Shahjalal University of Science and Technology, Sylhet.
Department of Computer Science and Engineering

Motion Tracking Using Gyro and Accelerometer For Gaming Applications: Developing a Software Based Sensor

Student: Md. Mustafizur Rhaman Shakil (2010331055)
Foysal Ahmed Emon (2010331046))
Adviser: Sabir Ismail, Lecturer, Department of Computer Science and Engineering

28th March, 2015

Motion Tracking Using Sensor Technology For Gaming Applications: Developing a Software Based Sensor

A Thesis/Project submitted to the Department of Computer Science and Engineering, Shahjalal University of Science and Technology, in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering.

Student: Md. Mustafizur Rhaman Shakil (2010331055)
Foysal Ahmed Emon (2010331046)
Adviser: Sabir Ismail, Lecture, Department of Computer Science and Engineering
28th March, 2015

Recommendation Letter from Thesis Supervisor

This Student, Foysal Ahmed Emon and Mustafizur Rhaman Shakil, whose thesis entitled "Motion Tracking Using Gyro and Accelerometer for Gaming Applications: Developing a Software Based Sensor , is under my supervision and agree to submit for examination.

Advisor : Sabir Ismail , Lecturer, Department of Computer Science and Engineering

Date : 28th March, 2015

Qualification Form of Bachelor Degree

Student Name : Foysal Ahmed Emon, Mustafizur Rhaman Shakil.

Thesis Title : Motion Tracking Using Gyro and Accelerometer for Gaming Applications: Developing a Software Based Sensor.

This is to certify that the thesis submitted by the student named above in 28th March, 2015. It is qualified and approved by the Thesis Examination Committee.

----------------------	----------------	---------------------------
Head of the Dept.	Chairman, Thesis Committee	Supervisor

ABSTRACT

Think of a new world. The world of limitless possibilities. People don't have to sit on table with keyboard and mouse to do their daily work. Teachers can just stand up and end his daily class lecture with some physical movement in the classroom. Just we need a new tech to make everything come to reality. Mobile sensor is not a newcomer in the tech world. It's been many years since mobile phone using various kinds of sensors. We are doing some fusion of those sensors and compiling that sensor to make a new type of motion sensor. The accelerometer sensor and gyroscope sensor is used in various purposes. We are introducing a combination of Gyroscope and accelerometer. The Gyroscope and accelerometer works in different way of their own functionality. As we know no sensors are strong and without drawbacks , every sensor has their drawbacks Like Accelerometer and Gyroscope has theirs own drawbacks.So we look in to these sensors pros and cons found that accelerometer has gravity issuse which we can not remove frm accelerometer motion sensor and gyroscope sensor has accumalted error but no gravity issue because it uses phone frame of reference.So we combine thses two sensor and did a huge mergin of filtering and use some algorithm and techniques and managed to solve this kind of errors and gravity issues and made a New Linear Accelerometer Sensor Which is based on these accelerometer and gyroscope sensor.

Acknowledgement

We are using this opportunity to express our gratitude to everyone who supported us throughout the course of this Thesis. We are thankful for their aspiring guidance, invaluably constructive criticism and friendly advice during the thesis work. We are sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to the thesis.

We express my warm thanks to **Sabir Ismail** for his support and guidance at my thesis work.

And mostly Google, Google Talk Tech, Android Documentation, Unity3D Documentation and Some Research Paper Like Kaleb Kircher helped us a lot for those knowledge and experiment to get them together.

Finally we want to thank Almighty to allow me this knowledge and passion.

Thank you,

Foysal Ahmed Emon & Md. Mustafizur Rhaman Shakil

TABLE OF CONTENTS

List of Figure

Illustration Of Symbols

LPF : Low Pass Filter
HPF : High Pass Filter
UDP : User Datagram Protocol
Gyro : Gyroscope Sensor
GUI : Graphic User Interface
Acc : Acceleration
LAcc: Linear Acceleration

Introduction

The new age of mobile applications is expanding with tons of new features and possibilities. Human is mostly depends on mobile phone applications. Though we need to make sure we can make maximum output of the current technology for our daily life. Mobile phone is a handy device. It comes with many sensors integrated. We can make those sensors work on our own way by combining them with some functionality. It can do lots of staffs when we use them in our own way.

1.1 Background

The purpose of the task is to work with mobile sensor like accelerometer, gyroscope. It’s really a problem people doesn't know how to use them, they don’t necessarily know how to work with them and how to take all data together. We are trying to introduce some new types of sensor which is made from the fusion on the existing sensors.

1.2 Motivation :

Gaming and other motion sensor application needs those motion sensors and gyroscopes. We found that we can take the data for both sensors and put it together and do some filtering to get data. Those data we can use for further applications. We are motivated from the gaming sensors widely used now a days. We found that the available device is costly. We want to introduce only a application that can solve this problem of all motion based problems .

History of Sensors

Sensors have been around for quite some time in various forms. The first thermostat came to market in 1883, and many consider this the first modern, man-made sensor. Infrared sensors have been around since the late 1940s, even though they’ve really only entered the popular nomenclature over the past few years. Motion detectors have been in use for a number of years. The answer is going to depend on the exact type of sensor in question. For the electric thermostat, the inventor would be Warren S. Johnson. While it might have seemed crude by the modern standards that we have today,

this thermostat was able to keep temperatures sensors within a degree of accuracy – something that's better than some of the low quality thermostats on the market today!
The first motion sensor used for an alarm system came about in the early part of the 1950s, and was the invention of Samuel Bagnio. His device made use of ultrasonic frequencies as well as the Doppler Effect.

1990
- First battery protector introduced
- First family of SP Combination Start and Protect Devices introduced

1992
- Low-g accelerometer introduced
- TIRIS read/write transponder debuts

1994
- Transmission Range Sensor production launches for Ford
- TI-RFID vehicle immobilizer developed

1995
- Development of DuraFoil™ material

1996
- Plant opens in Baoying, China
- Construction begins on new facility in Aguascalientes, Mexico

1997
- Mobil Oil selects TIRES for use in Speedpass pay-at-pump transactions
- Industry's first c24x low-cost motor controller introduced
- Land Grid Array (LGA) burn-in sockets selected by Intel for Pentium II
- High-Volume Manufacturing of 0.5mm pitch Chip Scale Package (CSP) begins

1998
- Wholly-owned China manufacturing operation opens in Baoying
- TIRIS announces Tag-It™ SMART label technology

1999
- Production of DuraFoil™ material begins
- Subcontracting agreement signed with Videoton in Hungary
- Tag-It™ SMART label low-cost RFID Transponder introduced
- Micro-Silicon Strain Gauge sensor introduced

2003

• Automotive passive entry system using an RFID-enabled key created

2004

• ARC Shield™ introduced (fault circuit Interrupter)

• Second plant in China opens in Changzhou Province after acquisition of the motor protector business of Chengcheng Company.

The Most important part is The smartphone sensors. We are only focusing the android and iphone smartphone sensors , though nokia symbian smartphone is the first one which uses the accelerometer sensor for their mobile but today these days nokia symbian smart phones are depreciated. Every Single person uses Android Or IPhone.
The Android platform provides several sensors that let We monitor the motion of a device. Two of these sensors are always hardware-based (the accelerometer and gyroscope), and three of these sensors can be either hardware-based or software-based (the gravity, linear acceleration, and rotation vector sensors). For example, on some devices the software-based sensors derive their data from the accelerometer and magnetometer, but on other devices they may also use the gyroscope to derive their data. Most Android-powered devices have an accelerometer, and many now include a gyroscope. The availability of the software-based sensors is more variable because they often rely on one or more hardware sensors to derive their data.

Motion sensors are useful for monitoring device movement, such as tilt, shake, rotation, or swing.
The movement is usually a reflection of direct user input (for example, a user steering a car in a game or a user controlling a ball in a game),
but it can also be a reflection of the physical environment in which the device is sitting (for example, moving with you. while We drive your car).
In the first case, We are monitoring motion relative to the device's frame of reference or We are application's frame of reference;
in the second case We are monitoring motion relative to the world's frame of reference.
Motion sensors by themselves are not typically used to monitor device position,
but they can be used with other sensors, such as the geomagnetic field sensor,
to determine a device's position relative to the world's frame of reference.

Background Story

3.1 Sensor Type

There are lots of sensors available. But some mobile has some sensors and other has some. Latest Smart Phones got the latest sensors available.
In real life there is a lot of sensors available but in smart phone many are introduced because making smart phones sensors are not that easy because smart phones
are small and sensors has to be low budget in making, So here are some of the sensors which is introduced in the field of smart phones like Android and iPhone

Motion sensors

- Sensors that can be used to detect motion
- Location sources (WiFi, GPS, cellular network)
- Light sensors (e.g. proximity detection)
- Accelerometer Sensors
- Gyroscope Sensors

Environmental Sensors

- Light Sensors
- Pressure Sensors
- Pressure Sensors
- Temperature Sensors
- Humidity Sensor

Position Sensors

- Geomagnetic Field Sensors
- Orientation Sensor
- Game Rotation Vector Sensor
- Geomagnetic Rotation Vector Sensor

3.2 How Sensor Works

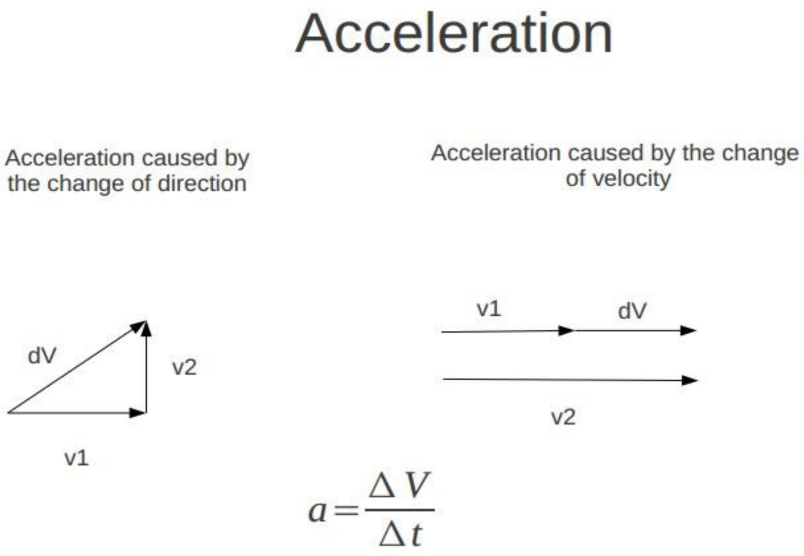


FIG : 1

- First smart-phone with built-in accelerometer: Nokia 5500 “sport device”: 2005 Q3 (Symbian)
- Android supports accelerometer sensor even from pre-1.0 API
- One frequent type in Android phones: Bosch BMA150
- One-chip solution with internal electromechanic structure
- -2g - +2g/-4g - +4g/-8g - +8g (10 bit resolution)
- “Any motion” interrupt mode
- Built-in thermometer

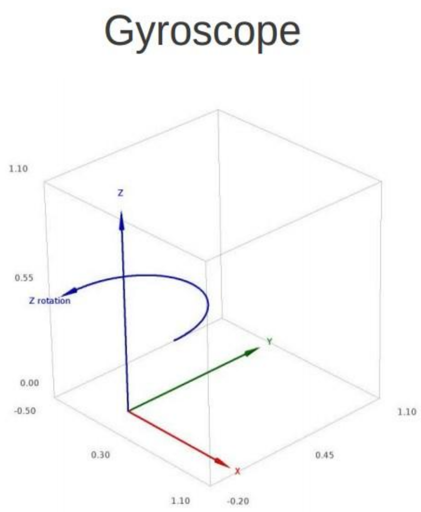


FIG : 2

- Very new phenomenon as gyroscopes suitable for consumer electronic devices appeared very recently
- First appearance: Wii Motionplus accessory, 2009 June
- Android supports gyroscope sensor even from pre-1.0 API
- First Android smartphone: Nexus S (end of 2010)
- Example: InvenSense MPU-3000
- One-chip solution with internal electromechanical structure
- Measures rotation along 3 axes
- 16 bit resolution

Compass :

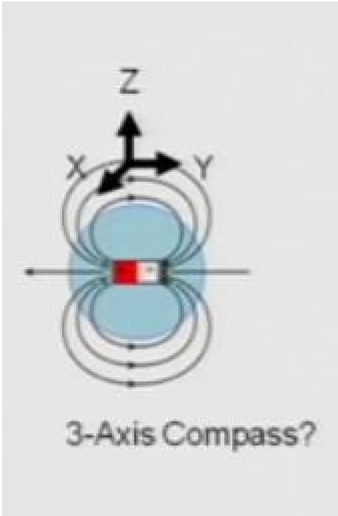


FIG: 3

- Measures the device orientation wrt. the magnetic vector of the Earth
- This vector points toward the magnetic center of the Earth – mainly down
- It has a component that points to the magnetic North pole – that's what we use for orientation
- If, however, the device is not held horizontally, the downward vector element influences the measurement
- Also sensitive for all sorts of metal objects
- Consequence: can be used for motion detection only in special cases

Android sensor support

- In android.hardware.SensorManager
- Listing sensors
- Sensor sampling
- Some processing functions like vector transformations
- In competing platforms:
- Snap, shake and double tap (with accelerometer)

detection in bada
● Shake detection in iOS

3.3 Mobile Sensor

1. Accelerometer Sensor
2. Gravity Sensor
3. Gyroscope
4. Linear Accelerometer
5. Rotation Vector Sensor
6. Significant Motion Sensor
7. Step Counter Sensor
8. Step Detector Sensor
9. Light Sensor
10. Pressure Sensor
11. Temperature Sensors
12. Humidity Sensor
13. Motion Sensor
14. Compass Sensor
15. Proximity Sensor etc.

3.4 New Sensor Technology

1. Fingerprint Sensor
2. Eye Detect Sensor
3. I-Beacon Technology
4. Blood Sensor
5. Life Detection Sensor
6. Smell Sensor

CURRENT TECHNOLOGY

In this days Accelerometer is the renowned and old sensor technology and it is vastly used by all kinds of smart phones. In 2009 a new sensor was introduced which changed the way it was and the name of that sensor is Gyroscope Sensor.
Accelerometer , Gyroscope, Compass, Magnetic, Vector Rotation Sensors are commonly used sensors.

Microsoft Kinect for Xbox 360

The system: Thanks to a sophisticated depth-sensing camera (actually a single color camera for image recognition, and two monochrome cameras placed a few inches apart to determine where you are in a three-dimensional space), Kinect can track your movements without a physical controller. All of the heavy lifting is handled by the Kinect sensor and the console, and you can navigate menus and play games without laying a finger on a piece of plastic. A microphone array adds voice recognition to the mix, letting users control the system using voice commands or hand-waves. Because the system is camera-only, it needs a lot of space; Microsoft recommends 6 to 8 feet between the Kinect sensor and the user.

Sony PlayStation Move

The system: The PlayStation Move combines a video camera with a physical controller packed with motion-sensing electronics, making it the technological cross between Kinect and the Nintendo Wii. The Move Motion Controller, or "wand," combines a gyroscope, accelerometer, and magnetic sensor (a sort of digital "compass" that uses the Earth's magnetic field to determine the controller's orientation) to track the controller in three dimensions, while the glowing ball at the end gives the PlayStation Eye camera a visual reference for handling aiming, cursor movement, and other motion. Like Kinect, PlayStation Move requires room to function; Sony recommends 5 to 9 feet between the player and the Playstation Eye, but you can play anywhere from 2 to 10 feet of the camera.

Nintendo Wii

The system: For the Wii, all the motion-control magic is in the remote. An accelerometer tracks movement, while an IR sensor monitors the positioning of lights emitted by the sensor bar. Its motion-sensing abilities weren't so great at first; initially, your movements with the Wiimote were reflected only approximately in games with gestures and broad motions. The addition of Wii MotionPlus, an accessory that gives the Wiimote a gyroscope sensor to complement the accelerometer, improves the motion detection greatly. Nintendo recently began to sell the Wii Remote Plus, a Wiimote with built-in MotionPlus sensors, removing the need for a separate accessory. The Wii's biggest weakness (weakness?) is its graphics; unlike the PlayStation 3 and Xbox 360, the Wii doesn't display high-definition content.

Paper Works :

Using this technology there is some paper work we found like

1 . "A Review on System Architectures for Sensor Fusion Applications Wilfried Elmenreich
Vienna University of Technology Treitlstrasse 3, 1040 Vienna, Austria
wil@vmars.tuwien.ac.at

2 .A Multi-Sensor Fusion System for Moving Object Detection and
Tracking in Urban Driving Environments
Hyunggi Cho, Young-Woo Seo, B.V.K. Vijaya Kumar, and Ragunathan (Raj) Rajkumar

3.Context-Aware Personal Navigation Using Embedded Sensor Fusion in Smartphones
Sara Saeedi *, Adel Moussa and Naser El-Sheimy

4. http://www.intechopen.com/books/sensor-fusion-and-its-applications

Limitations of Current technology:

Cost:
The mentioned system available in the market is costly. Like all the sensors come with extra hardware device and setup.

System Requirements:
What we use now needs highly required hardware and setup.

Methodology

Gyroscope measurement data
● It Measures rotation around 3 axes
● actually: measures rotation speed (mainly angular velocity)
Around the axes, how the speed of the rotation was.

$$v_x = \frac{\Delta \varphi}{\Delta t}$$

Getting the rotation angle

● Get the angle difference
● Get the absolute angle

$\Delta \varphi = v \; x \; \Delta t$

$\varphi ' = \varphi + \Delta \varphi$

Gyro as support sensor

● Because of accumulating error, gyro alone can be rarely used
● But
  * The accelerometer has no accumulated error but has the gravity component problem
  * The gyro has accumulated error but is not sensitive to gravity

So it seems that gyroscope has some problems and a great flexibility whether acceleration don't have
problems but not so much flexible. If we want to use a kind of sensor which has no accumulated errors and removing the gravity problems then we won't be able to use that.
So our efforts are to make that kind of sensor which won't have any kind of accumulated errors and not sensitive to gravity. We used both these sensors and compensate each other's weakness and we got a new kind of sensor.

Let us see what we have done to achieve this New Sensor.
It seems that we can do this in two ways the easiest way and the hardest way
Easiest Way

- We can use the virtual sensors that calculate gravity and linear acceleration from multiple sensors

Hardest Way

- We can use the raw accelerometer and gyroscope data and get the motion what we need for.

We won't go for the easiest way because some devices don't have Linear Accelerometer Sensor.
Here we are creating our own Linear Acceleration which don't have Linear Accelerometer Sensor.

Accelemotor WorkFlow:

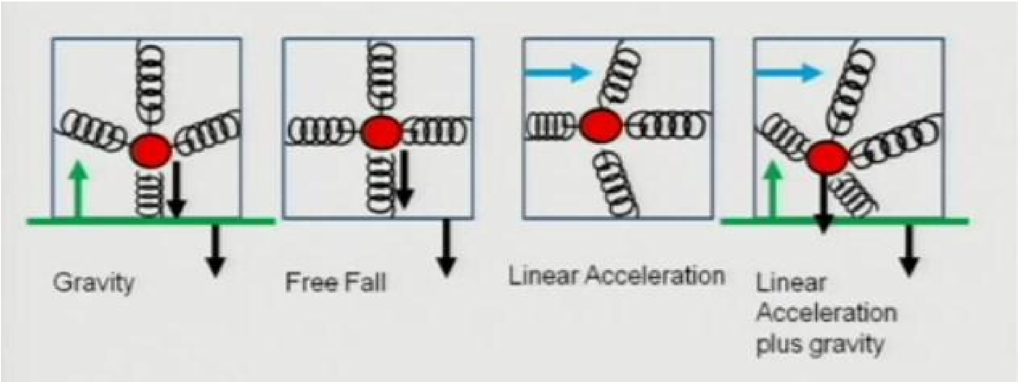


FIG : 4

Here will be the question why we are going for the hardest way why not go for the easiest way.
The answer is simple the easiest way which uses the virtual sensors and those sensors are:

1. Gravity and motion acceleration deduced from the accelerometer and the gyroscope
2. Roll/pitch/yaw from the compass
3. Drift-compensated gyroscope

We can see that virtual sensor which is not available right now. It is totally impossible to get drift compensated gyroscope.

So it seems we have to go for the hardest way. We can't subtract gravity from the accelerometer by getting motion acceleration.(For General Case)

So, Here we did the trick or method we compensate acceleration sensor and gyro sensor and separate gravity and motion acceleration with the help of gyroscope.

Communication

We communicate client to server via wifi. Server sends compressed data to client.We use UDP to send and receive data. Here we use unity3d game engine. We simply send compressed data using their NetworkView which uses RakNet Technology and RakNet uses UDP(User Datagram Protocol)
It sends compressed data through wifi using udp and synchronized data. And guaranteed data to be received and if any data sent is failed then it sends the data again and if some of these data are missing then it sends the missing data.

Security

Here we send data **via** Reliable Delta Compressed which makes the data compressed and confirm if the data has been received or not.

EXPERIMENTAL STUDY

Main Formula:
Linear Acceleration = Acceleration - Gravity

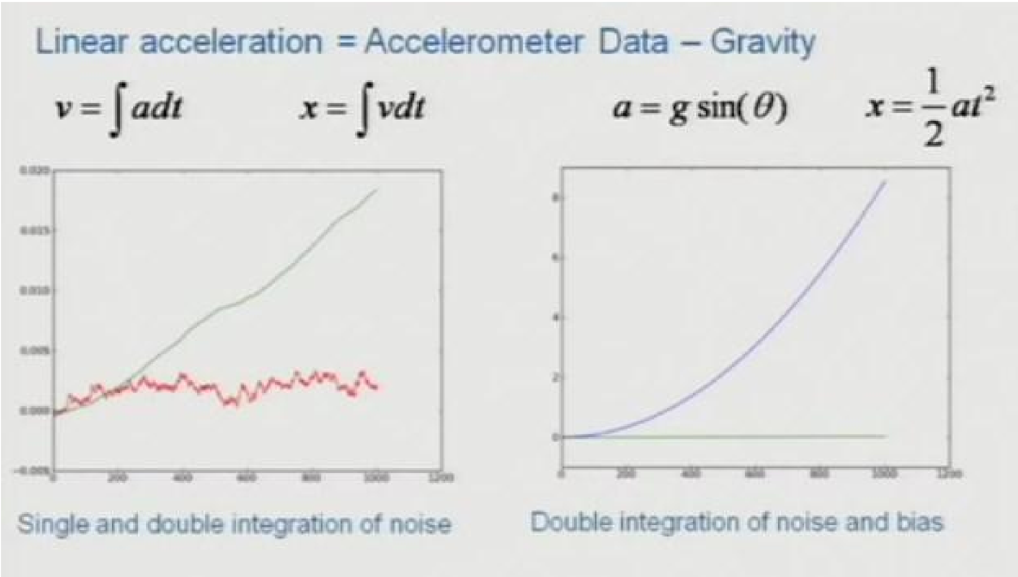


FIG : 5

To achieve what we desire for we try many ways and failed. We tried to remove gravity from accelerometer sensor to get the linear acceleration but we failed. Because we were doing it in the wrong way, it is not possible to remove gravity from accelerometer sensor and get motion. Here is sample code what we tried and didn't work:

```
gravity[0] = gravity[0] + event.values[0];
gravity[1] = gravity[1] + event.values[1];
gravity[2] = gravity[2] + event.values[2];
linear_acceleration[0] = event.values[0] - gravity[0];
linear_acceleration[1] = event.values[1] - gravity[1];
linear_acceleration[2] = event.values[2] - gravity[2];
```

So here we can see that we are not getting the actual data because we didn't able to remove the gravity from the accelerometer so we did some change in the code we see that

using a high pass filter algorithm we can remove the gravity and conversely using a low pass filter algorithm we can remove the gravity also , so we did some change and here is the code :

```
// Isolate the force of gravity with the low-pass filter.
  float alpha = 0.8;(Depends On Device)
  gravity[0] = alpha * gravity[0] + (1 - alpha) * event.values[0];
  gravity[1] = alpha * gravity[1] + (1 - alpha) * event.values[1];
  gravity[2] = alpha * gravity[2] + (1 - alpha) * event.values[2];
```

But without using this Low pass filter we can get the actual raw data of Acceleration but there will be some drawbacks and the drawbacks are we won't be able to remove the gravity from the accelerometer and there will be a lot noise.
So it seems if we use the low pass filter or high pass filter we can reduce noise and get smooth accelerometer and remove the gravity from accelerometer.
There is a saying that if we just double integrate the accelerometer we will get the linear acceleration, it is true but it is not that simple we won't get what we need and it gives a lot of mind breaking errors.

Linear Acceleration From Acceleration

We can also get Linear Acceleration from Accelerometer using no high pass filter or low pass filter , only a mind blowing algorithm.

The Algorithm:

- Calculate the magnitude of the acceleration
- Calculate the variance of the magnitude of the acceleration
- If the magnitude of the acceleration is < 0.8 or > 1.2 assume dynamic acceleration
- If the variance of the magnitude of the acceleration is > 0.1 assume dynamic acceleration
- Else assume static acceleration and calculate the gravity components of the acceleration
- Subtract the last known gravity components of the acceleration from the acceleration to determine linear acceleration.

The Main Code is here:

```
float magnitude = (float) (Math.sqrt(Math.pow(this.acceleration[0], 2)
+ Math.pow(this.acceleration[1], 2)
```

```
+ Math.pow(this.acceleration[2], 2)) / SensorManager.GRAVITY_EARTH);

// Attempt to estimate the gravity components when the device is
// stable and not experiencing any kind of linear acceleration.
if (var < 0.1 && magnitude > 0.8 && magnitude < 1.2)
{
        gravity[0] = this.acceleration[0];
        gravity[1] = this.acceleration[1];
        gravity[2] = this.acceleration[2];
}

Linear_Acceleration[0] = (acceleration[0] - gravity[0])
/ SensorManager.GRAVITY_EARTH;
Linear_Acceleration[1] = (acceleration[1] - gravity[1])
/ SensorManager.GRAVITY_EARTH;
Linear_Acceleration[2] = (acceleration[2] - gravity[2])
/ SensorManager.GRAVITY_EARTH;
```

Linear Acceleration From Both Accelerometer And Gyroscope Sensor Compensating Together

Work In Progress , We are gathering all of our data from above the experiments and we are hoping we will soon be able to reach our goal Using Our Experiments.

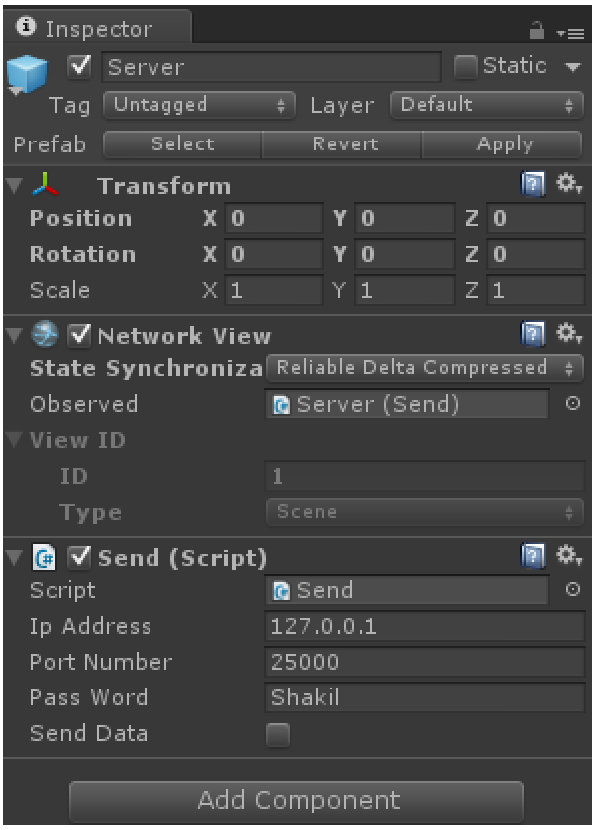


FIG : 6

First of all we made two application one is running on the PC and other is running on the Android Mobile Set. So we made two scene like send and receive scene. In the Send Scene we add NetworkView Component which uses udp and osc or rak net system to connect with client and send compressed data and add our Send.cs (C Sharp) script which is the main work.

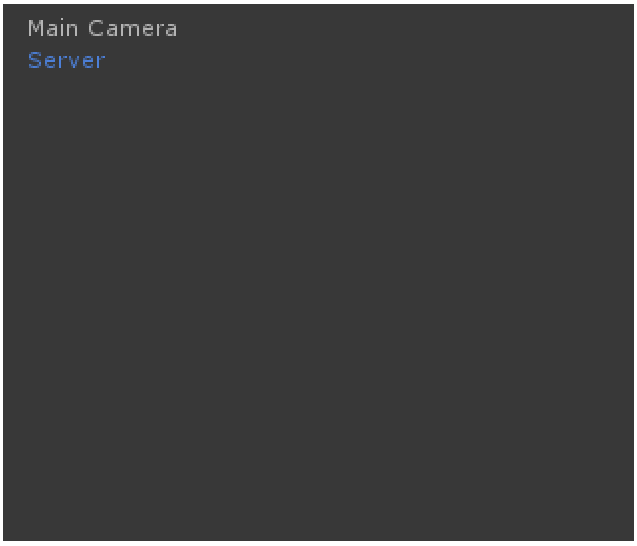


FIG : 7

Here is the server scene of server gameobject which will be running on android mobile and will send data to pc.

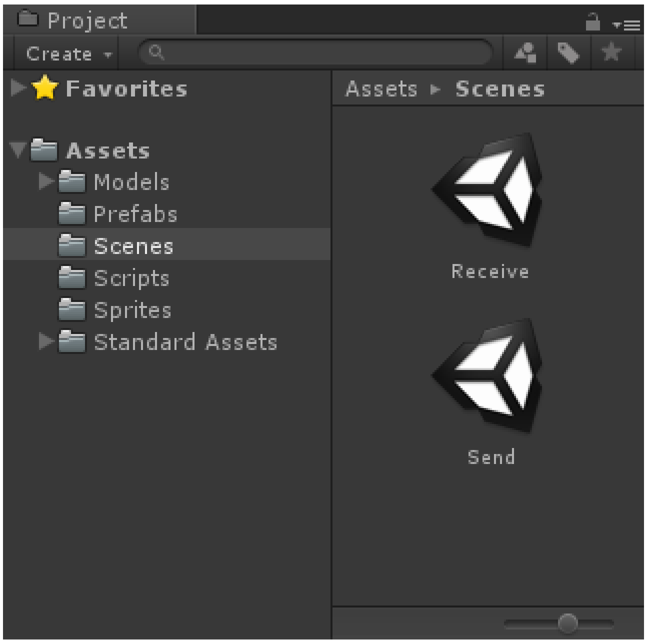


FIG : 8

Here is the two scene , one will be running on PC and other will be on android mobile , so we make the android mobile server and pc the receiver.

```
void OnSerializeNetworkView(BitStream stream, NetworkMessageInfo info)
{
    ////// Gyro Mouse Work In Progress
    //if (stream.isWriting && SendData)
    //{
    //    stream.Serialize(ref InputData);
    //    ServerInfo = "Setting: " + InputData.ToString();
    //}

    //// Accelerometer Without Low Pass Filtering
    stream.Serialize(ref AccelerationWOutFiltering);
    ServerInfo = "Setting: " + AccelerationWOutFiltering.ToString();
}
```

In this function we send our compressed data to the client which is running on PC. We send Vector3 for the acceleration part and Quaternion for Gyro Part. In Quaternion there will yaw pitch and roll.

```
void OnSerializeNetworkView(BitStream stream, NetworkMessageInfo info)
{
    if (stream.isReading)
    {
        ///GyroMouse Work
        //Quaternion InputData = Quaternion.identity;
        //stream.Serialize(ref InputData);
        //GyroMouseData = InputData;
        //ClientInfo = "Getting" + InputData.ToString();

        //// Accelerometer Data Work
        AccelerationValue = Vector3.zero;
        stream.Serialize(ref AccelerationValue);
        ClientInfo = "Getting" + AccelerationValue.ToString();
    }
}
```

This is the desktop part of code , here client receive the compressed data which the server sent to the client.

Running this GameObjects On PC version Scene.

Getting Linear Acceleration from Accelerometer Sensor Using Our Method

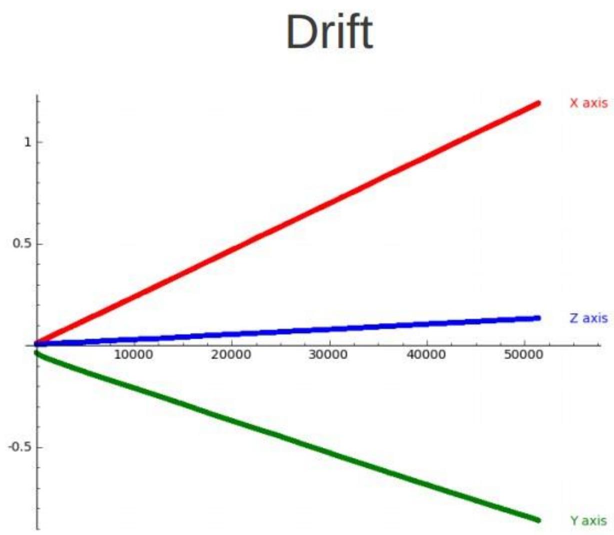
```
Vector3 LinearAcceleration(Vector3 acceleration)
{
    float magnitude = (float)(Mathf.Sqrt(Mathf.Pow(acceleration[0], 2)
        + Mathf.Pow(acceleration[1], 2)
        + Mathf.Pow(acceleration[2], 2)) / GRAVITY_EARTH);

    if (magnitude > 0.8 && magnitude < 1.2)
    {
        gravity[0] = acceleration[0];
        gravity[1] = acceleration[1];
        gravity[2] = acceleration[2];

    }

    Vector3 LinearAcceleration = Vector3.zero;

    LinearAcceleration[0] = (acceleration[0] - gravity[0]) / GRAVITY_EARTH;
    LinearAcceleration[1] = (acceleration[1] - gravity[1]) / GRAVITY_EARTH;
    LinearAcceleration[2] = (acceleration[2] - gravity[2]) / GRAVITY_EARTH;
    return LinearAcceleration;
}
```

```
Vector3 LowPassFiltering(Vector3 input, Vector3 output)
{
    if (output == null) return input;
    for (int i = 0; i < 3; i++)
    {
        output[i] = output[i] + ALPHA * (input[i] - output[i]);
    }
    return output;
}
```

We use our own Low Pass Filtering method

```
// Accelerometer Without Low Pass Filtering
AccelerationWOutFiltering = Input.acceleration;

// Accelerometer With Low Pass Filtering
AccelerationWithFiltering = LowPassFiltering(Input.acceleration, AccelerationWithFiltering);


// Linear Acceleratoin From Acceleration With low Pass Filtering
AccelerationWithFiltering = LowPassFiltering(Input.acceleration, AccelerationWithFiltering);
LinearAccelerationWithFiltering = LinearAcceleration(AccelerationWithFiltering);

//// Linear Acceleratoin From Acceleration With low Pass Filtering
LinearAccelerationWOutFiltering = LinearAcceleration(Input.acceleration);


Quaternion tempQuaternion = Input.gyro.attitude;
AccelerationWOutFiltering = new Vector3(tempQuaternion.x, AccelerationWOutFiltering.y, AccelerationWOutFiltering.z);
```

Here We just call our Functions and send values of the devices own raw values



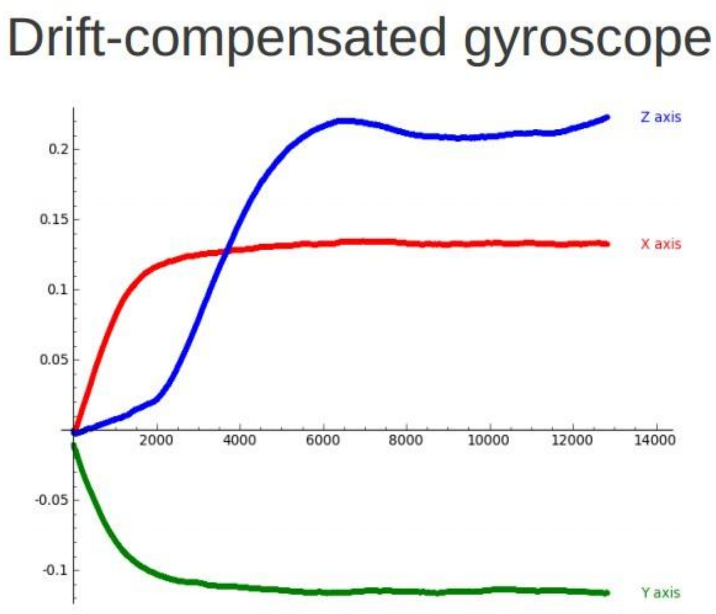

Drift-compensated gyroscope
0.2
0.15
0.1
0.05
-0.05
-0.1
2000
4000
6000
8000
10000
12000
14000
Z axis
X axis
Y axis

FIG : 9

RESULT ANALYSIS AND DISCUSSIONS

Results of Without Low Pass Filtering:

Results Of Filtering Gyro Values

Here we get this much noise for not filtering.

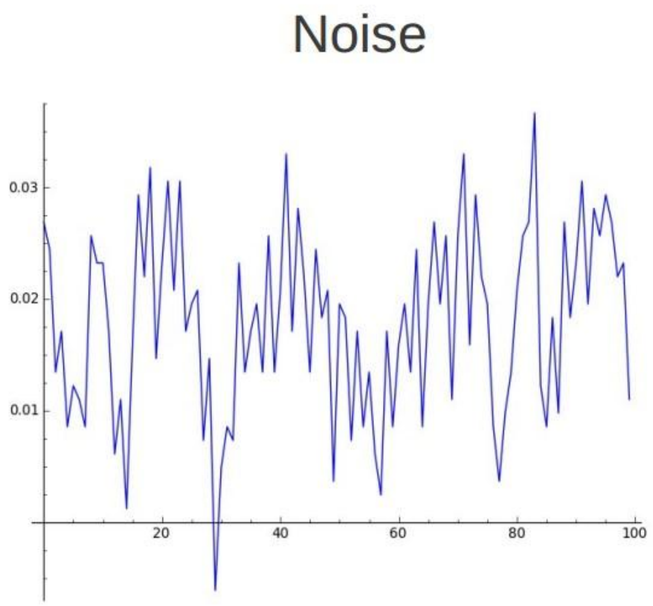


FIG : 10

Here we see that if we don't try to remove noise from the accelerometer then we won't get values as we expected but if we filter the raw values then we can easily remove the noise from acceleration sensor and it will be more powerful and same goes for the gyroscope sensor.

So the overall discussion here we see that if we just low pass filter accelerometer raw values and gyroscopes raw values and compensate both of these sensors and remove the gravity value from accelerometer using the help of gyro sensor we can get a Linear Accelerometer Sensor and it has no drawbacks and work fast.

OUR FUTURE WORK APPROACH

We tried to make a new software based sensor which will have less Limitations and weaknesses. We make a pinch to zoom system using the sensor which is pretty cool and just a couple of lines of code. We make a 3D game which uses gyro with the combination of accelerometer. We also made a soccer football game which uses our software based sensor.

Our Future Plan :

We are planning to make a new types of sensor with the accelerometer data and gyroscope data , We are able to use this fusion sensor for many application. Once we make the sensor working we will work on interactive Whiteboard and motion sensor gaming . Also we are planning to work with construction plan presentation overview . Actually once we get the accurate data as we expect we can use this application in limitless approach .

The possibilities with our research .

I. Activity Monitor and healthcare :

Sensors can be used for activity monitoring with the user . It can keep track of the daily physical activity the user does and suggest him remedy. It can be used as a health care.

II. Gaming control with Mobile :

Gaming must be fun with full body sensor. It can be much pleasure playing with physical movement and motion. We can just use the mobile phone to do all physical sensor game to work

.

III. Interactive Whiteboard for Classroom :

Teachers find lots of difficulty to run the projector slides and gather all the data together in the projector. But we are trying to make a interactive whiteboard with low cost using only mobile sensors and projector

.

IV. Security and Tracking

Mobile sensor can be used in security and tracking. With the sensor data we can analysis it and filter them in different activities and use them in criminal investigation.

Advantages

There is no other option for this kind of software based sensor. Because there is none in the market.

CONCLUSION

So the sensor we are developing is became a new concept on the existing hardware system. With this combined data we can do lots of functionalities and applications. Looking forward to get as much acquire data as possible from those sensors and use them in the best ways .

References

(1) Gabor Paller , gaborpaller@gmail.com **,** Friday, October 7, 2011

(2) http://developer.android.com/guide/topics/sensors/sensors_motion.html#sensors-motion-accel

(3) Kaleb Kircher,2012

(4) Sensor Fusion , Google Tech Talk , MIT , 2009

(5) Kaleb Kircher,2012,About Acceleration

(6) Kaleb Kircher,2012,About Linear Acceleration

(7) http://mylifewithandroid.blogspot.com/2011/10/my-presentation-about-motion.html

(8) http://www.sfonge.com/forum/topic/measuring-movement-accelerometer-and-gyroscope

(9) http://blogs.truthlabs.com/2014/05/23/sensor-input-with-unity3d/

(10) http://docs.unity3d.com/ScriptReference/index.html

(11) http://docs.unity3d.com/Manual/index.html