: Visual C# or Visual Basic 2008 Express Editions

Hardware: Nintendo Wii Remote(Wiimote) , a compatible PC Bluetooth adapter and stack

The Wii Device Library is a cross platform C# class library that provides an interface to various Wii related devices, like the Wiimote, Classic Controller, Nunchuk and Balance Board.

It's possible to use Wii devices like the Wiimote or Balance Board with your computer. All you need besides one of these devices is some kind of bluetooth device. This could either be a bluetooth dongle or bluetooth built into your pc or laptop.

When using the Wii Device Library you don't have to worry about all the bluetooth stuff, you can just access your device of choice through a simple and intuitive interface. Also new Wii devices and extensions can be implemented by this interface without recompiling Wii Device Library.

Another nice feature is the ability to scan for Wii devices through bluetooth and connect to them when they are available. This makes it possible to automatically connect to Wii devices when they are syncing. It uses the libraries of several bluetooth stacks to scan, connect and communicate with the Wii devices. Also this part of the library can be extended with support for more bluetooth stacks and/or operating systems.

available, try the font named Computer Modern Roman. On a Macintosh, use the font named Times. Right margins should be justified, not ragged.

V. REPORT TYPES

The library currently supports only a handful of the many report types the Wiimote is capable of producing, however the ones that are implemented are enough to get all required data for the Wiimote and all current extensions. Specifically, these reports are:

Buttons - Button data only

ButtonsAccel - Button and accelerometer data

IRAccel - Button, accelerometer and IR data

ButtonsExtension - Button and extension data

ExtensionAccel - Button, accelerometer and extension data

IRExtensionAccel - Button, accelerometer, extension and IR data

The report type can be set by calling the SetReportType method using the appropriate report type and whether or not

you would like the data to be sent continuously or only when the state of the controller has changed

VI. USAGE OF API

You will need to add a reference to the WiimoteLib.dll, which can be downloaded from the internet.

Next, you will need to pull the namespace into your application with the standard using/Imports statement. With that in place, you can now create an instance of the Wiimote class and start using it. Simply instantiate a new instance of the Wiimote class, setup events if you wish to use them, setup the report type of the data you want to be returned, and call the **Connect** method.

C#

using WiimoteLib;

```
private void Form1_Load(object sender, EventArgs e)
{
    // create a new instance of the Wiimote

    Wiimote wm = new Wiimote();

    // setup the event to handle state changes
    wm.WiimoteChanged += wm_WiimoteChanged;

    // setup the event to handle insertion/removal of extensions
    wm.WiimoteExtensionChanged +=
    wm_WiimoteExtensionChanged;

    // connect to the Wiimote
    wm.Connect();

    // set the report type to return the IR sensor and
    accelerometer data (buttons always come back)
    wm.SetReportType(Wiimote.InputReport.IRAccel, true);
}
```

Data can be retrieved from the API in two ways: events and polling. In event mode, one must subscribe to the WiimoteChanged event as shown above. Then, when data is sent from the Wiimote to the PC, an event will be posted to your event handler for processing in your application. If you elect to not use the event model, you may simply retrieve state information at any time from the WiimoteState property of the Wiimote class.

VII. LIGHTPEN

A light pen is a computer input device in the form of a light-sensitive wand used in conjunction with a computer's CRT display. It allows the user to point to displayed objects or draw on the screen in a similar way to a touch-screen but with greater positional accuracy. But in our project we actually

don't implement the light pen but we have a pen which functions similar to the light pen. The pen which is used in the project is constructed manually so that the user can operate the screen. The user will be able to select, drag, draw, write etc. onto the screen with this pen. This will provide the feeling of touch-screen to the user.



Figure 1. Lightpen

The Figure above is the sample schematic of the light pen. The LEDs that we use are running at 100mA, but lots of other LEDs will work too.

VIII. WIIMOTE

Nintendo's Wii Remote (forever known as the Wiimote) is a fantastic little controller for the Nintendo Wii system. Because it uses Bluetooth to communicate with the Wii, it can be connected to and used by practically any Bluetooth capable device.

As of June 2008, Nintendo has sold nearly 30 million Wii game consoles. This significantly exceeds the number of Tablet PCs in use today according to even the most generous estimates of Tablet PC sales. This makes the Wii Remote one of the most common computer input devices in the world. It also happens to be one of the most sophisticated. It contains a 1024x768 infrared camera with built-in hardware blob tracking of up to 4 points at 100Hz. This significantly outperforms any PC "webcam" available today. It also contains a +/-3g 8-bit 3-axis accelerometer also operating at 100Hz and an expansion port for even more capability. These projects are an effort to explore and demonstrate applications that the millions of Wii Remotes in world readily support.



Figure 2. Wiimote (or wiiremote)

IX. SETUP

System setup requires a Wiimote to be focused on the screen. Wiimote angle of sight is 45°. Wiimote works well at a distance of 5 meters from the display. Wiimote needs to be stationed properly at one position.



Figure 3. Setup of wiimote

The calibration screen provides Wiimote the alignment and the distance from the screen. After the calibration the Wiimote should not be moved and if moved the calibration of the screen should be done again.

The light pen should be visible to the Wiimote, as the infrared led should be detected by the Wiimote. It tracks its position on the screen and provides the x,y co-ordinate to the system to move the cursor to the co-ordinate on the display screen.

X. RESULT

With proper calibration of the display screen and Wiimote kept stationary the system performs as accepted. The following figure shows the actual implementation of the system.



Figure 4.Result of setup

The same system can be implemented using a projector screen display

XI. PROS

1. PROXIMITY TO THE SCREEN

If you are using a touch-screen monitor you have to be close to the screen in order to touch it. Virtual touchscreen does not require the person to be close to the screen. It only requires the person to be in the view of the Wiimote.

2. ACCURACY

Virtual touchscreen uses a light-pen so it provides greater accuracy than operating the screen by finger. Also the sensitivity of the Wiimote helps further in gaining accuracy.

3. SPEED

Virtual touchscreen eliminates the use of sensors in the screen, which is a need of touchscreen monitor hence it removes the sensor processing. As we are using OpenCV library files we are dealing with real-time processing hence there is further saving time. The tracking of the object using a Wiimote is also very fast which further enhances the speed.

4. SIMPLICITY

Virtual touchscreen is easy to use as compared to touchscreen monitor. The light pen helps in using the virtual touchscreen much easily.

5. SCALABILITY

The screen size can be scaled to any limit provided the light pen is detected by the Wiimote.

6. DISPLAY

VTS can work on any screen be it CRT, LCD, LED or a projector screen.

7. CHEAP

Virtual touchscreen is much cheaper as compared to a touchscreen monitor which makes it cost efficient in terms of touchscreen monitors.

XII. CONS

1. Once the screen is calibrated by the Wiimote you cannot move the Wiimote.

2. The light pen must always be in the view of Wiimote or else it will not track the movement. The light pen must clearly be visible to the Wiimote.

3. Positioning the Wiimote is important is because if the position of the Wiimote is not proper it will not track the light pen clearly.

4. Wiimote needs to be charged before it is used.

XIII. FUTURE OF VIRTUAL TOUCHSCREEN

1. The most important future scope to this system is with "Hologram" projector. A Hologram Projector is a technology being experimented. It displays a 3D object image of a real world. Using the light pen touching the hologram can be possible.

2. Using "Chalk" for black board and white board pen are waste of stationary but the proposed system can help to do the same thing but without using any stationary. It requires only a white board and a projector. The wiimote will calibrate the screen and you can write using the light pen.

REFERENCES

- [1] Johnny.Chung.Lee, "tracking infrared light with Wiimote", <http://johnnylee.net/projects/wii/>
- [2] Brian Peek, "Managed Library for Nintendo's Wiimote", <http://channel9.msdn.com/coding4fun/articles/Managed-Library-for-Nintendos-Wiimote1>
- [3] Wikipedia, "Wii Remote", <http://en.wikipedia.org/wiki/Wiimote>
- [4] "Controlling the Wiimote with C# or VB.NET", <http://arstechnica.com/civis/viewtopic.php?f=41&t=1100564>

- [5] [Uber Maker](http://www.instructables.com/id/How-To-Connect-your-Wiimote-to-your-PC-via-Bluetooth/),” How To: Connect your Wiimote to your computer via Bluetooth”, <http://www.instructables.com/id/How-To-Connect-your-Wiimote-to-your-PC-via-Bluetooth>

IJERT