ระบบตรวจจับความผิดปกติของผู้ป่วยที่มีความผิดปกติทางการเดิน โดยใช้เซนเซอร์ 3 มิติ (KINECT)

(SYSTEM ANALYZE GATE CONDITION OF PATIENT BY SENSOR 3D KINECT)

ทวีศักดิ์ ขจีฟ้า
THAWEESAK KHAJEEFA

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2557

ระบบตรวจจับความผิดปกติของผู้ป่วยที่มีความผิดปกติทางการเดิน โดยใช้เซนเซอร์ 3 มิติ (KINECT)

(SYSTEM ANALYZE GATE CONDITION OF PATIENT BY SENSOR 3D KINECT)

โดย

ทวีศักดิ์ ขจีฟ้า

อาจารย์ที่ปรึกษา

ผศ.ดร. ภัทรพงษ์ ผาสุขกิจ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2557

หัวข้อปริญญานิพนธ์ 2557

สาขา วิศวกรรมอิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบตรวจจับความผิดปกติของผู้ป่วยที่มีความผิดปกติทางการเดินโดยใช้เซนเซอร์ 3 มิติ(Kinect)

ผู้จัดทำ นายทวีศักดิ์ ขจีฟ้า รหัสประจำตัว 54010506

ปริญญานิพนธ์ฉบับนี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

ลงชื่อ.....

(ผศ.ดร.ภัทรพงษ์ ผาสุขกิจ)

อาจารย์ที่ปรึกษา

หัวข้อปริญญานิพนธ์	ระบบตรวจจับความผิดปกติของผู้ป่วยที่มีความผิดปกติทางการเดินโดยใช้เซนเซอร์ 3 มิติ(Kinect)	
นักศึกษา	นายทวีศักดิ์ ขจีฟ้า	รหัสประจำตัว54010506
ปริญญา	วิศวกรรมศาสตร์บัณฑิต	
สาขา	วิศวกรรมอิเล็กทรอนิกส์	
ปีการศึกษา	2557	
อาจารย์ที่ปรึกษา	ผศ.ดร.ภัทรพงษ์ ผาสุขกิจ	

บทคัดย่อ

โครงงานนี้เป็นการนำเสนอการประยุกต์ใช้เซนเซอร์ 3 มิติแบบไม่สัมผัส(Kinect) ในการตรวจจับข้อต่อของโครงสร้างกระดูกตามส่วนต่างๆของร่างกาย เช่น ข้อต่อบริเวณหัวเข่า ข้อต่อบริเวณข้อเท้า เป็นต้น เพื่อนำข้อมูลที่ได้ไปวิเคราะห์ และประมวลผลความผิดปกติของข้อต่อผู้ป่วย โดยใช้โปรแกรม Microsoft Visual studio 2010 (Visual basic) ในการพัฒนา และมีการเชื่อมต่อเก็บข้อมูลและประวัติผู้ป่วยไว้ในฐานข้อมูล SQL Server ซึ่งโครงงานนี้สามารถนำไปประยุกต์ใช้เพื่อทดสอบพัฒนาการทำกายภาพบำบัดและการช่วยการวิเคราะห์วินิจฉัยของแพทย์ในผู้ป่วยที่มีความผิดปกติของข้อต่อในโรงพยาบาล

Thesis Title	System Analyze Gait Condition of Patient by Sensor 3D Kinect
Student	Mr.Thaweesak Khajeefa ID 54010506
Degree	Bachelor of Engineering
Program	Engineering
Year	2014
Thesis Advisor	Asst Prof.Dr.Pattarapong Phasukkit

ABSTRACT

This project, present an application of 3D non-contact sensor Skeleton trackingby Kinect in detecting skeleton tracking such as knee joints and ankle joints, to analyze motion gait of the patient and get all information to be processed an abnormal gaitof the patients using Microsoft Visual studio 2010 (Visual basic) to the development and connected collect patient data in database SQL Server. For this project can be,applied expand to physical therapy and analysis of medical for abnormal gait of patients in hospital.

กิตติกรรมประกาศ

ปริญญานิพนธ์เล่มนี้จะไม่มีทางสำเร็จลุล่วงไปได้เลยหากไม่ได้ความช่วยเหลือและคำแนะนำจากบุคคลหลายๆท่าน โดยเฉพาะอาจารย์ที่ปรึกษาที่คอยให้คำแนะนำต่างๆและคอยช่วยเหลือด้านค่าใช้จ่าย และขอขอบคุณคณะผู้ทำกายภาพบำบัดจากโรงพยาบาลการุญเวชที่คอยให้ความช่วยเหลือทางด้านข้อมูลและการทดสอบโปรแกรมกับผู้ป่วย นอกจากนี้ยังต้องขอขอบคุณผู้ป่วยทุกท่านที่ให้ความร่วมมือในการทดสอบโปรแกรมเป็นอย่างดี และที่สำคัญต้องขอขอบพระคุณบิดา มารดา ที่ให้ทั้งกำลังใจให้ โอกาสทางการศึกษาและการสนับสนุนในทุกๆด้านด้วยดีตลอดมา สุดท้ายขอขอบบุคคลทั้งหมดเหล่านี้อีกครั้งและขออภัยที่ไม่ได้กล่าวรายชื่อบุคคลทั้งหมดไว้ในที่นี้ที่ทำให้ปริญญานิพนธ์เล่มนี้สำเร็จลุล่วงได้อย่างราบลื่น

นายทวีศักดิ์ ขจีฟ้า

สารบัญ

เรื่อง **หน้า**

บทคัดย่อ.....I

Abstract.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญ (ต่อ).....V

สารบัญ(ต่อ).....VI

สารบัญรูปภาพ.....VII

สารบัญรูปภาพ(ต่อ).....VIII

สารบัญตาราง.....IX

สารบัญกราฟ.....X

บทที่ 1 บทนำ

1.1 ความเป็นมาและความสำคัญของโครงงาน.....1

1.2 วัตถุประสงค์ของโครงงาน.....1

1.3ทฤษฎีและแนวคิดที่ใช้ในปริญญานิพนธ์.....1

1.4 ขอบเขตของโครงงาน.....1

1.5 ประโยชน์ที่คาดว่าจะได้รับ.....2

1.6 ขั้นตอนและเวลาในการศึกษาการจัดทำโครงงาน.....2

1.7 โครงสร้างปริญญานิพนธ์.....3

บทที่ 2 ทฤษฎีและหลักการ

2.1ทฤษฎีเกี่ยวกับการวัดมุม.....4

2.2 เครื่องวัดมุมโกนิโอมิเตอร์.....7

2.3 ทฤษฎีเวกเตอร์.....7

2.3.1 การรวมเวกเตอร์.....7

2.3.2 เวกเตอร์หนึ่งหน่วย (Unit Vector).....9

2.3.3 เวกเตอร์องค์ประกอบ (Component Vector).....10

2.3.4 เวกเตอร์ตำแหน่ง (Position Vector).....12

สารบัญ (ต่อ)

เรื่อง หน้า

2.4Kinect....13
2.4.1 ส่วนประกอบของ Kinect....13
2.4.2 ระบบต่างๆที่ Kinect รองรับ....13
2.4.3 หลักการทำงานของระบบตรวจวัดระยะ....15
2.4.4 Kinect สำหรับ Windows....17
2.5 โปรแกรม Microsoft Visual Studio 2010....18
2.6 ภาษา Visual Basic....19
2.6.1 ประวัติความเป็นมาของภาษา Visual Basic....19
2.6.2 Visual Basic....20
2.6.3 ส่วนประกอบของ Visual Basic....20
2.6.4 หลักในการเขียนโปรแกรมใน Visual Basic....20
2.6.5 การออกแบบหน้าจอของโปรแกรมด้วยคอนโทรล....20
2.6.6 WPF Application....21

บทที่ 3 หลักการทำงาน

3.1 หลักการ skeleton tacking....22
3.2การทำงานของระบบ....23
3.3 ลักษณะท่าทางในการทดสอบ....26

บทที่ 4 ผลการทดลอง

4.1การทดลองวัดค่ามุมของคนปกติ....28
4.1.1 วิธีการทดลองวัดค่ามุมของคนปกติ....28
4.1.2 ผลการทดลองวัดค่าคนปกติ....30
4.2การทดลองวัดค่ามุมของระบบในคนปกติเทียบกับเครื่องมือวัดโกนิโอมิเตอร์ (ก่อนทำการ Calibrate)....31
4.2.1วิธีการทดลองวัดค่ามุมของระบบในคนปกติเทียบกับเครื่องมือวัดโกนิโอมิเตอร์ ก่อนทำการ Calibrate....31
4.2.2 ผลการทดลองวัดค่ามุมของระบบในคนปกติเทียบกับเครื่องมือวัดโกนิโอมิเตอร์ ก่อนทำการ Calibrate....33

สารบัญ (ต่อ)

เรื่อง หน้า

4.3การทดลองวัดค่ามุมของคนปกติเทียบกับเครื่องมือวัดโกนิโอมิเตอร์ (หลังทำการ Calibrate).....34
4.3.1 วิธีการทดลองวัดค่ามุมของคนปกติเทียบกับเครื่องมือวัดโกนิโอมิเตอร์ หลังทำการ Calibrate.....34
4.3.2 ผลการทดลองวัดค่ามุมของระบบในคนปกติเทียบกับเครื่องมือวัดโกนิโอมิเตอร์ หลังทำการ Calibrate.....38
4.4 การทดลองวัดค่ามุมของผู้ป่วยที่ทำการกายภาพบำบัด.....38
4.4.1 ผู้ป่วยที่ปวดสะโพก.....38
4.4.2 ผู้ป่วยที่มีอาการเจ็บหลัง.....39
4.4.3 ผู้ป่วยที่ทำการผ่าตัดหัวเข่าซ้าย.....39
4.5 การทดลองผลการกายภาพบำบัด.....42

บทที่ 5 สรุปและวิจารณ์ผลการทดลอง
5.1 สรุปผลการทดลอง.....43
5.2ปัญหาและอุปสรรค.....44
5.3ข้อเสนอแนะ.....44

บรรณานุกรม.....45
ภาคผนวก ก. Code program
ภาคผนวก ข.รูปภาพการทำกายภาพบำบัด

สารบัญรูปภาพ

รูปที่ หน้า

2.1 มุมมาตรฐานของหัวไหล่และลักษณะท่าทาง....5
2.2 มุมมาตรฐานของข้อศอกและลักษณะท่าทาง....5
2.3 มุมมาตรฐานของสะโพกและลักษณะท่าทาง....6
2.4 มุมมาตรฐานของหัวเข่าและลักษณะท่าทาง....6
2.5 โกนิโอมิเตอร์ (Goniometer)....7
2.6 เวกเตอร์ $\vec{R} = \vec{P} + \vec{Q}$....8
2.7 เวกเตอร์ลัพธ์ R....8
2.8 การลบเวกเตอร์....9
2.9 เวกเตอร์ 3 มิติ....10
2.10 องค์ประกอบของมุมเวกเตอร์ 3 มิติ....11
2.11 เวกเตอร์บอกตำแหน่งแบบ 3 มิติ....12
2.12ประกอบต่างๆ ของ Kinect....13
2.13 ภาพแสดงการทำงานของระบบตรวจวัดระยะ....15
2.14 (ก)การจับจุด Skeletal Tacking....16
(ข) จุด Skeletal Tacking....16
2.16ระยะที่ใช้งานของ Kinect....18
2.16โปรแกรม Microsoft Visual Studio 2010....19
3.1 จุด Skeleton Tacking....22
3.2 ตำแหน่งที่แสดงค่ามุม ค่าแกนตั้งของแต่ละข้อ....23
3.2 หน้าต่างโปรแกรมโหมดการทำงานที่ 1 แบบเก็บข้อมูล....23
3.4 ภาพการเก็บค่าลงฐานข้อมูล SQL....24
3.5 Flow Chart การทำงานโหมดที่ 1....26
3.6 การทดสอบวัดค่ามุมของข้อต่อหัวไหล่....26
3.7 การทดสอบวัดค่ามุมของข้อศอก....26
3.8 การทดสอบวัดค่ามุมของข้อเข่า....27
4.1 ท่ายืนตรงแขนแนบลำตัว....28
4.2 ท่ายืนตรงกางแขนออกไปข้างลำตัว....28
4.3 ท่ายืนตรงกางแขนแล้วหักข้อศอกทำมุมประมาณ 90 องศา....29
4.4 ท่ายืนตรงยกแขนทั้งสองข้าง....30
4.5 (ก) กาวัดมุมด้วยระบบในท่ายืนตรง....31
(ข) การวัดมุมด้วยเครื่องมือวัดในท่ายืนตรง....31
4.6 (ก) การวัดมุมด้วยระบบท่ายืนตรงกางแขนออกไปข้างลำตัว....32
(ข) การวัดมุมด้วยเครื่องมือวัดท่ายืนตรงกางแขนออกไปข้างลำตัว....32

สารบัญรูปภาพ

รูปที่ หน้า

4.7 (ก) การวัดมุมด้วยระบบในท่ายืนตรงกางแขนแล้วหักข้อศอกทำมุมประมาณ 90 องศา....32
(ข) การวัดมุมด้วยเครื่องมือวัดยืนตรงกางแขนแล้วหักข้อศอกทำมุมประมาณ 90 องศา....32
4.8 (ก) การวัดมุมด้วยระบบในท่ายืนตรงยกแขนทั้งสองข้าง....33
(ข) การวัดมุมด้วยเครื่องมือวัดในท่ายืนตรงยกแขนทั้งสองข้าง....33
4.9 (ก) การวัดมุมหัวเข่าซ้ายด้วยระบบ....34
(ข) การวัดมุมเข่าขวาด้วยโกนิโอมิเตอร์....34
4.10 (ก) การวัดมุมหัวเข่าซ้ายด้วยระบบ....35
(ข) การวัดมุมเข่าขวาด้วยโกนิโอมิเตอร์....35
4.11 (ก) การวัดมุมที่ข้อศอกซ้ายด้วยระบบ....35
(ข) การวัดมุมที่ข้อศอกซ้ายด้วยโกนิโอมิเตอร์....35
4.12 (ก) การวัดมุมที่ข้อศอกขวาด้วยระบบ....36
(ข) การวัดมุมที่ข้อศอกขวาด้วยโกนิมิเตอร์....36
4.13 (ก) การวัดมุมที่หัวไหลซ้ายด้วยระบบ....36
(ข) การวัดมุมที่หัวไหลซ้ายด้วยโกนิโอมิเตอร์....36
4.14 (ก) การวัดมุมที่หัวไหล่ขวาด้วยระบบ....37
(ข) การวัดมุมที่หัวไหล่ขวาด้วยโกนิโอมิเตอร์....37
4.15 การวัดมุมที่สะโพกซ้าย – ขวา....37
4.16 (ก) ภาพจำลองของ skeletons ผู้ป่วยปวดสะโพก....38
(ข)ภาพแสดงท่าทางที่ผู้ป่วยปวดสะโพกทำการทดสอบ....38
4.17 (ก) ภาพจำลองของ skeletons ผู้ป่วยปวดหลัง....39
(ข) ภาพแสดงท่าทางที่ผู้ป่วยปวดหลังทำการทดสอบ....39
4.18 (ก) ภาพจำลองของ skeletons ผู้ป่วยผ่าตัดหัวเข่า (ขวา)....40
(ข) ภาพแสดงท่าทางที่ผู้ป่วยผ่าตัดหัวเข้าทำการทดสอบ....40
4.19 (ก) ภาพจำลองของ skeletons ผู้ป่วยผ่าตัดหัวเข่า (ซ้าย)....41
(ข) ภาพแสดงท่าทางที่ผู้ป่วยผ่าตัดหัวเข้าทำการทดสอบ....41

สารบัญตาราง

ตารางที่ **หน้า**

1.1 ขั้นตอนและระยะเวลาในการดำเนินงานของภาคเรียนที่ 1/2557.....2
1.2 ขั้นตอนและระยะเวลาในการดำเนินงานของภาคเรียนที่ 2/2557.....2
2.1 ตารางมาตรฐานของมุมข้อต่อต่างๆ.....4
2.2 ตารางแสดงรายละเอียดของ Kinect.....14
4.1 แสดงผลการทดลองวัดค่ามุมของคนปกติครั้งที่ 1.....30
4.2 แสดงผลการทดลองวัดค่ามุมของคนปกติครั้งที่ 2.....30
4.3 แสดงผลการทดลองวัดค่ามุมของคนปกติครั้งที่ 3.....31
4.4 ผลการทดลองวัดค่ามุมของระบบในคนปกติเทียบกับเครื่องมือวัดโกนิโอมิเตอร์ (ก่อนการ Calibrate).....32
4.5 ผลการทดลองวัดค่ามุมของระบบในคนปกติเทียบกับเครื่องมือวัดโกนิโอมิเตอร์ (หลังการ Calibrate).....38
4.6 ค่ามุมที่ทำการกายภาพบำบัด.....42

สารบัญกราฟ

กราฟที่ **หน้า**

4.1 ค่าแกนตั้งของสะโพกข้างซ้าย และขวา (ผู้ป่วยปวดสะโพก).....39
4.2ค่าแกนตั้งของสะโพกข้างซ้าย และขวา (ผู้ป่วยปวดหลัง).....40
4.3ค่ามุมของเข่าข้างซ้าย และขวา (ผู้ป่วยที่ผ่าตัดหัวเข่า).....41

บทที่ 1
บทนำ

ในบทนี้จะกล่าวถึงความเป็นมาและความสำคัญของปัญหา สาระสำคัญที่เป็นปัจจัยหลัก รวมทั้งวัตถุประสงค์ของการศึกษา ทฤษฎีและแนวความคิดที่ใช้ในการทำปริญญานิพนธ์ ขอบเขตของการศึกษา ประโยชน์ที่คาดว่าจะได้รับ ขั้นตอนและเวลาในการจัดทำโครงงาน รวมทั้งโครงสร้างปริญญานิพนธ์ฉบับนี้

1.1 ความเป็นมาและความสำคัญของโครงงาน

โครงงานวิจัยได้แสดงให้เห็นถึงความสำคัญการวัดพารามิเตอร์ของมุมและลักษณะข้อต่อของผู้ป่วยที่มีความผิดปกติทางการเดิน ซึ่งปัจจุบันทางการกายภาพบำบัดผู้ป่วยได้มีการวัดพารามิเตอร์ของค่ามุมโดยใช้เครื่องมือโกนิโอมิเตอร์ในการวัด เพื่อความสะดวกในการวิเคราะห์ของแพทย์และนักกายภาพบำบัด จึงได้มีการออกแบบระบบมีความสามารถในการวัดพารามิเตอร์ของมุมข้อต่อต่างๆ นอกจากนี้การศึกษาได้ชี้ให้เห็นว่าการเปลี่ยนของพารามิเตอร์ได้สอดคล้องกับการเดิน และอาจจะไม่สอดคล้องเนื่องจากการเดินที่ผิดปกติ โดยมีแพทย์ผู้เชี่ยวชาญเข้ารวมวิเคราะห์ข้อมูล ซึ่งระบบสามารถวัดค่ามุมของข้อต่อ เช่น ข้อต่อที่หัวเข่า สะโพก และข้อเท้ายังมีค่า Vertical ที่สะโพก ระบบสามารถให้ข้อมูลที่สามารถนำไปวิเคราะห์ความผิดปกติในการเดิน หรือ แยกแยะการเดินที่ผิดปกติออกจากคนปกติ ซึ่งสีผิวจะไม่ผลต่อการวัดค่าพารามิเตอร์จะไม่ติดมาร์คเกอร์ไว้ตามร่างกาย หรือเป็นการสร้างความรำคาญให้ผู้ป่วยและช่วยในการวินิจฉัยทางการด้านแพทย์

1.2วัตถุประสงค์ของโครงงาน

1. เพื่อใช้วิเคราะห์พัฒนาการของผู้ผิดปกติทางการเดิน
2. เพื่อช่วยในการวิเคราะห์การรักษาของแพทย์ให้มีความรวดเร็วและง่ายขึ้น

1.3 ทฤษฎีและแนวคิดที่ใช้ในปริญญานิพนธ์

ในปัจจุบันการวัดค่ามุมตามมาตรฐานที่โรงพยาบาลใช้วัดผู้ป่วยที่มีความผิดปกติทางการเดิน โดยมีการวัดด้วยเครื่องมือวัดโกนิโอมิเตอร์และท่าลักษณะการวัด ในจุดข้อต่อต่างๆ โดยทำการวัดในลักษณะท่าทางต่างๆ เช่น การวัดค่ามุมที่หัวไหล่จะวัดในท่าลักษณะยกแขนและเหยียดแขนลงไปด้านหลัง การวัดค่ามุมที่ศอกในท่ากางแขนเหยียดและพับแขนเข้า และการวัดค่ามุมที่ข้อเข่าในท่ายืนตรงและงอเข่าขึ้น เป็นต้น แพทย์จะนำค่ามุมที่ได้ไปวิเคราะห์ถึงลักษณะความผิดปกติเพื่อไปสู่การรักษาและการกายภาพบำบัดต่อไป

1.4ขอบเขตของโครงงาน

1. สามารถตรวจและแยกแยะระหว่างคนปกติกับผู้ที่มีความผิดปกติทางการเดินได้
2. สามารถบอกได้ว่าผู้ป่วยที่เข้ารับการทดสอบมีความผิดปกติทางด้านใด

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถนำระบบไปช่วยในการวิเคราะห์ความผิดปกทางการเดิน
2. สามารถช่วยให้ผู้ป่วยเชื่อหมั่นในการรักษา
3. ช่วยในการตรวจสอบทางกายภาพของผู้ป่วย

1.6 ขั้นตอนและเวลาในการศึกษาการจัดทำโครงงาน

ในการจัดทำโครงงาน ผู้จัดทำได้ออกแบบขั้นตอนในการศึกษาและระยะเวลาในการจัดทำโครงงานตั้งแต่การศึกษาค้นคว้าข้อมูล การออกแบบโครงสร้าง การออกแบบวงจรและการทดลองใช้งานได้จริง เพื่อให้ได้สำเร็จตามวัตถุประสงค์ที่ได้วางไว้ ซึ่งในการปฏิบัติงานดังตารางที่ 1.1 และ 1.2

ตารางที่ 1.1ขั้นตอนและระยะเวลาในการดำเนินงานของภาคเรียนที่ 1/2557

1/2557					
ขั้นตอนการดำเนินงาน	ระยะเวลา				
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.
1. ศึกษาข้อมูลที่เกี่ยวข้องกับโครงงาน	←	→			
2. จัดหาและศึกษาการทำงานเซนเซอร์		←	→		
3. ออกแบบโปรแกรม		←	—	→	
4. ทดสอบการทำงานของโปรแกรม				←	→
5. แก้ไขปรับปรุง					← →

ตารางที่ 1.2ขั้นตอนและระยะเวลาในการดำเนินงานของภาคเรียนที่ 2/2557

2/2557					
ขั้นตอนการดำเนินงาน	ระยะเวลา				
	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.
1. แก้ไขโปรแกรมและเพิ่มการเก็บข้อมูล	←	→			
2. ทดสอบการทำงานของโปรแกรม		←	→		
3. ทดสอบกับผู้ป่วยในโรงพยาบาล			←	→	

1.7 โครงสร้างปริญญานิพนธ์

ระบบตรวจจับความผิดปกติของผู้ป่วยที่มีความผิดปกติทางการเดินโดยใช้เซนเซอร์ 3 มิติ (Kinect) ได้อธิบายถึงขั้นตอนและวิธีการออกแบบ โดยส่วนเนื้อหาของบทต่างๆ ดังต่อไปนี้

บทที่ 1 บทนำ ในบทนี้จะกล่าวถึง ความเป็นมาและความสำคัญวัตถุประสงค์ของการศึกษา แนวความคิดที่ใช้ในการทำปริญญานิพนธ์ ขอบเขตของการศึกษา ประโยชน์ที่คาดว่าจะได้รับ ขั้นตอนและเวลาในการจัดทำโครงงาน รวมทั้งโครงสร้างปริญญานิพนธ์ เป็นต้น

บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง ในบทนี้จะกล่าวถึง ความรู้ที่เกี่ยวข้องกับการวัดมุมบริเวณต่างๆ เครื่องวัดมุมโกนิโอมิเตอร์ การคำนวณเวกเตอร์ เซนเซอร์ตรวจจับการเคลื่อนไหวแบบ 3 มิติ หรือ Kinect โปรแกรม Microsoft Visual Studio 2010เป็นต้น

บทที่ 3 การออกแบบระบบ ในบทนี้จะกล่าวถึงการออกแบบการทำงานของระบบว่ามีหลักการอย่างไรและมีขั้นตอนวิธีการทำงานของโปรแกรม แบ่งเป็นการอธิบายหลักการจับ Skeleton tacking ของกล้อง Kinect ว่ามีการจับส่วนใดอย่างไร และอธิบายขั้นตอนรูปแบบการทำงานของ 2 โหมดการทำงาน ได้แก่ โหมดแบบเก็บค่าและโหมดช่วยกายภาพ ออกเป็นแผนภาพ Flow Chart และท่าทางที่ใช้ในการทดสอบ

บทที่ 4 วิธีการและผลการทดลอง ในบทนี้จะกล่าวถึง วิธีการและผลการทดลองที่นำไปทดสอบตัวอย่างผู้ป่วยที่มีความผิดปกติทางการเดินในรูปแบบตารางและกราฟ และการวิเคราะห์ผลที่ได้ว่าผู้ป่วยมีความผิดปกติอย่างไรตรงจุดใด

บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ ในบทนี้จะกล่าวถึง ผลสรุปภาพโดยรวมของการทดสอบ การสรุปผลการทดสอบของตัวอย่างผู้ป่วยในลักษณะอาการที่ต่างกัน และวิเคราะห์ที่ได้จากบทที่ 4

บทที่ 2
ทฤษฎีและหลักการ

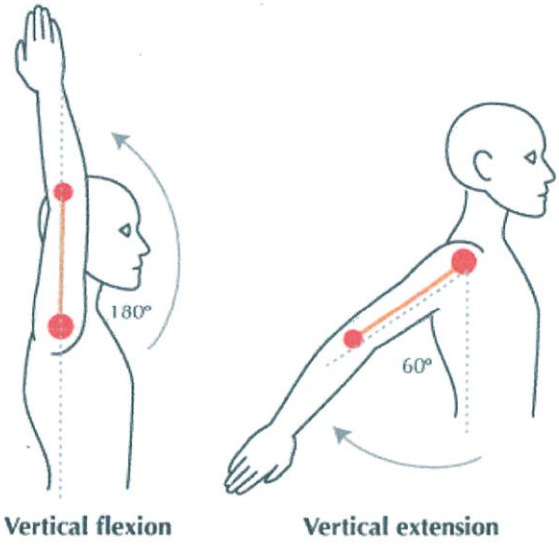
ในบทนี้จะกล่าวถึง การวัดมุมที่บริเวณข้อต่อ ซึ่งหากผู้ที่มีความผิดปกติทางการเดินจะมีมุมที่แต่ต่างจากมุมมาตรฐานในแต่ละอริยาบาท และเครื่องมือที่ใช้วัดมุมคือโกนิโอมิเตอร์ใช้วัดมุมบริเวณข้อต่อของผู้ป่วย โดยระบบจะใช้เซนเซอร์ชนิดไม่สัมผัส ซึ่งใช้สูตรการคำนวณแบบเวกเตอร์บนโปรแกรมMicrosoft Visual Studio 2010 ด้วยภาษา Visual Studio

2.1 ทฤษฎีเกี่ยวกับการวัดมุม

ตารางที่ 2.1 ตารางมาตรฐานของมุมข้อต่อต่างๆ

Joint	Motion	Range (°)
Shoulder	Flexion	180
	Extension	60
Elbow	Flexion	150
	Extension	0
Hip	Flexion	125
	Extension	10
Knee	Flexion	130
	Extension	0

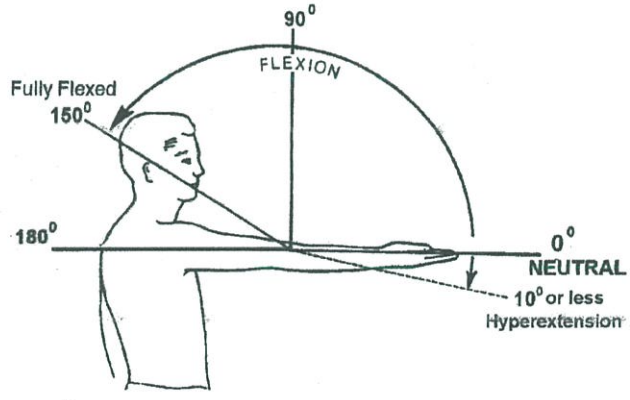
ที่มา : http://apamedcentral.org/search.php?where=aview&id=10.0000%2Famp.2009.43.1.16&code=0012AMP&vmode=AFTR



รูปที่ 2.1 มุมมาตรฐานของหัวไหล่และลักษณะท่าทาง

ที่มา :**http://design.tutsplus.com**/articles/human-anatomy-fundamentals-flexibility-and-%20joint-Limitations--vector-25401

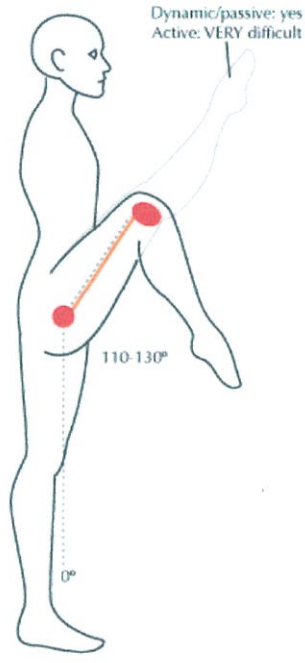
จากรูปที่ 2.1 แสดงค่ามุมมาตรฐานของหัวไหล่ของคนที่ปกติในลักษณะชูแขนแนวดิ่งและเหยียดตรงไปด้านหลังตามปกติค่ามุมของหัวไหล่ในลักษณะการชูแขนตรงในแนวดิ่งจะสามารถกางออกจากแกนอ้างอิงได้มากสุด 180 องศา ส่วนมุมของลักษณะเหยียดแขนตรงไปด้านหลังออกจาแกนอ้างอิงได้มากสุด 60 องศา



รูปที่ 2.2 มุมมาตรฐานของข้อศอกและลักษณะท่าทาง

ที่มา :http://www-ieem.ust.hk/dfaculty/ravi/cricket/cric_pics.html

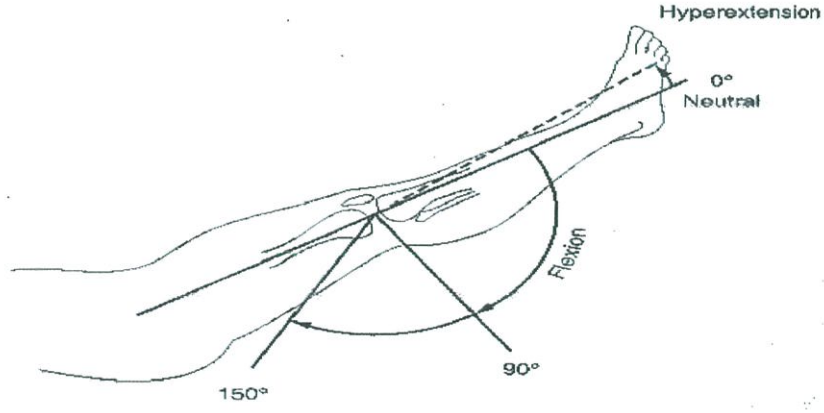
จากรูปที่ 2.2 แสดงค่ามุมมาตรฐานของข้อศอกของคนที่ปกติในลักษณะการกางแขนเหยียดตรงและงอพับแขนเข้าหาตัว ตามปกติค่ามุมของข้อศอกลักษณะการกางแขนเหยียดตรงจะสามารถกางออกจากแกนอ้างอิงได้มากสุด 0 องศา หรือมีค่าการกางออกเกินได้ 10 องศา ส่วนค่ามุมของลักษณะการงอแขนเข้าหาตัวได้ 90 องศา และงอได้มากสุด 150 องศา



รูปที่ 2.3มุมมาตรฐานของสะโพกและลักษณะท่าทาง

ที่มา :**http://design.tutsplus.com**/articles/human-anatomy-fundamentals-flexibility-and-%20joint-Limitations--vector-25401

จากรูปที่ 2.3 แสดงค่ามุมมาตรฐานของสะโพกของคนที่ปกติในลักษณะการยกขาไปด้านหน้าและเหยียดขาลง ตามปกติค่ามุมของสะโพกลักษณะยกขาไปด้านหน้าออกจากแกนอ้างอิงได้มากสุด 130 องศา ส่วนค่ามุมของสะโพกลักษณะเหยียดขาลงได้ 0 องศา

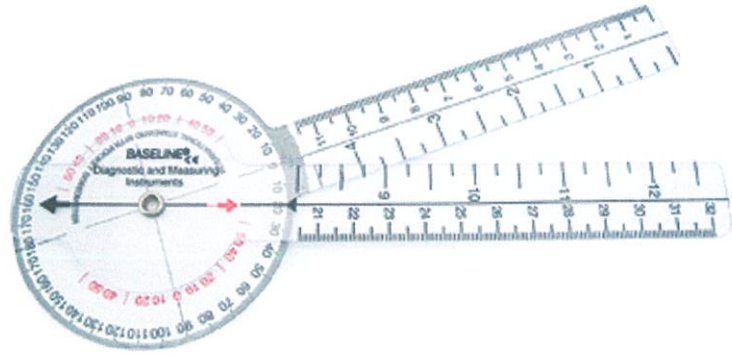


รูปที่ 2.4 มุมมาตรฐานของหัวเข่าและลักษณะท่าทาง

ที่มา :**http://design.tutsplus.com**/articles/human-anatomy-fundamentals-flexibility-and-%20joint-Limitations--vector-25401

จากรูปที่ 2.4 แสดงค่ามุมมาตรฐานของหัวเข่าของคนที่ปกติในลักษณะการงอหัวเข่าและเหยียดตรง ตามปกติค่ามุมของหัวเข่าลักษณะงอหัวเข่าออกจากแกนอ้างอิงได้ 90 องศาและมากสุด 150 องศา ส่วนค่ามุมของหัวเข่าลักษณะเหยียดตรงได้ 0 องศา

2.2 เครื่องวัดมุมโกนิโอมิเตอร์



รูปที่ 2.5 โกนิโอมิเตอร์ (Goniometer)

ที่มา : http://www.algeos.com/kilmartin-digital-goniometer.html

เครื่องมือวัดมุมโกนิโอมิเตอร์ (Goniometer) เป็นเครื่องมือที่ทั้งวัดมุมหรือช่วยให้วัตถุที่จะหมุนไปยังตำแหน่งเชิงมุมที่แม่นยำ goniometers นี้จะสามารถวัดช่วงของการเคลื่อนไหวรอบข้อต่อของส่วนต่างๆของร่างกาย มีวิธีการใช้ 4 ขั้นตอนดังนี้

1.วาง goniometer กว่าศูนย์กลางของข้อต่อ วาง goniometer ตามแนวเคลื่อนที่ของร่างกายหรือจุดแกนอ้างอิงของการวัดมุมนั้นๆ

2. ตรวจสอบดูการวางของ goniometer ให้แน่ใจว่าอยู่ในแกนกลางหรือแกนอ้างอิง

3. ผู้ที่วัดต้องอยู่นิ่งๆจนกว่าจะทำการวัดเสร็จ

4. ดูและอ่านค่ามุมที่วัดจากเครื่องวัด goniometer

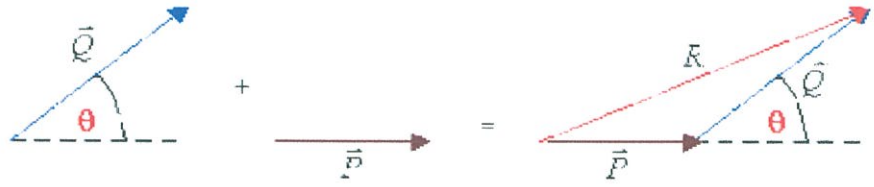
2.3 ทฤษฎีเวกเตอร์

เวกเตอร์ (Vector)คือ ปริมาณที่มีทั้งขนาดและทิศทางเช่น ความเร็ว , ความเร่ง ,การกระจัด , แรง ฯลฯ

2.3.1การรวมเวกเตอร์

การรวมเวกเตอร์ หมายถึง การบวกหรือลบกันของเวกเตอร์ตั้งแต่ 2 เวกเตอร์ ขึ้นไป ผลลัพธ์ที่ได้เป็นปริมาณเวกเตอร์ เรียกว่า เวกเตอร์ลัพธ์ (Resultant Vector) ซึ่งพิจารณาได้ ดังนี้

1. การบวกเวกเตอร์โดยวิธีการเขียนรูปทำได้โดยเขียนเวกเตอร์ที่เป็นตัวตั้งจากนั้นเอาหางของเวกเตอร์ที่เป็นผลบวกหรือผลต่าง มาต่อกับหัวของเวกเตอร์ตัวตั้งโดยเขียนให้ถูกต้องทั้งขนาดและทิศทาง เวกเตอร์ลัพธ์หาได้โดยการวัดระยะทาง จากหางเวกเตอร์แรกไปยังหัวเวกเตอร์สุดท้าย

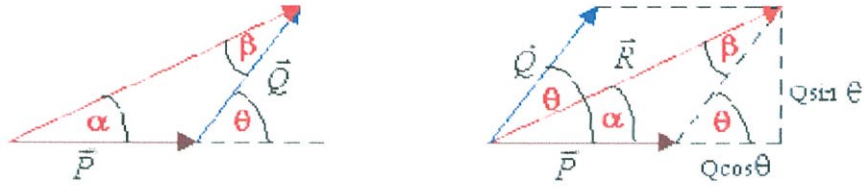


รูปที่ 2.6 เวกเตอร์ $\vec{R} = \vec{P} + \vec{Q}$

ที่มา : http://www.rsu.ac.th/science /physics/kan/general_phy/vector/vector.htm

2. การบวกเวกเตอร์โดยใช้วิธีการทางคณิตศาสตร์

ให้ เวกเตอร์$\vec{P}$ ทำมุมกับ$\vec{Q}$ เป็นมุม q คำนวณหาเวกเตอร์ลัพธ์ได้ ดังนี้



รูปที่ 2.7 เวกเตอร์ลัพธ์ R

ที่มา : http://www.rsu.ac.th/science /physics/kan/general_phy/vector/vector.htm

ขนาดของเวกเตอร์ลัพธ์คำนวณได้จากกฎของโคไซน์

$$R = \sqrt{P^2+Q^2+PQ\ COS\ \theta}(1)$$

ทิศทางของเวกเตอร์ลัพธ์หาได้จาก

$$a = \tan^{-1}\left(\frac{Q\sin\theta}{P+Q\cos\theta}\right) \qquad (2)$$

หรือหาได้จากกฎของไซน์ ดังนี้

$$\frac{P}{\sin\beta} = \frac{Q}{\sin\alpha} = \frac{R}{\sin(180-\theta)} \qquad (3)$$

ข้อสังเกต จากสมการที่ (1) พบว่า

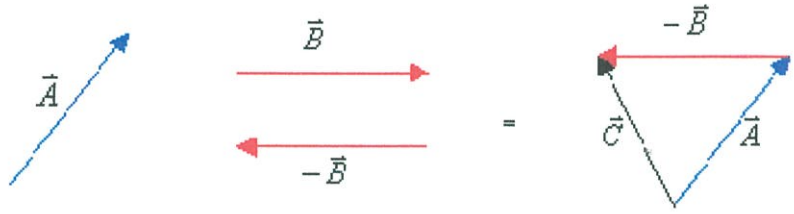
1. เมื่อ q = 0°(คือ$\vec{P}$ และ$\vec{Q}$ อยู่ในทิศทางเดียวกัน)จะได้ขนาดของ$\vec{R}$ = $\vec{P}$+ $\vec{Q}$ โดยทิศทางของ$\vec{R}$ มีทิศเดียวกับ$\vec{P}$ และ$\vec{Q}$
2. เมื่อ q = 180°
   2.1 ถ้า$\vec{P}$>$\vec{Q}$จะได้$\vec{R}$ = $\vec{P}$ - $\vec{Q}$และ$\vec{R}$มีทิศเดียวกับ$\vec{P}$
   2.2 ถ้า$\vec{P}$<$\vec{Q}$ จะได้$\vec{R}$= $\vec{Q}$ - $\vec{P}$และ$\vec{R}$มีทิศเดียวกับ$\vec{Q}$
3. เมื่อ q = 90 ° จะได้

ขนาด R $\sqrt{P^2+Q^2}$และ a = $\tan^{-1}(\frac{P}{Q})$

3. การลบเวกเตอร์

การลบเวกเตอร์ สามารถหาเวกเตอร์ลัพธ์ได้เช่นเดียวกับการบวกเวกเตอร์แต่ให้กลับทิศทางของเวกเตอร์ตัวลบ ดังนี้

ถ้า $\vec{C} = \vec{A} - \vec{B} = \vec{A} + (-\vec{B})$



รูปที่ 2.8 การลบเวกเตอร์

ที่มา : http://www.rsu.ac.th/science /physics/kan/general_phy/vector/vector.htm

2.3.2 เวกเตอร์หนึ่งหน่วย (Unit Vector)

เวกเตอร์หนึ่งหน่วย หมายถึง เวกเตอร์ที่มีขนาดหนึ่งหน่วยในทิศทางใดๆเช่น เวกเตอร์$\vec{A}$ สามารถเขียนได้ด้วยขนาดของ$\vec{A}$คูณกับเวกเตอร์หนึ่งหน่วย$\overrightarrow{e_A}$ซึ่งมีทิศทางเดียวกับ$\vec{A}$คือ

$$\vec{A}=A\overrightarrow{e_A} \qquad (4)$$

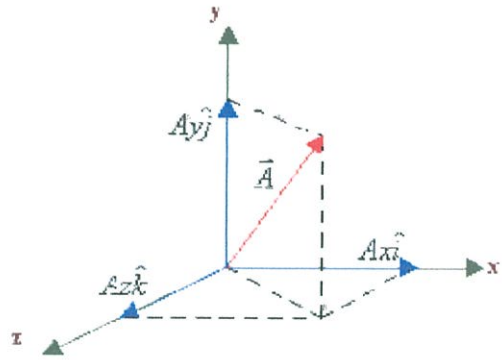
โดย $\vec{e_A}$ คือเวกเตอร์หนึ่งหน่วยที่มีขนาดหนึ่งหน่วยและทิศเดียวกันกับ $\vec{A}$

ในระบบแกนมุมฉาก เวกเตอร์หนึ่งหน่วยบนแกน x , y และ z แทนด้วยสัญลักษณ์ $\hat{i}$, $\hat{j}$ และ $\hat{k}$ ตามลำดับ จะได้

$$\hat{i}=\frac{\vec{A}_x}{A_x};\ \hat{j}=\frac{\vec{A}_y}{A_{xy}};\ \hat{k}=\frac{\vec{A}_z}{A_{zx}}$$

เมื่อ

$\vec{A}_x$ คือเวกเตอร์ที่มีขนาดเท่ากับ $\vec{A}_x$ มีทิศทางตามแนวแกน x
$\vec{A}_y$ คือเวกเตอร์ที่มีขนาดเท่ากับ $\vec{A}_y$ มีทิศทางตามแนวแกน y
$\vec{A}_z$ คือเวกเตอร์ที่มีขนาดเท่ากับ $\vec{A}_z$ มีทิศทางตามแนวแกน z



รูปที่ 2.9 เวกเตอร์ 3 มิติ
ที่มา : http://www.rsu.ac.th/science /physics/kan/general_phy/vector/vector.htm

2.3.3เวกเตอร์องค์ประกอบ (Component Vector)

1. องค์ประกอบของเวกเตอร์ใน 2 มิติ

ถ้า $\vec{A}$ อยู่ในระนาบ x , y โดย $\vec{A}$ ทำมุม q กับแกน x
องค์ประกอบของ $\vec{A}$ ตามแกน x คือ $\vec{A}_x$ โดย $\vec{A}_x$ = Acosq
องค์ประกอบของ $\vec{A}$ ตามแกน y คือ $\vec{A}_y$ โดย $\vec{A}_y$ = Asinq
ดังนั้น เวกเตอร์ $\vec{A}$ เขียนแยกเป็นองค์ประกอบได้ดังนี้

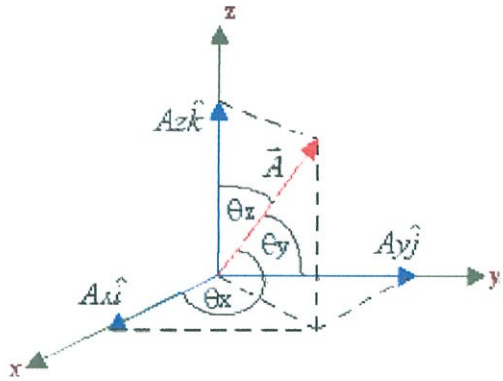
$$\vec{A}=\vec{A}_x\hat{i}+\vec{A}_y\hat{j} \quad \text{หรือ} \quad \vec{A}=A\cos q\,\hat{i}+A\sin q\,\hat{j} \qquad (5)$$

โดยที่ ขนาดของ $\vec{A}$

$$\vec{A}=\sqrt{\vec{A}_x^{\,2}+\vec{A}_y^{\,2}} \qquad (6)$$

2. องค์ประกอบของเวกเตอร์ใน 3 มิติ

กำหนดให้$\vec{A}$อยู่บนระนาบ x , y ,z โดยเวกเตอร์ $\vec{A}$ ทำมุมกับแกน x , y , z เป็นมุม q_x , q_y , q_zตามลำดับ เวกเตอร์$\vec{A}$สามารถแยกเป็นองค์ประกอบตามแกน x , y , z ได้ ดังนี้



รูปที่ 2.10 องค์ประกอบของมุมเวกเตอร์ 3 มิติ

ที่มา : http://www.rsu.ac.th/science /physics/kan/general_phy/vector/vector.htm

ขนาดของ $\vec{A}_x$แทนด้วย $A_x = A\cos q_x$โดยที่ $\cos q_x = \frac{\vec{A}_x}{A}$

ขนาดของ $\vec{A}_y$แทนด้วย $A_y = A\cos q_y$โดยที่ $\cos q_y = \frac{\vec{A}_y}{A}$

ขนาดของ $\vec{A}_z$แทนด้วย $A_z = A\cos q_z$โดยที่ $\cos q_z = \frac{\vec{A}_z}{A}$

ดังนั้น $\vec{A} = \vec{A}_x + \vec{A}_y + \vec{A}_z$

$$\vec{A} = A\cos\theta_x\hat{i} + A\cos\theta_y\hat{j} + A\cos\theta_z\hat{k} \qquad (7)$$

ขนาด$\vec{A}$คือ

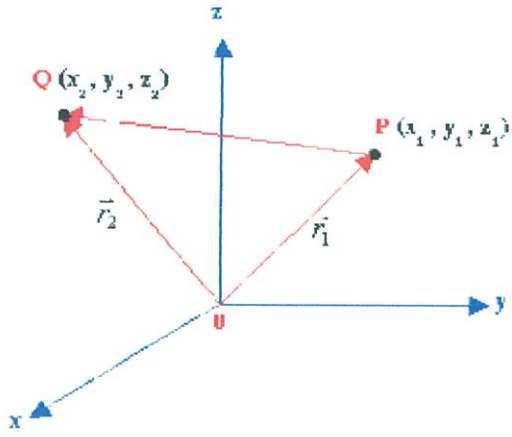
$$A = \sqrt{\vec{A}_x^{\,2} + \vec{A}_y^{\,2} + \vec{A}_z^{\,2}} \qquad (8)$$

ทิศทางของเวกเตอร์$\vec{A}$คือมุมที่$\vec{A}$ทำกับแกน x , y , z หาได้จาก

$$\theta_x = \cos^{-1}\left(\frac{\vec{A}_x}{A}\right), \theta_y = \cos^{-1}\left(\frac{\vec{A}_y}{A}\right), \theta_z = \cos^{-1}\left(\frac{\vec{A}_z}{A}\right)$$

2.3.4. เวกเตอร์ตำแหน่ง (Position Vector)

เวกเตอร์ตำแหน่ง หมายถึง เวกเตอร์ที่บอกตำแหน่งของวัตถุเทียบกับจุดใดจุดหนึ่งเรียกว่า จุดอ้างอิง



รูปที่ 2.11 เวกเตอร์บอกตำแหน่งแบบ 3 มิติ

ที่มา : http://www.rsu.ac.th/science /physics/kan/general_phy/vector/vector.htm

จากรูป 2.11เวกเตอร์$\vec{r_1}$และ$\vec{r_2}$เป็นเวกเตอร์บอกตำแหน่งของจุด P และ Q เทียบกับจุด O ในระบบพิกัด โดย

$$\vec{r_1} = \vec{OP} = x_1\hat{i} + y_1\hat{j} + z_1\hat{k}$$

$$\vec{r_1} = \vec{OQ} = x_2\hat{i} + y_2\hat{j} + z_2\hat{k}$$

จะได้

$$\vec{PQ} = \vec{r_2} - \vec{r_1} \qquad (9)$$

$$\vec{PQ} = (x_2 - x_1)\hat{i} + (y_2 - y_1)\hat{j} + (z_2 - z_1)\hat{k} \qquad (10)$$

โดยขนาดของ$\vec{PQ}$ คือ

$$PQ = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \qquad (11)$$

ทิศทางของ$\vec{PQ}$หาได้จาก

$$\theta_x = \cos^{-1}\left(\frac{(x_2 - x_1)}{PQ}\right),\ \theta_y = \cos^{-1}\left(\frac{(y_2 - y_1)}{PQ}\right),\ \theta_z = \cos^{-1}\left(\frac{(z_2 - z_1)}{PQ}\right)$$

2.4 เซนเซอร์ตรวจจับการเคลื่อนไหวแบบ 3 มิติ หรือ Kinect

Kinect คือ อุปกรณ์เสริมของเครื่องเล่นเกม Xbox360 จาก Microsoft โดยสามารถจดจำผู้เล่น (Facial Recognition) และให้ผู้เล่นควบคุมเกมส์ผ่านทางการเคลื่อนไหวร่างกายของผู้เล่นโดยตรง (3D Motion Recognition) โดยไม่จำเป็นต้องมี joystick อีกต่อไป และสามารถจดจำเสียงของผู้เล่นได้ (Voice Recognition) และนอกจากใช้ฟังค์ชั่นพวกนี้เล่นเกมส์แล้วเรายังใช้Kinect ควบคุมการดูหนัง ฟังเพลง

Kinect เป็นการพัฒนาร่วมระหว่าง Rare (บริษัทลูกของ Microsoft Game Studios) และ Prime Sense (บริษัทสัญชาติ Israel) โดย Rare เป็นผู้พัฒนาซอฟต์แวร์ ส่วน Prime Sense นั้นเป็นผู้พัฒนาระบบ Range Camera (กล้องที่มีสามารถความแตกต่างของระยะทางของวัตถุต่างๆในภาพ 2 มิติ) ซึ่งระบบ Range Camera นั้นสามารถวิเคราะห์อากัปกิริยาแสดงท่าทางของผู้ใช้ (Gestures Detection) และถูกนำมาใช้เป็นระบบควบคุมที่ไม่ต้องใช้มือจับต้อง (Hands-free Control) ระบบ Gestures Detection นั้นประกอบด้วย อินฟาเรดโปรเจคเตอร์ กล้อง และ ไมโครชิปที่สามารถติดตามการเคลื่อนไหวแบบสามมิติของวัตถุหลายชิ้นหรือแต่ละชิ้นได้ในเวลาเดียวกัน

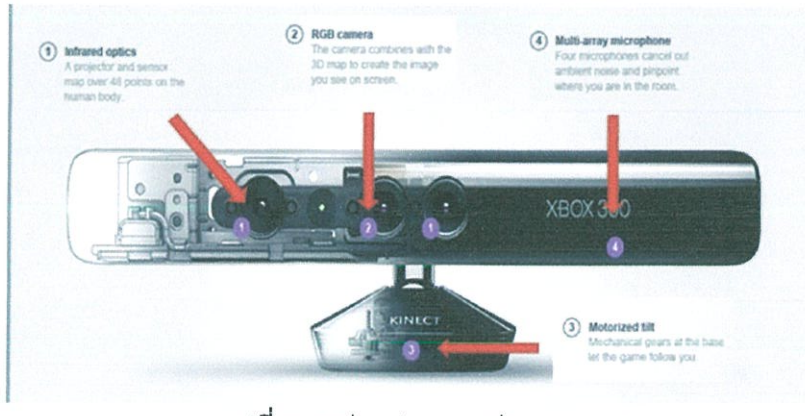
Microsoft ระบุว่า Kinect สำหรับ Xbox 360 นั้นสามารถติดตามความเคลื่อนไหวได้สูงสุด 6 คน โดยมีผู้เล่นได้พร้อมกัน 2 คน และสามารถตรวจจับการเคลื่อนไหวส่วนต่างๆของร่างกายผู้เล่นได้ทั้งหมด 20 จุด ทางด้าน Prime Sense นั้นระบุว่าจำนวนคนที่ระบบสามารถตรวจจับได้นั้นไม่จำกัด แต่จะถูกจำกัดโดยพื้นที่ที่กล้องสามารถจับภาพได้

2.4.1 ส่วนประกอบของ Kinect นั้นประกอบด้วย

1. มอเตอร์ที่สามารถเคลื่อนไหวขึ้นลง
2. กล้อง RGB (RGB Camera)
3. เซ็นเซอร์วัดระยะ (Depth Sensor)
4. ไมโครโฟน 4 ตัว (Multi-array Microphone)

2.4.2 ระบบต่างๆที่ Kinect รองรับ

1. ตรวจจับการเคลื่อนไหว 3 มิติแบบเต็มตัว (Full-body 3D Motion Capture)
2. จดจำใบหน้า (Facial Recognition)
3. จดจำเสียง (Voice Recognition)



รูปที่ 2.12ส่วนประกอบต่างๆ ของ Kinect

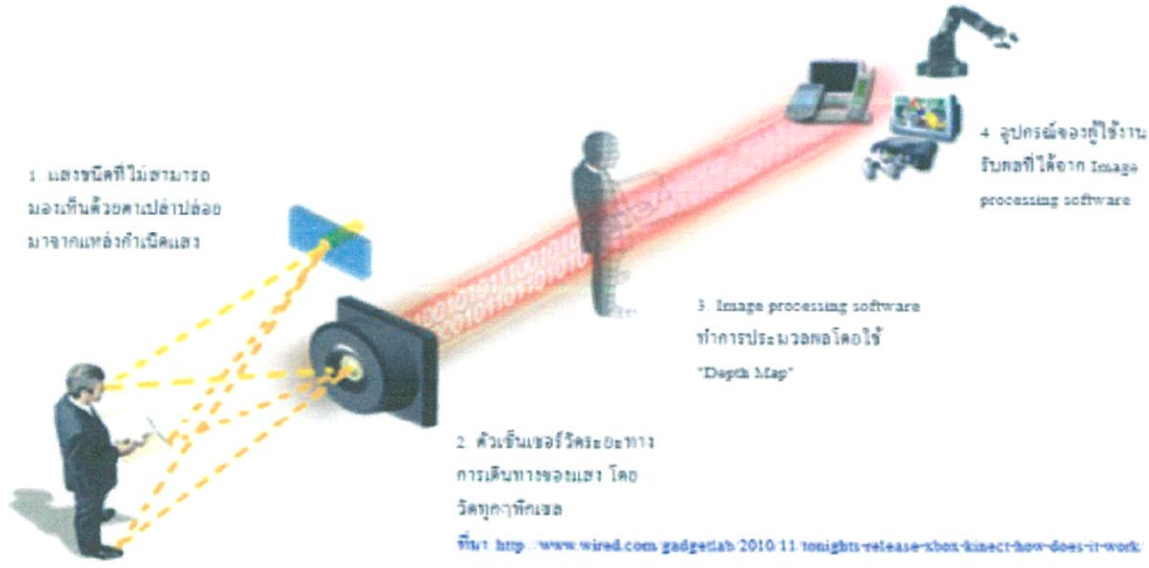
ที่มา : http://mcu56.learninginventions.org/?page_id=244

ตารางที่ 2.2ตารางแสดงรายละเอียดของ Kinect

Property	Spec
Field of View (Horizontal, Vertical, Diagonal)	58° H, 45° V, 70° D
Depth image size	VGA (640×480)
Spatial x/y resolution (@ 2m distance from sensor)	3mm
Depth z resolution (@ 2m distance from sensor)	1cm
Maximum image throughput (frame rate)	60fps
Operation range	0.8m – 3.5m
Color image size	UXGA (1600×1200)
Audio: built-in microphones	Two microphones
Audio: digital inputs	Four inputs
Data interface	USB 2.0
Power supply	USB 2.0
Power consumption	2.25W
Property	**Spec**
Dimensions (Width x Height x Depth)	14cm x 3.5cm x 5cm
Operation environment (every lighting condition)	Indoor
Operating temperature	0°C – 40°C

ที่มา : https://msdn.microsoft.com/en-us/library/jj131033.aspx

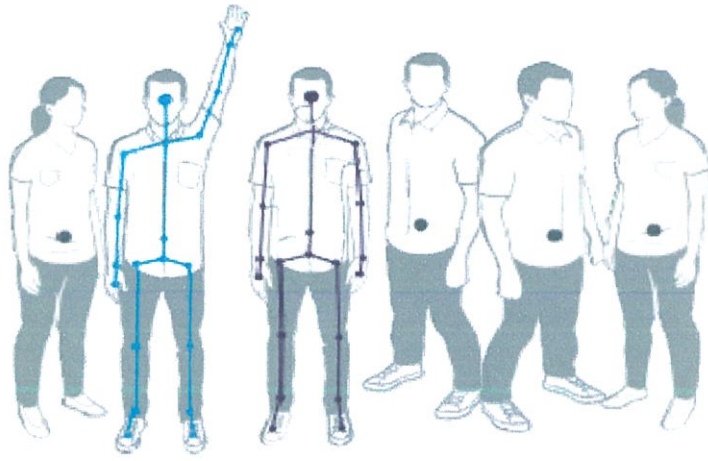
2.4.3 หลักการทำงานของระบบตรวจวัดระยะ



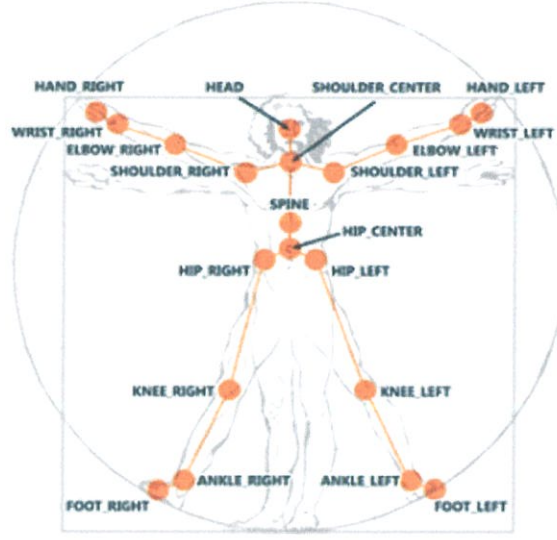
รูปที่ 2.13 แผนภาพแสดงการทำงานของระบบตรวจวัดระยะ

ในการวัดระยะหรือความลึกของวัตถุต่างๆนั้นอาศัยการทำงานของอุปกรณ์เซ็นเซอร์วัดระยะ (Depth Sensor) ซึ่งประกอบอุปกรณ์ 2 ชนิดคือ Infrared Laser Projector และ Monochrome CMOS Sensor โดยตัว Infrared Laser Projector จะส่งแสงอินฟราเรดไปกระทบกับวัตถุที่อยู่ในแนวรัศมี เมื่อแสงอินฟราเรดกระทบกับวัตถุก็จะสะท้อนกลับมาที่ตัวรับ Monochrome CMOS Sensor จากนั้นซอฟต์แวร์ก็จะคำนวณระยะทางอาศัยเวลาการเดินทางของแสงอินฟราเรด

เมื่อ Kinect รู้ระดับความตื้นลึกแล้วก็จะสามารถแยกผู้เล่นออกจากสภาพแวดล้อมภายในห้องได้ นอกจากนั้น Microsoft Kinect Sensor ยังมีระบบ Skeletal Tracking ที่ใช้ติดตามโครงกระดูกของผู้ใช้งาน ซึ่งสามารถติดตามได้มากที่สุด 2 คน แต่จะมองเห็นทั้งหมด 6 คน ซึ่งภาพที่แสดงจะเป็นภาพของโครงกระดูกมนุษย์ที่แทนด้วยข้อต่อ 20 จุดสำคัญตามร่างกาย



(ก)



(ข)

รูปที่ 2.14 (ก)การจับจุด Skeletal Tacking
(ข) จุด Skeletal Tacking

ที่มา : https://kinectasia.wordpress.com/tag/kinect/

ส่วนสำคัญของ Kinect นั้นคือ NUI API ซึ่งเป็นส่วนเชื่อมต่อระหว่าง Kinect Sensors (VGA Camera, Monochrome Camera, IR Camera, and 4 Microphones) กับคอมพิวเตอร์ นอกจากนี้หน้าที่สำคัญอีกอย่างหนึ่งคือการนำข้อมูลต่างๆ เช่น รูปภาพ ความตื้นลึกของรูป ไปใช้ต่อ โดยข้อมูลเหล่านี้เมื่อถูกประมวลผลแล้วสามารถนำไปใช้งานในรูปแบบ Skeletal Tracking

ไดรเวอร์สำหรับ Kinect ที่มาพร้อม NUI API สามารถรองรับการทำงานของ Kinect Sensors มากกว่า 1 เครื่อง ซึ่งจำนวนเครื่อง Kinect นั้นสามารถตั้งได้โดยใช้ฟังก์ชั่นที่มากับ NUI API นอกจากนี้ยังสามารถกำหนดชื่อให้แต่ละ Kinect แต่ไดร์ฟเวอร์ที่มีนี้ไม่รองรับการทำงาน มากกว่า 1 แอปฯในเวลาเดียวกัน

ในการนำข้อมูลต่างๆที่ได้จาก Kinect ไปใช้งานนั้นต้องมีการกำหนดก่อนว่าจะใช้การข้อมูลประเภทไหน หากไม่กำหนดตั้งแต่เริ่มต้นจะไม่สามารถนำข้อมูลมาใช้ได้ในระหว่างการทำงาน ประเภทของข้อมูลที่สามารถนำมาใช้ได้คือ

- สี (Color) แอปพลิเคชั่นสามารถนำสีของรูปที่ได้จาก Kinect มาใช้งาน
- ความลึก (Depth) ข้อมูลความตื้นลึกของภาพจาก Kinect จะถูกนำไปใช้ในแอปพลิเคชั่น
- ความลึกและหมายเลขผู้เล่น (Depth and Player index) สำหรับแอปฯที่สามารถใช้งานมากกว่า 1 คนขึ้นไป
- Skeleton เป็นข้อมูลตำแหน่งต่างๆของร่างกาย

2.4.4 Kinect สำหรับ Windows

ทางบริษัท Microsoft ได้ปล่อยชุดดีเวลลอป (Software Development Kit ,SDK) สำหรับ Windows 7 เมื่อวันที่ 16 มิถุนายน 2554 ซึ่งตัว SDK นี้จะมีไดร์ฟเวอร์ต่างๆของ Kinect สำหรับคอมพิวเตอร์พีซีรวมอยู่ด้วย ซึ่งโปรแกรมที่สามารถนำมาใช้พัฒนาแอปพลิเคชั่น Kinect ได้แก่ C++ C# และ Visual Basic ส่วนฟีเจอร์ต่างๆที่ถูกรวมไว้ใน SDK ด้วย ได้แก่

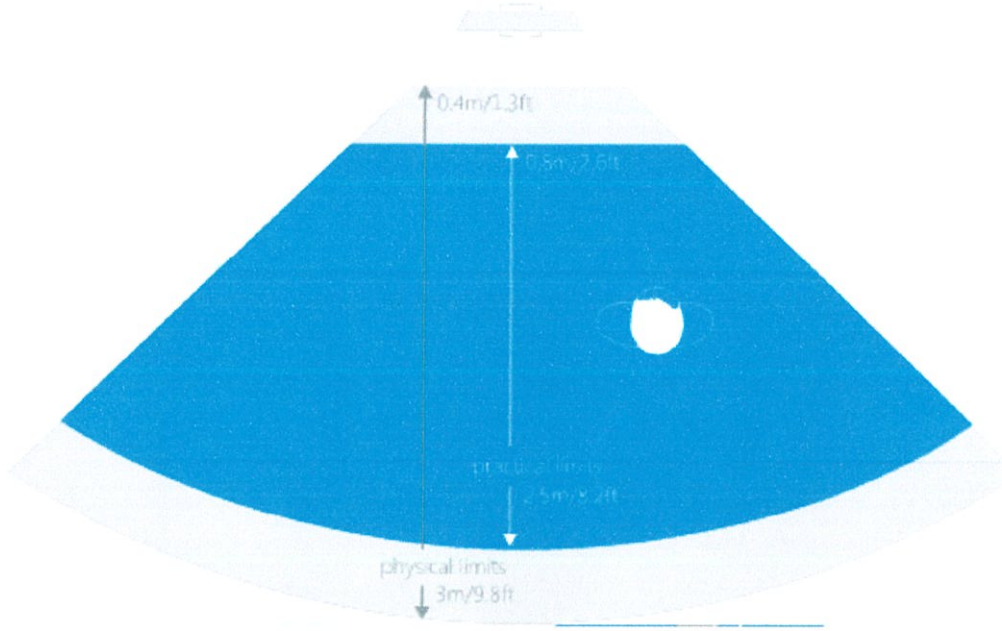
1. Raw Sensor Streams สำหรับการนำข้อมูลดิบที่ได้จากเซ็นเซอร์ต่างๆไปใช้งาน
2. Skeletal Tracking สำหรับติดตามโครงร่างของมนุษย์ ซึ่งสามารถติดตามได้สูงสุด 2 คน
3. พร้อมกัน ความโดดเด่นของฟีเจอร์นี้คือ สามารถนำไปพัฒนาเป็นแอปพลิเคชั่นความคุมด้วยท่าทาง (Gestures Control)
4. Advanced Audio Capabilities ใช้ในการจำแนกเสียงต่างๆ ซึ่งสามารถกำจัดเสียงรบกวนต่างๆ (Nouse Suppression and Echo Cancellation) และแยกเสียงของแต่ละบุคคลได้ ซึ่งฟีเจอร์นี้สามารถไปนำไปพัฒนาเป็นการควบคุมด้วยเสียง (Voice Recognition)
5. โค้ดโปรแกรมตัวอย่าง และ เอกสารประกอบ

System Requirements

1. ระบบประมวลผลแบบ 32 หรือ 64 บิต
2. ระบบประมวลผลขั้นต่ำ Dual-core 2.66 GHz
3. ช่อง USB 2.0 (ห้ามใช้ร่วมกับอุปกรณ์อื่น)
4. หน่วยความจำอย่างต่ำ 2 GB
5. ระบบปฏิบัติการ Windows 7

การพัฒนาแอปพลิเคชั่นนั้นสามารถใช้ได้ทั้ง Kinect ที่มากับเครื่อง Xbox 360 หรือจะใช้ Kinect สำหรับ Windows โดยราคาของ Kinext สำหรับ Xbox นั้นจะมีราคาถูกกว่ามาก หากใช้ Kinect ที่มาพร้อม Xbox รุ่นที่เป็น Xbox 360 (slim + Kinect) หรือ Xbox 360 (slim) ต้องใช้อุปกรณ์เสริมคือ Kinect Power Cable เพราะว่าไม่มีแถมมาให้ แต่ถ้าเป็น Xbox 360 รุ่นเก่าก็จะมาพร้อมสาย USB และ สาย Power ซึ่งพร้อมใช้งานได้ทันที

ส่วนความแตกต่างระหว่างของเวอร์ชั่นสำหรับ Xbox กับ Windows คือ Kinect สำหรับ Windows นั้นสามารถนำไปพัฒนาเพื่อธุรกิจได้ และ ระบบ Near Mode นั้นจะใช้ได้แต่ในรุ่นที่เป็น Windows เท่านั้น เพราะรุ่น Windows สามารถใช้งานได้ระยะใกล้สุดคือ 40 เซนติเมตร แต่รุ่น 360 ใช้ได้ใกล้สุดที่ 80 เซนติเมตร ส่วนฮาร์ดแวร์อื่นๆนั้นแตกต่างกันเล็กน้อย

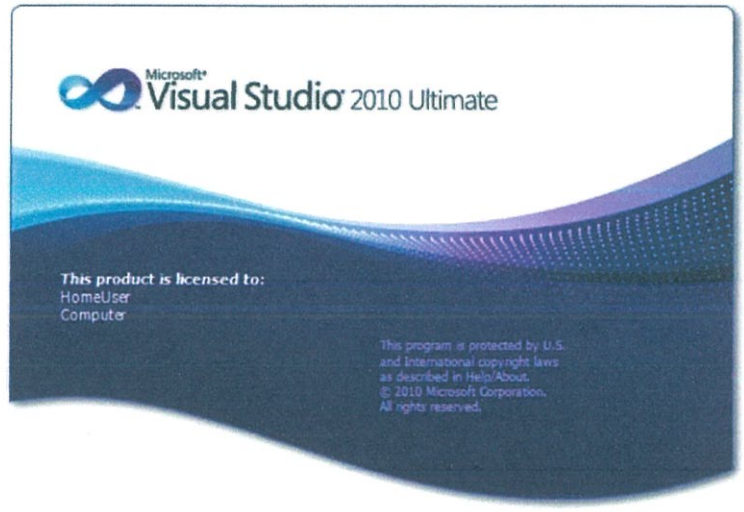


รูปที่ 2.15 ระยะที่สามารถใช้งานของ Kinect

ที่มา : https://msdn.microsoft.com/en-us/library/hh973074.aspx

2.5 โปรแกรม Microsoft Visual Studio 2010

Microsoft Visual Studio คือ Integrated Development Environment พัฒนาขึ้นโดย ไมโครซอฟท์ ซึ่งเป็นเครื่องมือที่ช่วยนักพัฒนาซอฟต์แวร์พัฒนาโปรแกรมคอมพิวเตอร์ web site, web application และ Web service ระบบที่รองรับการทำงานนั้นมี Microsoft Windows Pocket Pc, Smartphone และ Web browser ในปัจจุบัน Visual Studio นั้นสามารถใช้ภาษาโปรแกรมที่เป็น .Netในโปรแกรมเดียวกัน เช่น VB.NET C++ C# J# เป็นต้น



รูปที่ 2.16โปรแกรม Microsoft Visual Studio 2010

ที่มา : http://dakdown.net/img_src/util.120814135621.kHHpk.png

ขั้นตอนการเขียนโปรแกรมด้วย Visual Basic .NET

1. ออกแบบหน้าจอของโปรแกรม โดยนำคอนโทรลต่างๆ จาก toolbox มาวางบนฟอร์ม แล้วปรับขนาดและตำแหน่งตามที่ต้องการ
2. กำหนด Property ของฟอร์มและออบเจ็กต์ต่างๆ บนฟอร์ม โดยใช้วินโดว์ Properties
3. การเขียนโค้ดควบคุมการทำงานของโปรแกรม
4. รันโปรแกรมเพื่อทดสอบว่าโปรแกรมทำงานได้ตามที่ต้องการหรือไม่
5. ค้นหาข้อผิดพลาด ถ้าหากโปรแกรมเกิดข้อผิดพลาดหรือทำงานไม่ได้ตามที่ต้องการ
6. คอมไพล์ Project ไปเป็นไฟล์โปรแกรมที่รันได้ Executable File

2.6 ภาษา Visual Basic

2.6.1 ประวัติความเป็นมาของภาษา Visual Basic

ภาษา BASIC ถูกสร้างในปี ค.ศ. 1963 โดย Hohn Keneny และ Thomas Kurtz ที่วิทยาลัย Dartmouth ในเบื้องต้นพวกเขามีจุดมุ่งหมายในการพัฒนาภาษา Basic ขึ้น เพื่อใช้ในการสอนแนวในการเขียนโปรแกรม โดยเน้นที่รูปแบบง่าย ๆ เพื่อสะดวกในการใช้งาน ในปี 1970 Microsoftได้เริ่มผลิตตัวแปรภาษา Basic ใน Rom ขึ้น เช่น Chip Radio Sheek TRS-80 เป็นต้น ต่อมาได้พัฒนาเป็น GWBasic ซึ่ง เป็น Interpreter ภาษาที่ใช้กับ MS-Dos และในปี 1982 Microsoft QuickBasic ได้รับการพัฒนาขั้นโดยเพิ่มความสามารถในการรันโปรแกรมให้เป็น Executed Program รวมทั้งทำให้ Basicมีความเป็น "Structured Programming" มากขึ้น โดยการตัด Line Number ทิ้งไปเพื่อลบข้อกล่าวหาว่าเป็นภาษาคอมพิวเตอร์ที่มีโครงสร้างในลักษณะSpaghetti Code มาใช้รูปแบบของ Subprogram และ User Defined รวมทั้งการใช้ Structured Data Type และการ

พัฒนาการใช้งานด้านกราฟิกให้มีการใช้งานในระดับที่สูงขึ้น รวมทั้งมีการใช้เสียงประกอบได้ เหมือนกับภาษาคอมพิวเตอร์อื่น ๆ เช่น Turbo C และ Turbo Pascal เป็นต้น Visual Basicเป็นภาษาคอมพิวเตอร์ที่ได้รับความนิยมในการนำมาใช้งานพัฒนาโปรแกรมบนระบบ Windows เนื่องจาก เป็นภาษาคอมพิวเตอร์ที่ใช้เทคโนโลยีในลักษณะ Visualizeนั่นก็คือจะสะดวกในการหยิบเครื่องไม้เครื่องมือที่โปรแกรมได้จัดเตรียมไว้ให้สำหรับออกแบบหน้าจอและสิ่งต่าง ๆ สำหรับในการเขียนโปรแกรมให้เรียบร้อยต่างจากสมัยก่อนเวลาจะออกแบบหน้าจอก็ยังคงต้องมานั่งเขียนSource Codeให้ลำบาก

Visual Basicเป็นเครื่องมือที่ใช้ในการพัฒนาโปรแกรมขึ้นใช้งาน ที่ใช้ได้ตั้งแต่ระดับต้น เพื่อใช้สร้างโปรแกรมง่าย ๆ บน Windows หรือโปรแกรมเมอร์ระดับกลาง ที่จะเรียกใช้ฟังชั่นต่าง ๆได้อย่างมีประสิทธิภาพ ตลอดจนโปรแกรมเมอร์ระดับมืออาชีพ ที่จะพัฒนาโปรแกรมในระดับสูง โดยการใช้ Object Linking and Embedding (OLE) และ Application Programming Interface (API) ของระบบ windows มาประกอบการเขียนโปรแกรม

2.6.2 Visual Basic

เป็นเครื่องมือที่ใช้สร้างโปรแกรมต่างๆ เช่น

- โปรแกรมที่รันบนระบบปฏิบัติการ windows เช่น โปรแกรมคำนวณเลข
- โปรแกรมฐานข้อมูล เช่น Microsoft access , Microsoft SQL server
- คอมโพแน้นต์ทางด้าน Active X
- โปรแกรมที่รันบนอินเตอร์เน็ต

2.6.3 ส่วนประกอบของ Visual Basic

โดยทั่วไป เราจะใช้ Project Standard. EXE ซึ่งเป็นการเขียนโปรแกรมที่รันบนวินโดวส์Project คือ กลุ่มของ File ที่เราจะนำมารวมกันเพื่อสร้างโปรแกรม
รายระเอียดของส่วนประกอบต่างๆ ของหน้าจอ

- Menu bar
- Tool bar
- Tool box
- Project explorer
- Properties window
- Form

2.6.4 หลักในการเขียนโปรแกรมใน Visual Basic ได้แบ่งออกเป็น 2 ขั้นตอนหลัก คือ

การออกแบบหน้าจอของโปรแกรมเป็นส่วนที่ใช้ติดต่อกับผู้ใช้ เรียกว่า user interface การเขียนโปรแกรมเป็นการกำหนดคุณสมบัติของคอนโทรล บนฟอร์มให้เหมาะสม และเขียนคำสั่งตอบสนองอีเว็นต์

2.6.5 การออกแบบหน้าจอของโปรแกรมด้วยคอนโทรล

- คอนโทรล (Control) เป็นเครื่องมือที่ใช้ในการออกแบบคอนโทรลที่เป็นพื้นฐาน
- เท็กบ็อกซ์ text box ใช้รับข้อมูลจากผู้ใช้

- เลเบล (Label) ใช้แสดงข้อมูลบางอย่างแก่ผู้ใช้
- ปุ่มคำสั่ง (Command button) ให้ผู้ใช้คลิกเมาส์เพื่อทำให้คอมพิวเตอร์ทำงานบางอย่าง
- คุณสมบัติ (Properties) คือ ลักษณะต่างๆ ของคอนโทรลที่ถูกนำมาวางบนฟอร์มที่เราสามารถกำหนดได้เช่น ข้อความที่ปรากฏบนคอนโทรล, รูปแบบฟอนต์
- เท็กบ็อกซ์ text box มีคุณสมบัติ text ที่ใช้กำหนดข้อความที่จะแสดง
- เลเบล (Label) มีคุณสมบัติ Caption ที่ใช้กำหนดข้อความที่จะแสดง
- ปุ่มคำสั่ง (Command button) มีคุณสมบัติ caption ที่ใช้กำหนดข้อความที่จะแสดง

2.6.6 WPF Application

WPF ย่อมาจาก Windows Presentation Foundation เป็นแนวคิดการพัฒนา UI (User Interface) แนวหนึ่งซึ่งมีความยืดหยุ่นสูงมาก มีกระบวนการ Data Binding ที่ยืดหยุ่นสูง ทำให้คุณสามารถเขียน UI ที่ซับซ้อนได้สะดวก โดย Programmer สามารถออกแบบในมุมมองเสมือนว่ามันเป็น Projection ของ Data ได้ ข้อดีที่เด่นๆ ของมันก็คือ คุณสามารถเขียน View Model Object เพื่อควบคุมการไหลเวียนของ Data ครั้งเดียว โดยที่ UI จะเป็นเหมือนหน้ากากครอบ โดยที่มันจะเปลี่ยนไปเป็นแบบไหนก็ได้ ตราบเท่าที่มัน Bind กับ View Model Object อย่างถูกต้อง ทำให้สามารถที่จะ Unit Test ตัว UI ได้โดยผ่าน View Model Object ออกแบบมาเป็นอย่างดี

นอกจากนี้ WPF ยังจะสนับสนุน UI ที่สร้างสรรค์ มี Animation Framework และอนุญาตให้คุณทำแทบทุกอย่างที่ Graphic Designer ออกแบบมาได้ โดยผ่าน XAML ซึ่งสะดวก ง่ายมากถึงมากที่สุดในการที่จะทำให้โปรแกรมมีหน้าตาดีๆ มีลูกเล่น Animation ดังที่คุณอาจจะเคยเห็นใน Mac OSX หรือโปรแกรมสมัยใหม่เช่นบน Microsoft Surface คุณสามารถสร้างสรรค์สิ่งเหล่านั้นได้ผ่าน WPF

สรุปข้อดีของ WPF Application

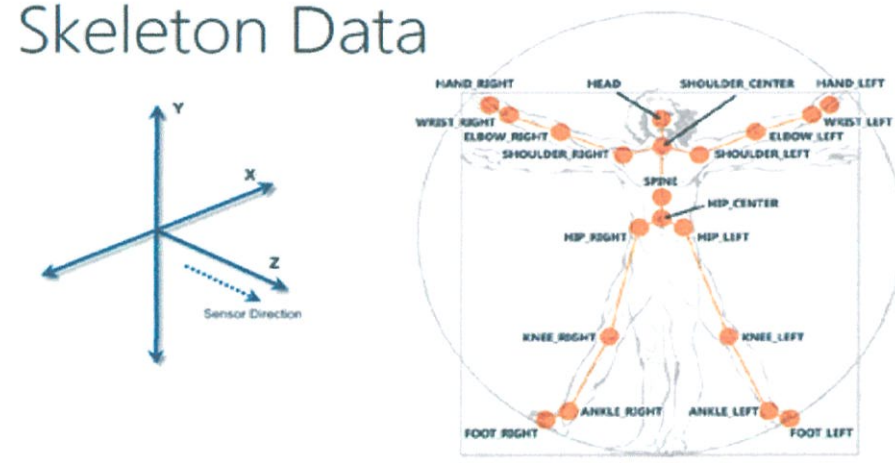
- สามารถสร้างโปรแกรมให้สวยงามได้ด้วยStyle(คล้าย CSS ของ web)
- มีกราฟฟิคที่สวยงาม
- สร้าง Animation ในโปรแกรมได้ง่ายด้วย Storyboard
- สามารถแบ่งหน้าที่การทำงานระหว่าง Designer และ Programmer ได้อย่างชัดเจน
- กำหนดControl(Tool) ของเราเองได้ด้วย User Control
- ไม่ต้องพิมพ์ Code มาก ใช้ interface สั่งการส่วนใหญ่

บทที่ 3
หลักการทำงานและการออกแบบ

บทนี้จะกล่าวถึง หลักการทำงานฟังก์ชั่นSkeleton tacking ของ Kinect ว่าระบุการบอกพิกัดตำแหน่งของแต่ละจุด การทำงานและแผนภาพการทำงานของระบบแบบเป็นขั้นตอน และลักษณะท่าทางการทดสอบมุมของทั้ง 3 จุด คือ หัวไหล่ ศอก และหัวเข่า

3.1 หลักการ skeleton tacking

เมื่อทำการเปิดฟังก์ชั่น skeleton tacking จับระยะห่างจากตัวกล้อง Kinect ประมาณ 3.5 เมตร เซ็นเซอร์จะทำการจับระยะจุดพิกัดและติดตามการเคลื่อนไหวของผู้ทดสอบ โดยพิกัดที่จับจะเป็นแบบ 3 มิติ ซึ่งจะวัดจากข้อต่อกระดูกของร่างกายทั้งหมด 20 จุด พิกัดเริ่มต้นระยะ (0,0,0) วัดจากจุดที่ กระดูกเชิงกราน(Hip center) เป็นจุดกลาง พิกัดและแกนจะเป็นดังรูป

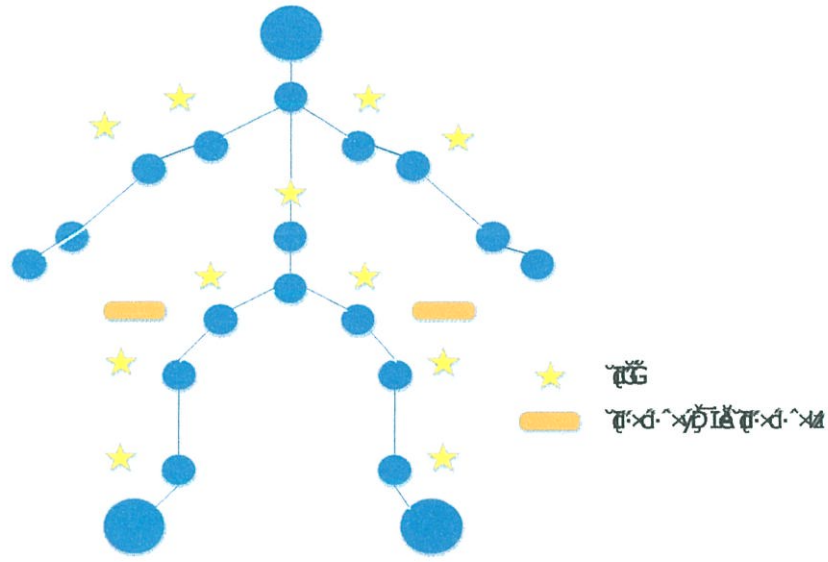


รูปที่ 3.1 จุด Skeleton Tacking

ที่มา : http://i.imgur.com/OdxJm.png

ในโครงงานนี้จะทำการจับจุด skeleton tacking ทั้งหมด 11 จุด คือ หัวไหล่ซ้าย (Shoulder left) หัวไหล่ขวา (Shoulder Right) ศอกซ้าย (Elbow lift) ศอกขวา (Elbow Right) กระดูกเชิงกราน(Hip center) สะโพกซ้าย (Hip left) สะโพกขวา (Hip right) เข่าซ้าย (Knee left) เข่าขวา (Knee right) เท้าซ้าย (Foot left) และ เท้าขวา(Foot right) โดยจะใช้ค่าจุดในแนวแกน Y (Position Y) ทำการวัดระยะของจุดกระดูกเพื่อดูค่าความผิดปกติซึ่งค่าจุดที่เข่าและเท้าจะเป็นค่าติดลบเนื่องจากพิกัดจุดกลาง (Position center)จุดที่กระดูกเชิงกราน (Hip center) เป็นจุดเริ่มต้น

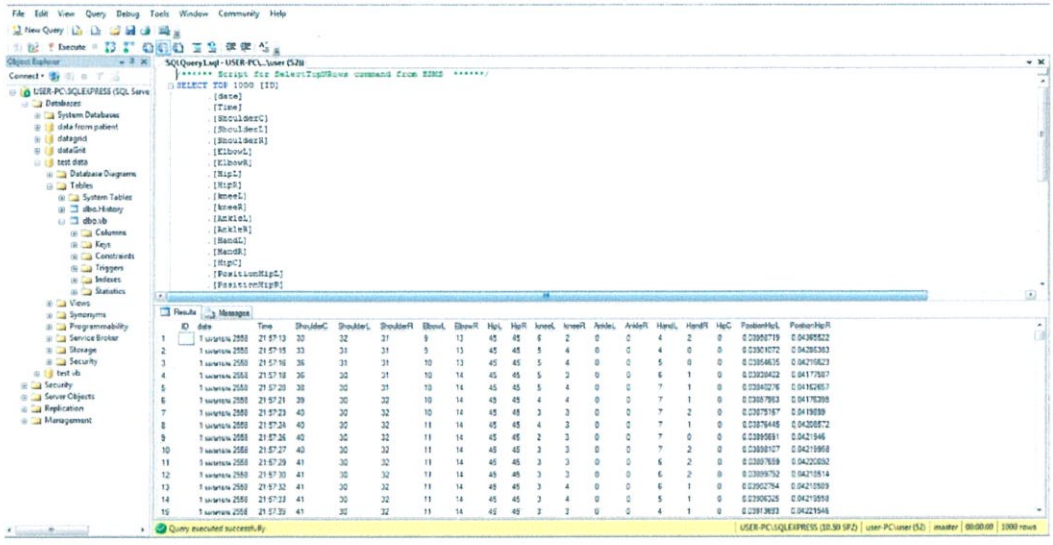
3.2 การทำงานของระบบ



รูปที่ 3.2 ตำแหน่งที่แสดงค่ามุม ค่าแกนตั้งของแต่ละข้อ

จากรูปที่ 3.2 แสดงตำแหน่งที่แสดงค่ามุม และค่าแกนตั้งในแต่ละข้อต่อซึ่งจากคำแนะนำของนักกายภาพบำบัดได้แนะนำว่า “ที่บริเวณสะโพกควรใช้ค่าแนวแกนตั้ง หรือค่าแกน Z เพื่อใช้วิเคราะห์กรณีที่ต้องการดูการยกตัวของสะโพก หรือผู้ป่วยในกลุ่มปวดหลังปวดสะโพก และส่วนอื่นใช้ค่ามุมเพื่อดูความผิดปกติ” ซึ่งในแต่ละข้อต่อจะมีท่าทางในการทดสอบต่างกันขึ้นอยู่กับอาการปวด โดยนักกายภาพจะเป็นผู้กำหนด และสรุปการรักษา

รูปที่ 3.3หน้าต่างโปรแกรมโหมดการทำงานที่ 1 แบบเก็บข้อมูล

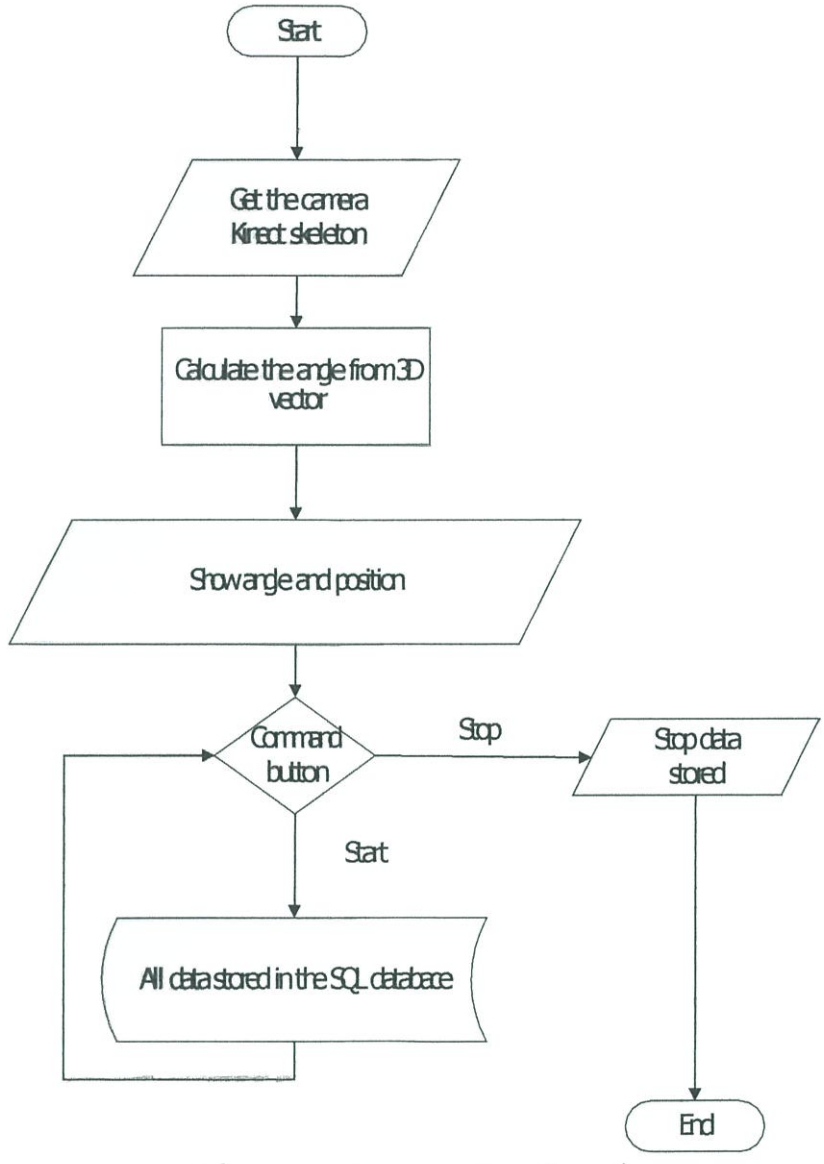


รูปที่ 3.4ภาพการเก็บค่าลงบนฐานข้อมูล SQL

จากรูปที่ 3.3 และ 3.4แสดงโปรแกรมในโหมดที่ 1 คือ โหมดที่ใช้เก็บประวัติผู้ป่วยโดยจะมีช่องให้เจ้าหน้าที่กรอกเช่น HN หรือรหัสผู้ป่วย ชื่อ-สกุล อายุ เพศ น้ำหนัก ความสูง และอุบัติเหตุที่เคยเกิดขึ้นในอดีตหรืออาการความเจ็บปวดของผู้ป่วย โหมดนี้ยังสามารถที่จะเก็บค่ามุมที่อ่านได้จากเซนเซอร์ทั้งหมด 11 จุดดั้งนี้ หัวไหล่ซ้าย (Shoulder left) หัวไหล่ขวา (Shoulder Right) ศอกซ้าย (Elbow lift) ศอกขวา (Elbow Right) กระดูกเชิงกราน (Hip center) สะโพกซ้าย (Hip left) สะโพกขวา (Hip right) เข่าซ้าย (Knee left) เข่าขวา (Knee right) เท้าซ้าย (Foot left) และ เท้าขวา (Foot right) ไว้ในฐานข้อมูล SQL ข้อมูลเหล่านี้ถูกนำมาวิเคราะห์ลักษณะความผิดปกติของผู้ป่วยและการรักษา และระบบยังสามารถถ่ายภาพการเคลื่อนไหวขณะผู้ป่วยทำการทดสอบเพื่อให้นักกายภาพวิเคราะห์ได้ง่ายขึ้นขณะใช่ระบบนี้

ซึ่งการวิเคราะห์ข้อมูลที่ได้จากระบบ ระบบจะอ่านค่ามุมแต่จุดออก ค่าได้จากเซนเซอร์มีค่าใกล้เคียงกับวัดด้วยเครื่องวัดโกนิโอมิเตอร์ โดยนักกายภาพจะใช่มุมสองมุมซ้ายและขวาเปรียบเทียบกัน เช่น มุมที่เข้าซ้ายและเข่าขวา เปรียบเทียบกัน เนื่องจากแต่ละคนมีค่ามุมแต่ละจุดต่างกัน ต้องอ้างอิงมุมซ้ายและมุมขวาแทนการอ้างอิงค่าของมุมมาตรฐาน ซึ่งจะนำค่ามุมมาตรฐานมาดูเพียงว่ามุมข้างไหนมีค่าใกล้เคียงกับมุมมาตรฐาน และข้างที่ผิดปกติจะมีมุมที่แตกต่างจากมุมอีกข้างอย่างเห็นได้ชัดจึงทำให้ทราบถึงความผิดปกติแต่จะมากหรือน้อยขึ้นอยู่กับมุมที่แตกต่างของมุมข้างซ้ายและมุมข้างขวา

ซึ่งโหมดนี้ช่วยให้นักกายภาพบำบัดการวิเคราะห์ลักษณะความผิดปกติทางการเดิน ร่วมไปถึงการรักษาง่ายขึ้นอีกด้วย ซึ่งข้อมูลที่ได้จากระบบเป็นข้อมูลแบบเวลาปัจจุบัน ระบบนี้จะช่วยให้ผู้ป่วยเห็นความผิดปกติของตนได้อย่างเด่นชัด โดยจะมีหลักการทำงานดังนี้



รูปที่ 3.5 Flow Chart การทำงานโหมดที่ 1

จากรูปที่3.5 แสดง Flow Chart การทำงานในโหมดแบบเก็บข้อมูล โดยเริ่มจากเปิดโปรแกรม Kinect จะเริ่มทำการจับค่า Skeleton tacking ทั้ง 20 จุดจากนั้นจะทำการแสดงค่ามุมและตำแหน่งและเมื่อต้องการเก็บข้อมูลให้กรอกประวัติผู้ที่ทำการทดสอบแล้วกดปุ่มบันทึก ต่อจากนั้นกดปุ่ม Start โปรแกรมจะเริ่มทำการเก็บข้อมูลทั้งหมดลงบนตารางใน SQL หากต้องการหยุดเก็บค่าให้กดปุ่ม Stop

3.3 ลักษณะท่าทางในการทดสอบ

การทดสอบจะแบ่งเป็นการดูในค่ามุม 3 จุด ซึ่งมีลักษณะและท่าทางการทดสอบ ดังนี้

รูปที่ 3.6 การทดสอบวัดค่ามุมของข้อต่อหัวไหล่

จากรูปที่3.6 เป็นท่าการทดสอบการวัดค่ามุมของข้อต่อหัวไหล่ในท่ายืนยกแขนขึ้นสุดและท่าเหยียดแขนลงแนบลำตัวโดยทำการทดสอบทีละข้าง ข้างซ้ายและขวาตามลำดับ

รูปที่ 3.7 การทดสอบวัดค่ามุมของข้อศอก

จากรูปที่ 3.7 เป็นท่าการทดสอบการวัดค่ามุมของข้อศอกในท่ากางแขนออกด้านข้างตรงขนานกับแนวพื้นและงอเข้าหาลำตัวจนสุดโดยทำการทดสอบทีละข้าง ข้างซ้ายและขวาตามลำดับ

รูปที่ 3.8การทดสอบวัดค่ามุมของข้อเข่า

จากรูปที่ 3.8 เป็นท่าการทดสอบการวัดค่ามุมของข้อเข่าในท่ายืนตรงและงอขาไปด้านหลัง โดยทำการเอียงตัวไปทางตรงข้ามกับข้างที่ทำการทดลองประมาณ 30 – 45 องศาโดยทำการทดสอบทีละข้าง ข้างซ้ายและขวาตามลำดับ

บทที่4
การทดลองและผลการทดลอง

4.1 การทดลองวัดค่ามุมของคนปกติ

ทำการทดลองวัดค่ามุมของคนปกติจากระบบหลังจากเขียนโปรแกรมเสร็จเพื่อตรวจสอบค่าที่ได้จากระบบว่าจะมีแนวโน้มค่ามุมที่ได้

4.1.1 การทดลองวัดค่ามุมของคนปกติ

1. ทำการติดตั้งระบบโดยให้ผู้ทดสอบยืนอยู่ห่างจากกล้องประมาณ 2 เมตรจึงจะทำการเปิดโปรแกรม โดยระหว่างที่ทำการทดลองนั้นจะต้องไม่ให้มีผู้อื่นที่ไม่ใช่ผู้ทำการทดสอบเข้ามาในกล้อง

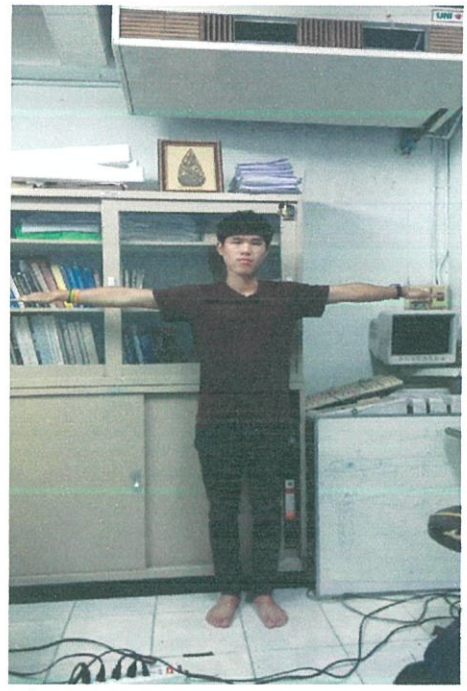
2. เริ่มทำการทดลองโดยให้ผู้ที่เข้าทดสอบทำท่าทางตามที่ออกแบบไว้จากนั้นก็เริ่มทำการเก็บค่าจากการทดลอง

3. ทำการทดลองซ้ำ 3 ครั้ง โดยมีท่าที่ใช้ในการทดลองดังรูปที่ 4.1 ถึง รูปที่ 4.4

- ท่ายืนตรงแขนแนบลำตัว

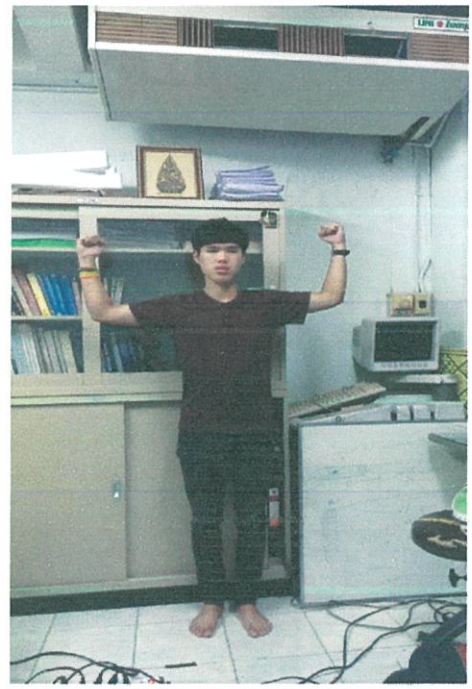
รูปที่ 4.1 ท่ายืนตรงแขนแนบลำตัว

- ท่ายืนตรงกางแขนออกไปข้างลำตัว



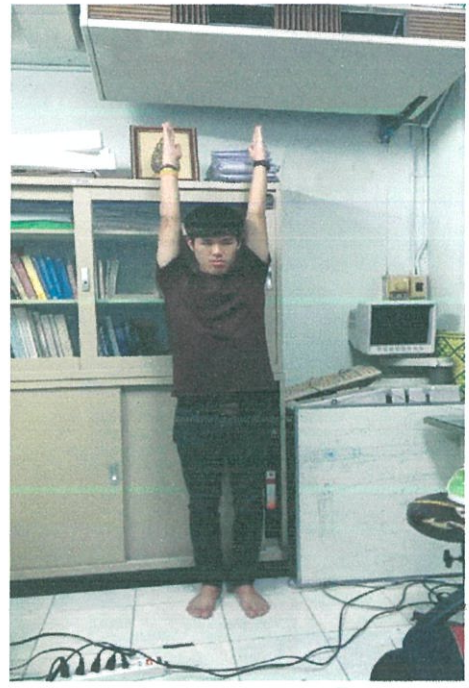
รูปที่ 4.2 ท่ายืนตรงกางแขนออกไปข้างลำตัว

- ท่ายืนตรงกางแขนแล้วหักข้อศอกทำมุมประมาณ 90 องศา



รูปที่ 4.3 ท่ายืนตรงกางแขนแล้วหักข้อศอกทำมุมประมาณ 90 องศา

- ท่ายืนตรงยกแขนทั้งสองข้าง



รูปที่ 4.4 ท่ายืนตรงยกแขนทั้งสองข้าง

4.1.2 ผลการทดลองวัดค่ามุมของคนปกติ

ตารางที่ 4.1ผลการทดลองวัดค่ามุมของคนปกติครั้งที่ 1

ท่าทดลอง	เข่าซ้าย	เข่าขวา	ข้อศอกซ้าย	ข้อศอกขวา	ไหล่ซ้าย	ไหล่ขวา
ยืนตรง	11.75	8.78	9.71	13.61	160.38	159.13
ยกแขนสุด	10.93	7.93	13.48	10.41	31.66	34.4
กางแขนตรง	13.79	10.89	6.47	4.37	69.99	71.25
งอข้อศอก	13.42	10.4	74.69	81.92	70.16	77.06

ตารางที่ 4.2ผลการทดลองวัดค่ามุมของคนปกติครั้งที่ 2

ท่าทดลอง	เข่าซ้าย	เข่าขวา	ข้อศอกซ้าย	ข้อศอกขวา	ไหล่ซ้าย	ไหล่ขวา
ยืนตรง	15.86	11.2	10.24	14.18	161.82	158.99
ยกแขนสุด	10.44	7.64	16.43	15.11	31.82	26.97
กางแขนตรง	14.67	9.79	12.31	5.2	67.41	69.38
งอข้อศอก	15.23	11.88	42.05	64.56	47.92	68.57

ตารางที่ 4.3ผลการทดลองวัดค่ามุมของคนปกติครั้งที่ 3

ท่าทดลอง	เข่าซ้าย	เข่าขวา	ข้อศอกซ้าย	ข้อศอกขวา	ไหล่ซ้าย	ไหล่ขวา
ยืนตรง	11.87	12.9	161.21	158.61	10.53	12.84
ยกแขนสุด	9.71	5.7	28.42	28.51	30.85	18.51
กางแขนตรง	14.37	12.1	70.87	72.07	5.57	3.99
งอข้อศอก	13.45	12.23	61.29	71.81	57.32	66.94

4.2 การทดลองวัดค่ามุมของระบบในคนปกติเทียบกับเครื่องมือวัดโกนิโอมิเตอร์

ทำการทดลองวัดค่ามุมข้อต่อของคนปกติแล้วนำไปเปรียบเทียบกับเครื่องมือวัดโกนิโอมิเตอร์เพื่อต้องการทราบความผิดพลาดของระบบก่อนที่จะทำการ Calibrate

4.2.1 การทดลองวัดค่ามุมของระบบในคนปกติเทียบกับเครื่องมือวัดก่อนทำการ Calibrate

1. ทำการติดตั้งระบบโดยให้ผู้ทดสอบยืนอยู่ห่างจากกล้องประมาณ 2 เมตรจึงจะทำการเปิดโปรแกรม โดยระหว่างที่ทำการทดลองนั้นจะต้องไม่ให้มีผู้อื่นที่ไม่ใช่ผู้ทำการทดสอบเข้ามาในกล้อง

2. เริ่มทำการทดลองโดยให้ผู้ที่เข้าทดสอบทำท่าทางตามที่ออกแบบไว้จากนั้นก็เริ่มทำการเก็บค่าจากการทดลองของระบบที่ละท่าสลับกับการวัดด้วยเครื่องมือวัด

3.ทำการทดลอง 3 คน คนละ 3 ครั้ง โดยมีท่าทางในการทดลองดังรูปที่ 4.5 ถึง รูปที่ 4.8

- ท่ายืนตรงแขนแนบลำตัว

(ก) (ข)

รูปที่ 4.5 (ก) กาวัดมุมด้วยระบบในท่ายืนตรง
(ข) การวัดมุมด้วยเครื่องมือวัดในท่ายืนตรง

- ท่ายืนตรงกางแขนออกไปข้างลำตัว

(ก) (ข)

รูปที่ 4.6 (ก) การวัดมุมด้วยระบบท่ายืนตรงกางแขนออกไปข้างลำตัว
(ข)การวัดมุมด้วยเครื่องมือวัดท่ายืนตรงกางแขนออกไปข้างลำตัว

- ท่ายืนตรงกางแขนแล้วหักข้อศอกทำมุมประมาณ 90 องศา

(ก) (ข)

รูปที่ 4.7 (ก)การวัดมุมด้วยระบบในท่ายืนตรงกางแขนแล้วหักข้อศอกทำมุมประมาณ 90 องศา
(ข)การวัดมุมด้วยเครื่องมือวัดยืนตรงกางแขนแล้วหักข้อศอกทำมุมประมาณ 90 องศา

- ท่ายืนตรงยกแขนทั้งสองข้าง

(ก) (ข)

รูปที่ 4.8 (ก)การวัดมุมด้วยระบบในท่ายืนตรงยกแขนทั้งสองข้าง
(ข)การวัดมุมด้วยเครื่องมือวัดในท่ายืนตรงยกแขนทั้งสองข้าง

4.2.2 ผลการทดลองวัดค่ามุมของระบบในคนปกติเทียบกับเครื่องมือวัดโกนิโอมิเตอร์ (ก่อนCalibrate)

ตารางที่ 4.4ผลการทดลองวัดค่ามุมของระบบในคนปกติเทียบกับเครื่องมือวัดโกนิโอมิเตอร์ (ก่อนCalibrate)

บริเวณ	ค่ามุมที่วัดจากระบบ (องศา)			เครื่องมือโกนิโอมิเตอร์ (องศา)			ค่าความผิดพลาด(%)
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	
หัวไหล่ซ้าย	154	161	155	180	180	180	12.778
หัวไหล่ขวา	157	157	156	180	180	180	12.778
ศอกซ้าย	55	49	56	90	90	90	20.00
ศอกขวา	78	64	103	90	90	90	4.444
หัวเข่าซ้าย	4	12	8	0	0	0	4.444
หัวเข่าขวา	6	15	9	0	0	0	5.556

4.3 การทดลองวัดค่ามุมจากระบบเทียบกับเครื่องมือวัดโกนิโอมิเตอร์

ทำการทดลองเพื่อทดสอบความแม่นยำของระบบด้วยการวัดมุมข้อต่อตามส่วนต่างๆของร่างกายด้วยระบบแล้วนำผลที่ได้ไปเทียบกับผลที่ได้จากการวัดด้วยเครื่องมือวัดโกนิโอมิเตอร์อีกครั้งหลังจากทำการ Calibrate แล้ว

4.3.1วิธีการทดลองวัดค่ามุมจากระบบเทียบกับเครื่องมือวัดโกนิโอมิเตอร์หลังทำการ Calibrate

1.จะทำการทดลองโดยให้ผู้ที่ทำการทดสอบยืนอยู่ห่างจากกล้องประมาณ 2 เมตร จึงจะทำการเปิดโปรแกรม โดยระหว่างที่ทำการทดลองนั้นจะต้องไม่ให้มีผู้อื่นที่ไม่ใช่ผู้ทำการทดสอบเข้ามาในกล้อง

2.ทำการเลือกโหมดการทดลองแล้วให้ผู้ที่เข้าทดสอบทำท่าให้ให้ตรงกับโหมดที่ต้องการวัด

- การวัดมุมที่หัวเข่าซ้าย

(ก) (ข)

รูปที่ 4.9(ก)การวดมุมที่หัวเข่าซ้ายด้วยระบบ
(ข)การวัดมุมที่หัวเข่าซ้ายด้วยโกนิโอมิเตอร์

- การวัดมุมที่หัวเข่าขวา

(ก) (ข)

รูปที่ 4.10(ก)การวัดมุมหัวเข่าซ้ายด้วยระบบ
(ข)การวัดมุมเข่าขวาด้วยโกนิโอมิเตอร์

- การวัดมุมที่ข้อศอกซ้าย

(ก) (ข)

รูปที่ 4.11(ก)การวัดมุมที่ข้อศอกซ้ายด้วยระบบ
(ข)การวัดมุมที่ข้อศอกซ้ายด้วยโกนิโอมิเตอร์

- การวัดมุมที่ข้อศอกขวา

(ก) (ข)

รูปที่ 4.12 (ก)การวัดมุมที่ข้อศอกขวาด้วยระบบ
(ข) การวัดมุมที่ข้อศอกขวาด้วยโกนิมิเตอร์

- การวัดมุมที่หัวไหล่ซ้าย

(ก) (ข)

รูปที่ 4.13(ก) การวัดมุมที่หัวไหลซ้ายด้วยระบบ
(ข) การวัดมุมที่หัวไหลซ้ายด้วยโกนิโอมิเตอร์

- การวัดมุมที่หัวไหล่ขวา

(ก) (ข)

รูปที่ 4.14(ก)การวัดมุมที่หัวไหล่ขวาด้วยระบบ
(ข) การวัดมุมที่หัวไหล่ขวาด้วยโกนิโอมิเตอร์

- การวัดมุมที่สะโพกซ้าย-ขวา

รูปที่ 4.15การวัดมุมที่สะโพกซ้าย- ขวา

4.3.2 ผลการทดลองวัดค่ามุมของระบบในคนปกติเทียบกับเครื่องมือวัดโกนิโอมิเตอร์ (หลังการ Calibrate)

ตารางที่ 4.5ผลการทดลองวัดค่ามุมของระบบในคนปกติเทียบกับเครื่องมือวัดโกนิโอมิเตอร์ (หลังการ Calibrate)

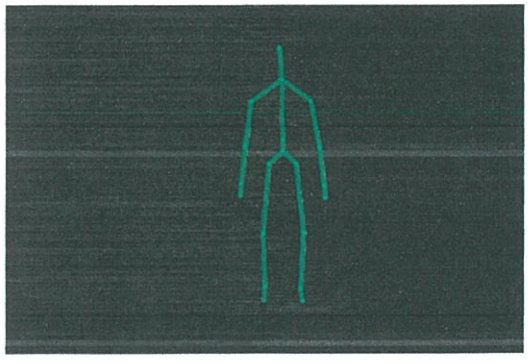
ท่าทาง	บริเวณ	ค่ามุมที่วัดจากระบบ (องศา)			เครื่องมือโกนิโอมิเตอร์ (องศา)			ค่าความผิดพลาด (%)
		ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	
เหยียดตรง	หัวไหล่ซ้าย	172	172	176	180	180	180	3.703
เหยียดตรง	หัวไหล่ขวา	172	173	173	180	180	180	4.074
งอ	ศอกซ้าย	137	133	138	136	135	136	0.185
งอ	ศอกขวา	132	139	133	134	134	132	0.740
งอ	หัวเข่าซ้าย	82	65	87	110	93	109	14.444
งอ	หัวเข่าขวา	75	78	72	109	92	98	13.703

4.4การทดลองวัดค่ามุมของผู้ป่วยที่ทำกายภาพบำบัด

เป็นการทดลองโดยการนำระบบออกไปใช้กับผู้ป่วยจริงตามโรงพยาบาล ซึ่งจะได้พบกับผู้ป่วยที่อยู่ระหว่างการทำกายภาพบำบัดโดยจะแบ่งผู้ป่วยออกเป็น case ต่างๆ

4.2.1 ผู้ป่วยปวดสะโพก

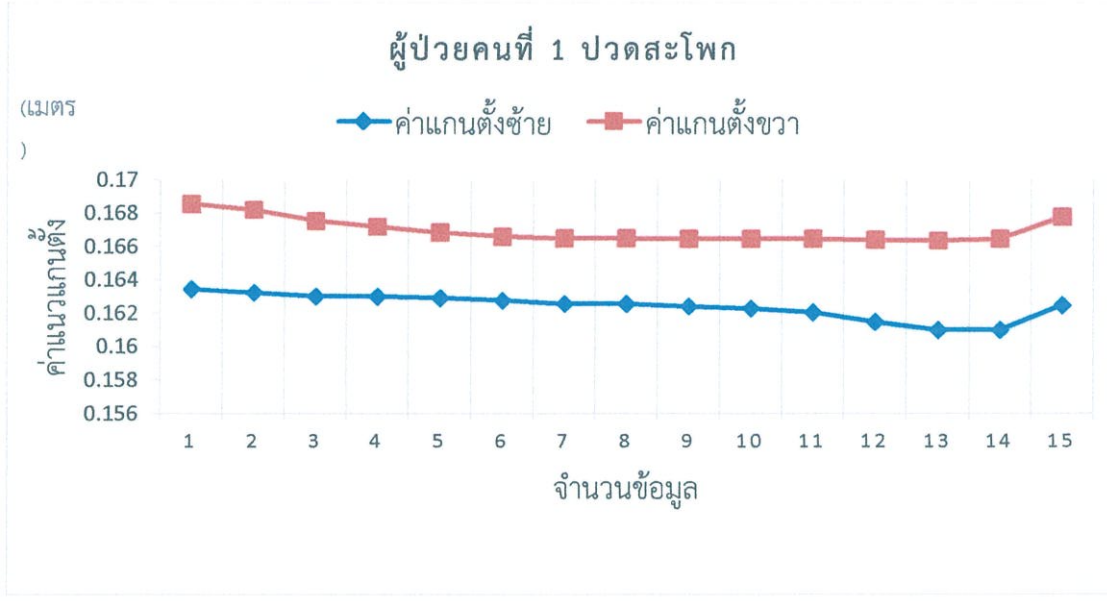
ชื่อ นายวิฑูรย์ เพชรวัตร์ อายุ 51 ปี ส่วนสูง 174 เซนติเมตร น้ำหนัก 90 กิโลกรัม



(ก)

(ข)

รูปที่ 4.16(ก)ภาพจำลองของ skeletons ผู้ป่วยปวดสะโพก
(ข) ภาพแสดงท่าทางที่ผู้ป่วยปวดสะโพกทำการทดสอบ



กราฟที่ 4.1ค่าแกนตั้งของสะโพกข้างซ้าย และขวา

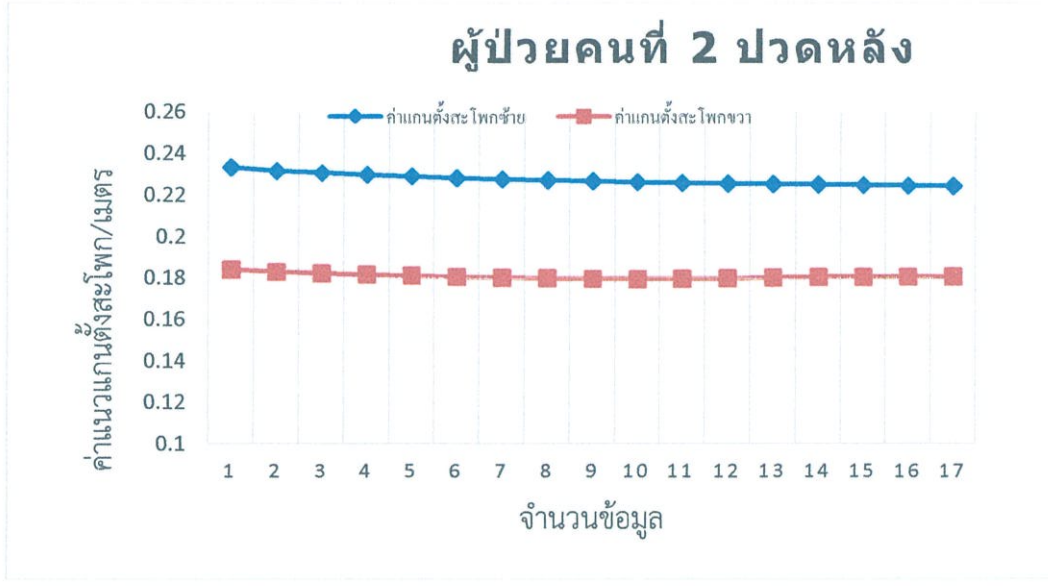
4.2.2 ผู้ป่วยปวดหลัง

ชื่อ นายพิเชษฐ์ อายุ 40ปี ส่วนสูง 180เซนติเมตร น้ำหนัก 75กิโลกรัม

(ก)

(ข)

รูปที่ 4.17(ก)ภาพจำลองของ skeletons ผู้ป่วยปวดหลัง
(ข) ภาพแสดงท่าทางที่ผู้ป่วยปวดหลังทำการทดสอบ



กราฟที่ 4.2ค่าแกนตั้งของสะโพกข้างซ้าย และขวา

4.2.3 ผู้ป่วยที่ผ่าตัดหัวเข่า

1. หัวเข่าขวา

ชื่อ นางสำราญ วิโรจน์ อายุ 54ปี ส่วนสูง 157เซนติเมตร น้ำหนัก 61กิโลกรัม

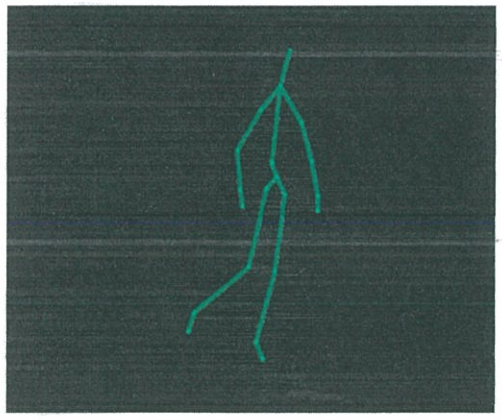
(ก)

(ข)

รูปที่ 4.18(ก)ภาพจำลองของ skeletons ผู้ป่วยผ่าตัดหัวเข่า

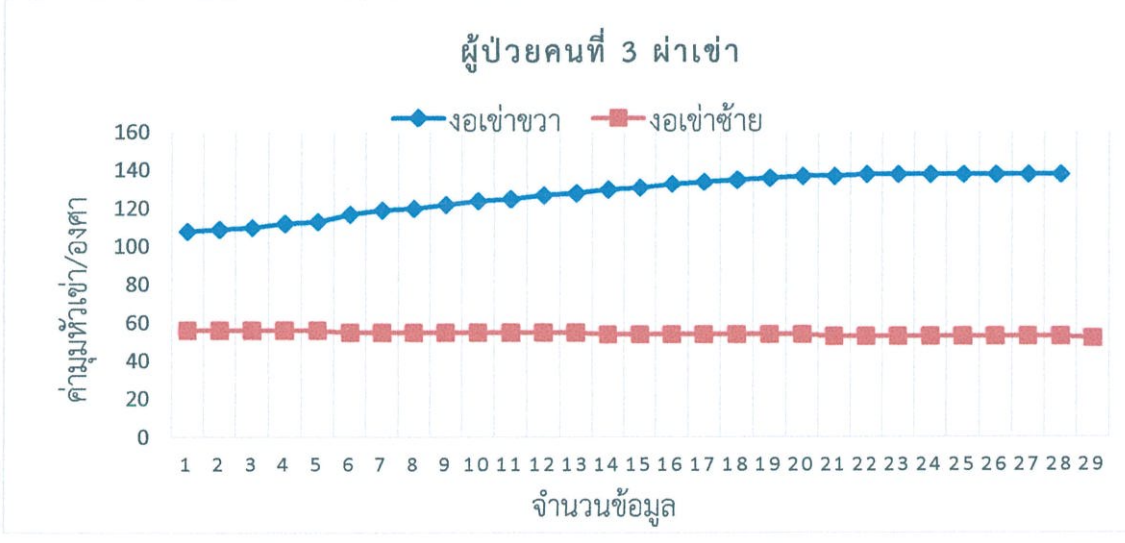
(ข) ภาพแสดงท่าทางที่ผู้ป่วยผ่าตัดหัวเข้าทำการทดสอบ

2. หัวเข่าซ้าย



(ก) (ข)

รูปที่ 4.19(ก)ภาพจำลองของ skeletons ผู้ป่วยผ่าตัดหัวเข่า
(ข) ภาพแสดงท่าทางที่ผู้ป่วยผ่าตัดหัวเข้าทำการทดสอบ



กราฟที่ 4.3ค่ามุมของเข่าข้างซ้าย และขวา

4.5การทดลองผลการกายภาพบำบัด

เป็นการวัดและประเมินผลของการกายภาพบำบัดของคนปกติเทียบกับค่ามาตรฐาน

ตารางที่ 4.6ค่ามุมที่ทำการกายภาพบำบัด

ท่าทาง	บริเวณที่ทดสอบ	ค่ามุมมาตรฐาน (องศา)	ค่ามุมที่วัดได้จากระบบ			ค่าเฉลี่ยค่ามุม (องศา)
			ครั้งที่ 1 (องศา)	ครั้งที่ 2 (องศา)	ครั้งที่ 3 (องศา)	
ยืนตรง	ข้อเท้า	0	2	2	1	1.666
ยืนตรง	หัวเข่าขวา	0	4	4	1	3.00
งอ	หัวเข่าขวา	135	113	110	119	127.00
ยืนตรง	หัวเข่าซ้าย	0	6	4	6	5.333
งอ	หัวเข่าซ้าย	135	114	116	121	117.00
เหยียดตรง	ข้อศอกขวา	0	6	3	3	4.00
งอ	ข้อศอกขวา	140	136	129	140	135.00
เหยียดตรง	ข้อศอกซ้าย	0	6	5	6	5.600
งอ	ข้อศอกซ้าย	140	138	134	127	133.000
เหยียดตรง	หัวไหล่ขวา	180	173	168	179	173.333
เหยียดตรง	หัวไหล่ซ้าย	180	177	174	171	174.000

บทที่5
สรุปผลการทดลองและข้อเสนอแนะ

ในบทนี้จะกล่าวถึงสรุปผลการทดลอง ข้อเสนอแนะและปัญหาอุปสรรคต่างๆ เพื่อนำไปปรับปรุงแก้ไขพัฒนาโครงงานต่อไป ซึ่งอธิบายถึงปัญหาและอุปสรรคขณะทำโครงงานรวมทั้งการแก้ไขในส่วนของการวัดค่ามุมให้มีความแม่นยำมากยิ่งขึ้น

5.1 สรุปการทดลอง

จากการทดลองระบบตรวจจับความผิดปกติของผู้ป่วยที่มีความผิดปกติทางการเดิน สามารถสร้างระบบตรวจจับ และเก็บข้อมูลค่ามุมและตำแหน่งของข้อต่อร่างกายโดยระบบจะทำการวัดค่าตำแหน่งของข้อต่อต่างๆของร่างกายและทำการคำนวณค่า ออกมาเป็นมุมแล้วนำมาแสดงที่หน้าต่างแสดงผลของโปรแกรมโดยระบบนี้จะสามมารถเก็บข้อมูลลงบนฐานข้อมูล SQL ได้โดยจะสามารถเลือกได้ว่าจะให้เก็บข้อมูลเวลาไหนโดยจะทำการเก็บข้อมูลแบบ real โดยระบบสามารถวัดค่ามุมข้อต่อของร่างกายและสามารถแยกแยะความผิดปกติของข้อต่อข้างใดข้างหนึ่งได้เมื่อนำข้อมูลของข้อต่อทั้งสองข้างมาเปรียบเทียบกันซึ่งตรงกับจุดประสงค์ที่ต้องการ

จากการทดลองการสร้างระบบวัดค่ามุมข้อต่อของร่างกายพบว่าสามารถวัดค่าได้อย่างแม่นยำเมื่อนำไปเปรียบเทียบกับเครื่องมือวัดทางการแพทย์ โกนิโอมิเตอร์อยู่ 2 จุดด้วยกันคือข้อต่อบริเวณหัวไหล ที่สามารถวัดได้ค่อนข้างแม่นยำโดยในท่ายกแขนหัวไหล่ซ้ายและขวามีความผิดพลาดเพียง 3.70 กับ 4.07 เปอร์เซ็นต์ ซึ่งถือว่าอยู่ในเกณฑ์ที่ยอมรับได้ ส่วนอีกจุดหนึ่งก็คือข้อต่อบริเวณข้อศอก ซึ่งความผิดพลาดในการวัดมุมของข้อศอกด้วยระบบเมื่อนำไปเปรียบเทียบกับค่าที่วัดด้วยเครื่องวัดโกนิโอมิเตอร์ พบว่าระบบวัดได้ค่อนข้างแม่นยำโดยความผิดพลาดของข้อศอกข้างซ้ายและขวา อยู่ที่ 0.18และ0.14 เปอร์เซ็นต์ ซึ่งอยู่ในเกณฑ์ที่ยอมรับได้เช่นกันและอีกจุดหนึ่งที่มีการทำการทดลองคือข้อต่อบริเวณหัวเข่าแต่กลับพบว่าระบบมีความผิดพลาดค่อนข้างมากโดยความผิดพลาดของหัวเข่าซ้ายและขวา คือ 14.44 และ 13.70 เปอร์เซ็นต์ เป็นค่าที่ไม่สามารถยอมรับได้ เนื่องจากทางการแพทย์มีค่าความผิดพลาดที่สามารถยอมรับได้อยู่ที่ประมาณไม่เกิน5 เปอร์เซ็นต์ โดยได้มีการเพิ่มการทดลองในส่วนของการวัดตำแหน่งในแนวแกนดิ่งของสะโพกเพื่อหาความแตกต่างของสะโพกในผู้ป่วยที่มีความผิดปกติบริเวณสะโพกพบว่าระบบสามมารถทำงานได้เป็นอย่างดีโดยจะสามารถบอกได้ว่าผู้ป่วยมีความผิดปกติที่สะโพกข้างใดจากการเทียบตำแหน่งในแนวแกนดิ่งของสะโพกทั้งสองข้างแต่ระบบยังมีความผิดพลาดเป็นบ้างบริเวณซึ่งจะทำการแก้ไข้ไปในอนาคต และระบบยังสามารถที่จะใช้ตรวจสอบข้อต่อหัวไหล่ และข้อศอก ซึ่งในอนาคตอาจจะใช้ตรวจสอบผู้ป่วยที่มีอาการไหล่ติด หรือผู้ป่วยที่มีปัญหาเรื่องข้อศอก

5.2 ปัญหาและอุปสรรค

1. การใช้อุปกรณ์ Kinect ในการวัดค่ามุมของข้อต่อบริเวณร่างกายอย่างละเอียดที่ใช้ในทางการแพทย์สามารทำได้ค่อนข้างยาก เนื่องจากความไม่ละเอียดของตัวอุปกรณ์จึงทำให้เกิดความผิดพลาดขึ้นได้ค่อนข้างมาก

2. ข้อจำกัดของตัวอุปกรณ์ Kinect ที่มีข้อจำกัดค่อนข้างมาก ทั้งในด้านการทำงานของกล้องที่จะต้องจับคนแค่คนเดียวและต้องอยู่ที่ระยะที่เหมาะสม จึงจะช่วยลดความผิดพลาดในการวัดค่ามุมได้

3. อุปกรณ์ Kinectไม่สามารถจับจุดข้อต่อที่ซ้อนกันได้ ซึ่งหากมีการซ้อนของจุดจะเกิดค่ามุมที่ผิดเพี้ยน

5.3 ข้อเสนอแนะ

1. Kinect รุ่นปัจจุบันที่ใช้เป็นไปได้ยากที่จะทำการวัดค่ามุมออกมาอย่างละเอียดตามที่ต้องการ

2. การทดลองเก็บค่ามุมโดยใช้ Kinect ที่ต้องการความแม่นยำควรจะทำในพื้นที่ปิดที่มีการรบกวนจากสิ่งที่เคลื่อนไหวน้อยที่สุด

3. การวัดค่ามุมของเครื่องมือวัดโกนิโอมิเตอร์นักกายภาพบำบัดจะทำการวัดจากแนวเคลื่อนที่เดิมจึงทำให้ค่าที่ได้ไม่ตรงกับการวัดด้วยเซ็นเซอร์ 3 มิติ

บรรณานุกรม

[1] Acta Med Philipp. 2009 Jan-Mar; 43(1):16-21. English. Published online 2009 December 31, 2009 "Measurement of Performance of Basic Daily Skills and Assessment of Motor Function of Filipino Adolescent and Adult Persons with Down Syndrome" University of the Philippines Manila and Philippine Council for Health Research and Development of the Department of Science and Technology [online]. เข้าถึงได้จาก : www.http://apamedcentral.org

[2] B Gao, "Accuracies of skinmarkerbasedkneemotionanalysisusing Different technigues". University of Florida, Gainesville, Florida USA, 2007

[3] Joumana Medlej ,6 Mar 2014, "Human Anatomy Fundamentals: Flexibility and Joint Limitations" [online]. เข้าถึงได้จาก : www.design.tutsplus.com

[4] K. Berger, K. Ruhl, Y. Schroeder, C. Bruemmer, Al. Scholz, andM. Magnor, "Markerless Motion Capture using multiple Color-Depth Sensors". Proc. Vision, Modeling and Visualization (VMV) 2011, pp. 317–324, Oct. 2011.

[5]Microsoft Dep.Network, "Kinect for Windows Sensor Components and Specifications" [online]. เข้าถึงได้จาก: www.msdn.microsoft.com

[6]RSU, "Vector", [online]เข้าถึงได้จาก:www.rsu.ac.th

[7]Thanawat Raibroycharoen, 9 April 2012 "เจาะลึกโครงสร้างของ Kinect สำหรับ Windows" [online]. เข้าถึงได้จาก : www.kinectasia.wordpress.com

[8] เฉลิม คงเขียว .วิทยานิพนธ์ เรื่อง ระบบเคลื่อนไหวสำหรับการวิเคราะห์การเดินของมนุษย์. มหาวิทยาลัยสงขลานครินทร์ .2553

[9]"ประวัติความเป็นมาของVisual Basic", เข้าถึงได้จาก: www.sites.google.com

ภาคผนวก ก.

Code programm

โปรแกรมตรวจจับและเก็บค่ามุม

```
Imports System
Imports System.IO.Ports
Imports System.Threading

Imports System.Text
Imports Microsoft.Kinect
Imports Coding4Fun.Kinect.Wpf
Imports System.IO
Imports System.Data.SqlClient


Namespace SkeletalTracking


PartialPublicClassMainWindow
InheritsWindow


Dim Sqlcon AsSqlConnection
Dim ds AsDataSet
Dim da AsSqlDataAdapter
Dim constr AsString = "server=USER-
PC\SQLEXPRESS;Database=test
data;Trusted_connection=True;"

PublicShared ScreenMaxX AsInteger = 320
PublicShared ScreenMaxY AsInteger = 480
PrivateWithEvents sensor AsKinectSensor
PublicSubNew()

InitializeComponent()

EndSub

PrivateSub Window_Loaded(ByVal sender
AsObject, ByVal e AsRoutedEventArgs)
DiscoverKinectSensor()

EndSub

PrivateSub
sensor_SkeletonFrameReady(ByVal sender
AsObject, ByVal e As
Microsoft.Kinect.SkeletonFrameReadyEventA
rgs) Handles sensor.SkeletonFrameReady
Dim skFrame AsSkeletonFrame
Dim skeletonSlot AsInteger = 0
Dim playerSkeleton AsSkeleton

skFrame = e.OpenSkeletonFrame

IfNot skFrame IsNothingThen
DimskeletonData(skFrame.SkeletonArrayLen
gth - 1) AsSkeleton
skFrame.CopySkeletonDataTo(skeletonData)
playerSkeleton = (From s In skeletonData
Where s.TrackingState =
SkeletonTrackingState.Tracked Select
s).FirstOrDefault()
EndIf
IfNot playerSkeleton IsNothingThen

Dim FootRKinectX =
(playerSkeleton.Joints(JointType.FootRight).P
osition.X)
Dim FootRKinectY =
(playerSkeleton.Joints(JointType.FootRight).P
osition.Y)
```

```
Dim FootRKinectZ =
(playerSkeleton.Joints(JointType.FootRight).P
osition.Z)
Dim aFootRKinectZ =
(playerSkeleton.Joints(JointType.FootRight).P
osition.Z)
Dim aFootRKinectX =
(playerSkeleton.Joints(JointType.FootRight).P
osition.X)
Dim aFootRKinectY =
(playerSkeleton.Joints(JointType.FootRight).P
osition.Y)

Dim spineKinectX =
(playerSkeleton.Joints(JointType.Spine).Positi
on.X)
Dim spineKinectY =
(playerSkeleton.Joints(JointType.Spine).Positi
on.Y)
Dim spineKinectZ =
(playerSkeleton.Joints(JointType.Spine).Positi
on.Z)



Dim FootLKinectX =
(playerSkeleton.Joints(JointType.FootLeft).P
osition.X)
Dim FootLKinectY =
(playerSkeleton.Joints(JointType.FootLeft).P
osition.Y)
Dim FootLKinectZ =
(playerSkeleton.Joints(JointType.FootLeft).P
osition.Z)
```

```
Dim aFootLKinectZ =
(playerSkeleton.Joints(JointType.FootLeft).P
osition.Z)
Dim aFootLKinectX =
(playerSkeleton.Joints(JointType.FootLeft).P
osition.X)
Dim aFootLKinectY =
(playerSkeleton.Joints(JointType.FootLeft).P
osition.Y)

Dim KneeLKinectX =
(playerSkeleton.Joints(JointType.KneeLeft).P
osition.X)
Dim KneeLKinectY =
(playerSkeleton.Joints(JointType.KneeLeft).P
osition.Y)
Dim KneeLKinectZ =
(playerSkeleton.Joints(JointType.KneeLeft).P
osition.Z)
Dim aKneeLKinectX =
(playerSkeleton.Joints(JointType.KneeLeft).P
osition.X)
Dim aKneeLKinectY =
(playerSkeleton.Joints(JointType.KneeLeft).P
osition.Y)
Dim aKneeLKinectZ =
(playerSkeleton.Joints(JointType.KneeLeft).P
osition.Z)

Dim KneeRKinectX =
(playerSkeleton.Joints(JointType.KneeRight).
Position.X)
Dim KneeRKinectY =
(playerSkeleton.Joints(JointType.KneeRight).
Position.Y)
```

```
Dim KneeRKinectZ =
(playerSkeleton.Joints(JointType.KneeRight).
Position.Z)
Dim aKneeRKinectX =
(playerSkeleton.Joints(JointType.KneeRight).
Position.X)
Dim aKneeRKinectY =
(playerSkeleton.Joints(JointType.KneeRight).
Position.Y)
Dim aKneeRKinectZ =
(playerSkeleton.Joints(JointType.KneeRight).
Position.Z)


Dim HipleftKinectX =
(playerSkeleton.Joints(JointType.HipLeft).Pos
ition.X)
Dim HipleftKinectY =
(playerSkeleton.Joints(JointType.HipLeft).Pos
ition.Y)
Dim HipleftKinectZ =
(playerSkeleton.Joints(JointType.HipLeft).Pos
ition.Z)



HipL.Text =
(playerSkeleton.Joints(JointType.HipLeft).Pos
ition.Y)

Dim HipRightKinectX =
(playerSkeleton.Joints(JointType.HipRight).Po
sition.X)
Dim HipRightKinectY =
(playerSkeleton.Joints(JointType.HipRight).Po
sition.Y)
Dim HipRightKinectZ =
(playerSkeleton.Joints(JointType.HipRight).Po
sition.Z)
                HipR.Text =
(playerSkeleton.Joints(JointType.HipRight).Po
sition.Y)

Dim HipCenterKinectX =
(playerSkeleton.Joints(JointType.HipCenter).
Position.X)
Dim HipCenterKinectY =
(playerSkeleton.Joints(JointType.HipCenter).
Position.Y)
Dim HipCenterKinectZ =
(playerSkeleton.Joints(JointType.HipCenter).
Position.Z)

Dim AnklerightKinectX =
(playerSkeleton.Joints(JointType.AnkleRight).
Position.X)
Dim AnklerightKinectY =
(playerSkeleton.Joints(JointType.AnkleRight).
Position.Y)
Dim AnklerightKinectZ =
(playerSkeleton.Joints(JointType.AnkleRight).
Position.Z)

Dim footrightKinectX =
(playerSkeleton.Joints(JointType.FootRight).P
osition.X)
```

```
Dim footrightKinectY =
(playerSkeleton.Joints(JointType.FootRight).P
osition.Y)
Dim footrightKinectZ =
(playerSkeleton.Joints(JointType.FootRight).P
osition.Z)

Dim AnkleleftKinectX =
(playerSkeleton.Joints(JointType.AnkleLeft).P
osition.X)
Dim AnkleleftKinectY =
(playerSkeleton.Joints(JointType.AnkleLeft).P
osition.Y)
Dim AnkleleftKinectZ =
(playerSkeleton.Joints(JointType.AnkleLeft).P
osition.Z)

Dim footleftKinectX =
(playerSkeleton.Joints(JointType.FootLeft).P
osition.X)
Dim footleftKinectY =
(playerSkeleton.Joints(JointType.FootLeft).P
osition.Y)
Dim footleftKinectZ =
(playerSkeleton.Joints(JointType.FootLeft).P
osition.Z)

Dim HandleftKinectX =
(playerSkeleton.Joints(JointType.HandLeft).P
osition.X)
Dim HandleftKinectY =
(playerSkeleton.Joints(JointType.HandLeft).P
osition.Y)
Dim HandleftKinectZ =
(playerSkeleton.Joints(JointType.HandLeft).P
osition.Z)

Dim HandrightKinectX =
(playerSkeleton.Joints(JointType.HandRight).
Position.X)
Dim HandrightKinectY =
(playerSkeleton.Joints(JointType.HandRight).
Position.Y)
Dim HandrightKinectZ =
(playerSkeleton.Joints(JointType.HandRight).
Position.Z)

Dim WristleftKinectX =
(playerSkeleton.Joints(JointType.WristLeft).P
osition.X)
Dim WristleftKinectY =
(playerSkeleton.Joints(JointType.WristLeft).P
osition.Y)
Dim WristleftKinectZ =
(playerSkeleton.Joints(JointType.WristLeft).P
osition.Z)

Dim WristrightKinectX =
(playerSkeleton.Joints(JointType.WristRight).
Position.X)
Dim WristrightKinectY =
(playerSkeleton.Joints(JointType.WristRight).
Position.Y)
Dim WristrightKinectZ =
(playerSkeleton.Joints(JointType.WristRight).
Position.Z)
```

```
Dim ElbowleftKinectX =
(playerSkeleton.Joints(JointType.ElbowLeft).
Position.X)
Dim ElbowleftKinectY =
(playerSkeleton.Joints(JointType.ElbowLeft).
Position.Y)
Dim ElbowleftKinectZ =
(playerSkeleton.Joints(JointType.ElbowLeft).
Position.Z)

Dim ElbowrightKinectX =
(playerSkeleton.Joints(JointType.ElbowRight)
.Position.X)
Dim ElbowrightKinectY =
(playerSkeleton.Joints(JointType.ElbowRight)
.Position.Y)
Dim ElbowrightKinectZ =
(playerSkeleton.Joints(JointType.ElbowRight)
.Position.Z)

Dim ShoulderleftKinectX =
(playerSkeleton.Joints(JointType.ShoulderLe
ft).Position.X)
Dim ShoulderleftKinectY =
(playerSkeleton.Joints(JointType.ShoulderLe
ft).Position.Y)
Dim ShoulderleftKinectZ =
(playerSkeleton.Joints(JointType.ShoulderLe
ft).Position.Z)

Dim ShoulderrightKinectX =
(playerSkeleton.Joints(JointType.ShoulderRig
ht).Position.X)
Dim ShoulderrightKinectY =
(playerSkeleton.Joints(JointType.ShoulderRig
ht).Position.Y)
Dim ShoulderrightKinectZ =
(playerSkeleton.Joints(JointType.ShoulderRig
ht).Position.Z)

Dim ShoulderCenterKinectX =
(playerSkeleton.Joints(JointType.ShoulderCe
nter).Position.X)
Dim ShoulderCenterKinectY =
(playerSkeleton.Joints(JointType.ShoulderCe
nter).Position.Y)
Dim ShoulderCenterKinectZ =
(playerSkeleton.Joints(JointType.ShoulderCe
nter).Position.Z)

Dim HipRAngleXZ = ((-1 * 180 *
Math.Atan(playerSkeleton.Joints(JointType.Hi
pRight).Position.X /
playerSkeleton.Joints(JointType.HipRight).Po
sition.Z)) / 3.14159)
Dim HipRAngleYZ = ((-1 * 180 *
Math.Atan(playerSkeleton.Joints(JointType.Hi
pRight).Position.Y /
playerSkeleton.Joints(JointType.HipRight).Po
sition.Z)) / 3.14159)
Dim HipRAngleXY = ((-1 * 180 *
Math.Atan(playerSkeleton.Joints(JointType.Hi
pRight).Position.X /
playerSkeleton.Joints(JointType.HipRight).Po
sition.Y)) / 3.14159)

)
```

```
Dim HipLAngleYZ = ((-1 * 180 *
Math.Atan(playerSkeleton.Joints(JointType.Hi
pLeft).Position.Y /
playerSkeleton.Joints(JointType.HipLeft).Posi
tion.Z)) / 3.14159)
Dim HipLAngleXZ = ((-1 * 180 *
Math.Atan(playerSkeleton.Joints(JointType.Hi
pLeft).Position.X /
playerSkeleton.Joints(JointType.HipLeft).Posi
tion.Z)) / 3.14159)
Dim HipLAngleXY = ((-1 * 180 *
Math.Atan(playerSkeleton.Joints(JointType.Hi
pLeft).Position.X /
playerSkeleton.Joints(JointType.HipLeft).Posi
tion.Y)) / 3.14159)
Dim HipCAngleXY = ((-1 * 180 *
Math.Atan(playerSkeleton.Joints(JointType.Hi
pCenter).Position.X /
playerSkeleton.Joints(JointType.HipCenter).P
osition.Y)) / 3.14159)
Dim HipCAngleXZ = ((-1 * 180 *
Math.Atan(playerSkeleton.Joints(JointType.Hi
pCenter).Position.X /
playerSkeleton.Joints(JointType.HipCenter).P
osition.Z)) / 3.14159)
Dim HipCAngleYZ = ((-1 * 180 *
Math.Atan(playerSkeleton.Joints(JointType.Hi
pCenter).Position.Y /
playerSkeleton.Joints(JointType.HipCenter).P
osition.Z)) / 3.14159)

Dim ABVecterX = ((HipleftKinectX -
KneeLKinectX) * (FootLKinectX -
KneeLKinectX)) + ((HipleftKinectY -
KneeLKinectY) * (FootLKinectY -
KneeLKinectY)) + ((HipleftKinectZ -
KneeLKinectZ) * (FootLKinectZ -
KneeLKinectZ))

Dim ABVecter = (((((HipleftKinectX -
KneeLKinectX) ^ 2) + ((HipleftKinectY -
KneeLKinectY) ^ 2) + ((HipleftKinectZ -
KneeLKinectZ) ^ 2)) ^ 0.5) * ((((FootLKinectX
- KneeLKinectX) ^ 2) + ((FootLKinectY -
KneeLKinectY) ^ 2) + ((FootLKinectZ -
KneeLKinectZ) ^ 2)) ^ 0.5))
Dim s1 = CInt((175 - (180 *
Math.Acos((ABVecterX) / (ABVecter))) /
3.14159))
Dim s2 = CInt((175 - (180 *
Math.Acos((ABVecterX) / (ABVecter))) /
3.14159))
Dim s3 = CInt((175 - (180 *
Math.Acos((ABVecterX) / (ABVecter))) /
3.14159))
Dim s4 = CInt((175 - (180 *
Math.Acos((ABVecterX) / (ABVecter))) /
3.14159))
Dim s5 = CInt((175 - (180 *
Math.Acos((ABVecterX) / (ABVecter))) /
3.14159))
            TextBox101.Text = (((s1) + (s2) +
(s3) + (s4) + (s5)) / 5)


Dim ABVecterR = ((HipRightKinectX -
KneeRKinectX) * (FootRKinectX -
KneeRKinectX)) + ((HipRightKinectY -
KneeRKinectY) * (FootRKinectY -
KneeRKinectY)) + ((HipRightKinectZ -
```

```
KneeRKinectZ) * (FootRKinectZ -
KneeRKinectZ))

Dim AB2Vecter = (((((HipRightKinectX -
KneeRKinectX) ^ 2) + ((HipRightKinectY -
KneeRKinectY) ^ 2) + ((HipRightKinectZ -
KneeRKinectZ) ^ 2)) ^ 0.5) * ((((FootRKinectX
- KneeRKinectX) ^ 2) + ((FootRKinectY -
KneeRKinectY) ^ 2) + ((FootRKinectZ -
KneeRKinectZ) ^ 2)) ^ 0.5))
Dim s6 = CInt((175 - (180 *
Math.Acos((ABVecterR) / (AB2Vecter))) /
3.14159))
Dim s7 = CInt((175 - (180 *
Math.Acos((ABVecterR) / (AB2Vecter))) /
3.14159))
Dim s8 = CInt((175 - (180 *
Math.Acos((ABVecterR) / (AB2Vecter))) /
3.14159))
Dim s9 = CInt((175 - (180 *
Math.Acos((ABVecterR) / (AB2Vecter))) /
3.14159))
Dim s10 = CInt((175 - (180 *
Math.Acos((ABVecterR) / (AB2Vecter))) /
3.14159))

            TextBox102.Text = (((s6) + (s7) +
(s8) + (s9) + (s10)) / 5)

Dim HipleftVecter = ((HipCenterKinectX -
HipleftKinectX) * (HipRightKinectX -
HipleftKinectX)) + ((HipCenterKinectY -
HipleftKinectY) * (HipRightKinectY -
HipleftKinectY)) + ((HipCenterKinectZ -
HipleftKinectZ) * (HipRightKinectZ -
HipleftKinectZ))

Dim scaleHipleft = (((((HipCenterKinectX -
HipleftKinectX) ^ 2) + ((HipCenterKinectY -
HipleftKinectY) ^ 2) + ((HipCenterKinectZ -
HipleftKinectZ) ^ 2)) ^ 0.5) *
((((HipRightKinectX - HipleftKinectX) ^ 2) +
((HipRightKinectY - HipleftKinectY) ^ 2) +
((HipRightKinectZ - HipleftKinectZ) ^ 2)) ^
0.5))

            TextBox103.Text = CInt((180 *
Math.Acos((HipleftVecter) / (scaleHipleft))) /
3.14159)

Dim HiprightVecter = ((HipCenterKinectX -
HipRightKinectX) * (HipleftKinectX -
HipRightKinectX)) + ((HipCenterKinectY -
HipRightKinectY) * (HipleftKinectY -
HipRightKinectY)) + ((HipCenterKinectZ -
HipRightKinectZ) * (HipleftKinectZ -
HipRightKinectZ))

Dim scaleHipright = (((((HipCenterKinectX -
HipRightKinectX) ^ 2) + ((HipCenterKinectY -
HipRightKinectY) ^ 2) + ((HipCenterKinectZ -
HipRightKinectZ) ^ 2)) ^ 0.5) *
((((HipleftKinectX - HipRightKinectX) ^ 2) +
((HipleftKinectY - HipRightKinectY) ^ 2) +
((HipleftKinectZ - HipRightKinectZ) ^ 2)) ^
0.5))
```

```
            TextBox104.Text = CInt((180 *
Math.Acos((HiprightVecter) / (scaleHipright)))
/ 3.14159)

Dim HipCenterVecter = ((HipleftKinectX -
HipCenterKinectX) * (HipCenterKinectX -
HipRightKinectX)) + ((HipleftKinectY -
HipCenterKinectY) * (HipCenterKinectY -
HipRightKinectY)) + ((HipleftKinectZ -
HipCenterKinectZ) * (HipCenterKinectZ -
HipRightKinectZ))

Dim scaleHipCenter = (((((HipleftKinectX -
HipCenterKinectX) ^ 2) + ((HipleftKinectY -
HipCenterKinectY) ^ 2) + ((HipleftKinectZ -
HipCenterKinectZ) ^ 2)) ^ 0.5) *
((((HipRightKinectX - HipCenterKinectX) ^ 2)
+ ((HipRightKinectY - HipCenterKinectY) ^ 2)
+ ((HipRightKinectZ - HipCenterKinectZ) ^ 2))
^ 0.5))


Dim s11 = CInt(90 - ((180 *
Math.Acos((HipCenterVecter) /
(scaleHipCenter))) / 3.14159))
Dim s12 = CInt(90 - ((180 *
Math.Acos((HipCenterVecter) /
(scaleHipCenter))) / 3.14159))
Dim s13 = CInt(90 - ((180 *
Math.Acos((HipCenterVecter) /
(scaleHipCenter))) / 3.14159))
Dim s14 = CInt(90 - ((180 *
Math.Acos((HipCenterVecter) /
(scaleHipCenter))) / 3.14159))
Dim s15 = CInt(90 - ((180 *
Math.Acos((HipCenterVecter) /
(scaleHipCenter))) / 3.14159))

            TextBox105.Text = (((s11) + (s12)
+ (s13) + (s14) + (s15)) / 5)

Dim AnklerightVecter = ((KneeRKinectX -
AnklerightKinectX) * (AnklerightKinectX -
footrightKinectX)) + ((KneeRKinectY -
AnklerightKinectY) * (AnklerightKinectY -
footrightKinectY)) + ((KneeRKinectZ -
AnklerightKinectZ) * (AnklerightKinectZ -
footrightKinectZ))

Dim scaleAnkleright = (((((KneeRKinectX -
AnklerightKinectX) ^ 2) + ((KneeRKinectY -
AnklerightKinectY) ^ 2) + ((KneeRKinectZ -
AnklerightKinectZ) ^ 2)) ^ 0.5) *
((((footrightKinectX - AnklerightKinectX) ^ 2)
+ ((footrightKinectY - AnklerightKinectY) ^ 2)
+ ((footrightKinectZ - AnklerightKinectZ) ^
2)) ^ 0.5))

Dim s16 = CInt((180 *
Math.Acos((AnklerightVecter) /
(scaleAnkleright))) / 3.14159)
Dim s17 = CInt((180 *
Math.Acos((AnklerightVecter) /
(scaleAnkleright))) / 3.14159)
Dim s18 = CInt((180 *
Math.Acos((AnklerightVecter) /
(scaleAnkleright))) / 3.14159)
```

```
Dim s19 = CInt((180 *
Math.Acos((AnklerightVecter) /
(scaleAnkleright))) / 3.14159)
Dim s20 = CInt((180 *
Math.Acos((AnklerightVecter) /
(scaleAnkleright))) / 3.14159)

            TextBox107.Text = 45 - (((s16) +
(s17) + (s18) + (s19) + (s20)) / 5)

Dim AnkleleftVecter = ((KneeLKinectX -
AnkleleftKinectX) * (AnkleleftKinectX -
footleftKinectX)) + ((KneeLKinectY -
AnkleleftKinectY) * (AnkleleftKinectY -
footleftKinectY)) + ((KneeLKinectZ -
AnkleleftKinectZ) * (AnkleleftKinectZ -
footleftKinectZ))

Dim scaleAnkleleft = (((((KneeLKinectX -
AnkleleftKinectX) ^ 2) + ((KneeLKinectY -
AnkleleftKinectY) ^ 2) + ((KneeRKinectZ -
AnkleleftKinectZ) ^ 2)) ^ 0.5) *
((((footleftKinectX - AnkleleftKinectX) ^ 2) +
((footleftKinectY - AnkleleftKinectY) ^ 2) +
((footleftKinectZ - AnkleleftKinectZ) ^ 2)) ^
0.5))

Dim s21 = CInt((180 *
Math.Acos((AnkleleftVecter) /
(scaleAnkleleft))) / 3.14159)
Dim s22 = CInt((180 *
Math.Acos((AnkleleftVecter) /
(scaleAnkleleft))) / 3.14159)
Dim s23 = CInt((180 *
Math.Acos((AnkleleftVecter) /
(scaleAnkleleft))) / 3.14159)
Dim s24 = CInt((180 *
Math.Acos((AnkleleftVecter) /
(scaleAnkleleft))) / 3.14159)
Dim s25 = CInt((180 *
Math.Acos((AnkleleftVecter) /
(scaleAnkleleft))) / 3.14159)

            TextBox106.Text = 45 - (((s21) +
(s22) + (s23) + (s24) + (s25)) / 5)


Dim wristrightVecter = ((HandrightKinectX -
WristrightKinectX) * (WristrightKinectX -
ElbowrightKinectX)) + ((HandrightKinectY -
WristrightKinectY) * (WristrightKinectY -
ElbowrightKinectY)) + ((HandrightKinectZ -
WristrightKinectZ) * (WristrightKinectZ -
ElbowrightKinectZ))

Dim scalewristright = (((((HandrightKinectX -
WristrightKinectX) ^ 2) + ((HandrightKinectY -
WristrightKinectY) ^ 2) + ((HandrightKinectZ -
WristrightKinectZ) ^ 2)) ^ 0.5) *
((((ElbowrightKinectX - WristrightKinectX) ^ 2)
+ ((ElbowrightKinectY - WristrightKinectY) ^
2) + ((ElbowrightKinectZ - WristrightKinectZ)
^ 2)) ^ 0.5))

Dim s26 = CInt((180 *
Math.Acos((wristrightVecter) /
(scalewristright))) / 3.14159)
```

```
Dim s27 = CInt((180 *
Math.Acos((wristrightVecter) /
(scalewristright))) / 3.14159)
Dim s28 = CInt((180 *
Math.Acos((wristrightVecter) /
(scalewristright))) / 3.14159)
Dim s29 = CInt((180 *
Math.Acos((wristrightVecter) /
(scalewristright))) / 3.14159)
Dim s30 = CInt((180 *
Math.Acos((wristrightVecter) /
(scalewristright))) / 3.14159)

            TextBox109.Text = (((s26) + (s27)
+ (s28) + (s29) + (s30)) / 5)

Dim wristleftVecter = ((HandleftKinectX -
WristleftKinectX) * (WristleftKinectX -
ElbowleftKinectX)) + ((HandleftKinectY -
WristleftKinectY) * (WristleftKinectY -
ElbowleftKinectY)) + ((HandleftKinectZ -
WristleftKinectZ) * (WristleftKinectZ -
ElbowleftKinectZ))

Dim scalewristleft = (((((HandleftKinectX -
WristleftKinectX) ^ 2) + ((HandleftKinectY -
WristleftKinectY) ^ 2) + ((HandleftKinectZ -
WristleftKinectZ) ^ 2)) ^ 0.5) *
((((ElbowleftKinectX - WristleftKinectX) ^ 2)
+ ((ElbowleftKinectY - WristleftKinectY) ^ 2)
+ ((ElbowleftKinectZ - WristleftKinectZ) ^ 2))
^ 0.5))

Dim s31 = CInt((180 *
Math.Acos((wristleftVecter) / (scalewristleft)))
/ 3.14159)
Dim s32 = CInt((180 *
Math.Acos((wristleftVecter) / (scalewristleft)))
/ 3.14159)
Dim s33 = CInt((180 *
Math.Acos((wristleftVecter) / (scalewristleft)))
/ 3.14159)
Dim s34 = CInt((180 *
Math.Acos((wristleftVecter) / (scalewristleft)))
/ 3.14159)
Dim s35 = CInt((180 *
Math.Acos((wristleftVecter) / (scalewristleft)))
/ 3.14159)

            TextBox108.Text = (((s31) + (s32)
+ (s33) + (s34) + (s35)) / 5)

Dim ElbowrightVecter = ((WristrightKinectX -
ElbowrightKinectX) * (ElbowrightKinectX -
ShoulderrightKinectX)) + ((WristrightKinectY -
ElbowrightKinectY) * (ElbowrightKinectY -
ShoulderrightKinectY)) + ((WristrightKinectZ -
ElbowrightKinectZ) * (ElbowrightKinectZ -
ShoulderrightKinectZ))

Dim scaleElbowright = (((((WristrightKinectX -
ElbowrightKinectX) ^ 2) + ((WristrightKinectY
- ElbowrightKinectY) ^ 2) +
((WristrightKinectZ - ElbowrightKinectZ) ^ 2))
^ 0.5) * ((((ShoulderrightKinectX -
ElbowrightKinectX) ^ 2) +
((ShoulderrightKinectY - ElbowrightKinectY) ^
```

```
2) + ((ShoulderrightKinectZ -
ElbowrightKinectZ) ^ 2)) ^ 0.5))

Dim s36 = CInt(((180 *
Math.Acos((ElbowrightVecter) /
(scaleElbowright))) / 3.14159))
Dim s37 = CInt(((180 *
Math.Acos((ElbowrightVecter) /
(scaleElbowright))) / 3.14159))
Dim s38 = CInt(((180 *
Math.Acos((ElbowrightVecter) /
(scaleElbowright))) / 3.14159))
Dim s39 = CInt(((180 *
Math.Acos((ElbowrightVecter) /
(scaleElbowright))) / 3.14159))
Dim s40 = CInt(((180 *
Math.Acos((ElbowrightVecter) /
(scaleElbowright))) / 3.14159))

            TextBox111.Text = (((s36) + (s37)
+ (s38) + (s39) + (s40)) / 5)

Dim ElbowleftVecter = ((WristleftKinectX -
ElbowleftKinectX) * (ElbowleftKinectX -
ShoulderleftKinectX)) + ((WristleftKinectY -
ElbowleftKinectY) * (ElbowleftKinectY -
ShoulderleftKinectY)) + ((WristleftKinectZ -
ElbowleftKinectZ) * (ElbowleftKinectZ -
ShoulderleftKinectZ))

Dim scaleElbowleft = (((((WristleftKinectX -
ElbowleftKinectX) ^ 2) + ((WristleftKinectY -
ElbowleftKinectY) ^ 2) + ((WristleftKinectZ -
ElbowleftKinectZ) ^ 2)) ^ 0.5) *
((((ShoulderleftKinectX - ElbowleftKinectX) ^
2) + ((ShoulderleftKinectY -
ElbowleftKinectY) ^ 2) +
((ShoulderleftKinectZ - ElbowleftKinectZ) ^
2)) ^ 0.5))

Dim s41 = CInt(((180 *
Math.Acos((ElbowleftVecter) /
(scaleElbowleft))) / 3.14159))
Dim s42 = CInt(((180 *
Math.Acos((ElbowleftVecter) /
(scaleElbowleft))) / 3.14159))
Dim s43 = CInt(((180 *
Math.Acos((ElbowleftVecter) /
(scaleElbowleft))) / 3.14159))
Dim s44 = CInt(((180 *
Math.Acos((ElbowleftVecter) /
(scaleElbowleft))) / 3.14159))
Dim s45 = CInt(((180 *
Math.Acos((ElbowleftVecter) /
(scaleElbowleft))) / 3.14159))

            TextBox110.Text = (((s41) + (s42)
+ (s43) + (s44) + (s45)) / 5)

Dim ShoulderrightVecter =
((ElbowrightKinectX - ShoulderrightKinectX) *
(ShoulderrightKinectX - HipRightKinectX)) +
((ElbowrightKinectY - ShoulderrightKinectY) *
(ShoulderrightKinectY - HipRightKinectY)) +
((ElbowrightKinectZ - ShoulderrightKinectZ) *
(ShoulderrightKinectZ - HipRightKinectZ))

Dim scaleShoulderright =
(((((ElbowrightKinectX -
```

```
ShoulderrightKinectX) ^ 2) +
((ElbowrightKinectY - ShoulderrightKinectY) ^
2) + ((ElbowrightKinectZ -
ShoulderrightKinectZ) ^ 2)) ^ 0.5) *
((((ShoulderrightKinectX - HipRightKinectX) ^
2) + ((ShoulderrightKinectY -
HipRightKinectY) ^ 2) +
((ShoulderrightKinectZ - HipRightKinectZ) ^
2)) ^ 0.5))

Dim s46 = CInt((180 *
Math.Acos((ShoulderrightVecter) /
(scaleShoulderright))) / 3.14159)
Dim s47 = CInt((180 *
Math.Acos((ShoulderrightVecter) /
(scaleShoulderright))) / 3.14159)
Dim s48 = CInt((180 *
Math.Acos((ShoulderrightVecter) /
(scaleShoulderright))) / 3.14159)
Dim s49 = CInt((180 *
Math.Acos((ShoulderrightVecter) /
(scaleShoulderright))) / 3.14159)
Dim s50 = CInt((180 *
Math.Acos((ShoulderrightVecter) /
(scaleShoulderright))) / 3.14159)


            TextBox113.Text = (190 - (((s46)
+ (s47) + (s48) + (s49) + (s50)) / 5))

Dim ShoulderleftVecter = ((ElbowleftKinectX
- ShoulderleftKinectX) *
(ShoulderleftKinectX - HipleftKinectX)) +
((ElbowleftKinectY - ShoulderleftKinectY) *
(ShoulderleftKinectY - HipleftKinectY)) +
((ElbowleftKinectZ - ShoulderleftKinectZ) *
(ShoulderleftKinectZ - HipleftKinectZ))

Dim scaleShoulderleft =
(((((ElbowleftKinectX - ShoulderleftKinectX)
^ 2) + ((ElbowleftKinectY -
ShoulderleftKinectY) ^ 2) +
((ElbowleftKinectZ - ShoulderleftKinectZ) ^
2)) ^ 0.5) * ((((ShoulderleftKinectX -
HipleftKinectX) ^ 2) + ((ShoulderleftKinectY -
HipleftKinectY) ^ 2) + ((ShoulderleftKinectZ -
HipleftKinectZ) ^ 2)) ^ 0.5))

Dim s51 = CInt((180 *
Math.Acos((ShoulderleftVecter) /
(scaleShoulderleft))) / 3.14159)
Dim s52 = CInt((180 *
Math.Acos((ShoulderleftVecter) /
(scaleShoulderleft))) / 3.14159)
Dim s53 = CInt((180 *
Math.Acos((ShoulderleftVecter) /
(scaleShoulderleft))) / 3.14159)
Dim s54 = CInt((180 *
Math.Acos((ShoulderleftVecter) /
(scaleShoulderleft))) / 3.14159)
Dim s55 = CInt((180 *
Math.Acos((ShoulderleftVecter) /
(scaleShoulderleft))) / 3.14159)


            TextBox112.Text = (190 - (((s51)
+ (s52) + (s53) + (s54) + (s55)) / 5))


Dim SpineVecter = ((ShoulderCenterKinectX
- spineKinectX) * (spineKinectX -
```

```
HipCenterKinectX) + (ShoulderCenterKinectY
- spineKinectY) * (spineKinectY -
HipCenterKinectY) + (ShoulderCenterKinectZ
- spineKinectZ) * (spineKinectZ -
HipCenterKinectZ))
Dim scaleSpine = (((((ShoulderCenterKinectX
- spineKinectX) ^ 2) +
((ShoulderCenterKinectY - spineKinectY) ^ 2)
+ ((ShoulderCenterKinectZ - spineKinectZ) ^
2)) ^ 0.5) * ((((spineKinectX -
HipCenterKinectX) ^ 2) + ((spineKinectY -
HipCenterKinectY) ^ 2) + ((spineKinectZ -
HipCenterKinectZ) ^ 2)) ^ 0.5))
            TextBox114.Text = CInt(50 - (180
* Math.Acos((SpineVecter) / (scaleSpine))) /
3.14159)


If ((TextBox101.Text) - (0)) <= 0 Then
            KneeLAngle.Text = 0
ElseIf ((TextBox101.Text) - (0)) >= 180 Then
            KneeLAngle.Text = 180
Else
            KneeLAngle.Text =
TextBox101.Text
EndIf

If ((TextBox102.Text) - (0)) <= 0 Then
            KneeRAngle.Text = 0
ElseIf ((TextBox102.Text) - (0)) >= 180 Then
            KneeRAngle.Text = 180
Else
            KneeRAngle.Text =
TextBox102.Text
EndIf

If ((TextBox103.Text) - (0)) <= 0 Then
            Hipleftangle.Text = 0
ElseIf ((TextBox103.Text) - (0)) >= 180 Then
            Hipleftangle.Text = 180
Else
            Hipleftangle.Text =
TextBox103.Text
EndIf

If ((TextBox104.Text) - (0)) <= 0 Then
            HipRightangle.Text = 0
ElseIf ((TextBox104.Text) - (0)) >= 180 Then
            HipRightangle.Text = 180
Else
            HipRightangle.Text =
TextBox104.Text
EndIf

If ((TextBox105.Text) - (0)) <= 0 Then
            HipCenterangle.Text = 0
ElseIf ((TextBox105.Text) - (0)) >= 180 Then
            HipCenterangle.Text = 180
Else
            HipCenterangle.Text =
TextBox105.Text
EndIf

If ((TextBox106.Text) - (0)) <= 0 Then
            Ankleleftangle.Text = 0
ElseIf ((TextBox106.Text) - (0)) >= 180 Then
            Ankleleftangle.Text = 180
Else
            Ankleleftangle.Text =
TextBox106.Text
```

```
EndIf

If ((TextBox107.Text) - (0)) <= 0 Then
                Anklerightangle.Text = 0
ElseIf ((TextBox107.Text) - (0)) >= 180 Then
                Anklerightangle.Text = 180
Else
                Anklerightangle.Text =
TextBox107.Text
EndIf

If ((TextBox108.Text) - (0)) <= 0 Then
                HandLeftangle.Text = 0
ElseIf ((TextBox108.Text) - (0)) >= 180 Then
                HandLeftangle.Text = 180
Else
                HandLeftangle.Text =
TextBox108.Text
EndIf

If ((TextBox109.Text) - (0)) <= 0 Then
                HandRightangle.Text = 0
ElseIf ((TextBox109.Text) - (0)) >= 180 Then
                HandRightangle.Text = 180
Else
                HandRightangle.Text =
TextBox109.Text
EndIf

If ((TextBox110.Text) - (0)) <= 0 Then
                ElbowLeftangle.Text = 0
ElseIf ((TextBox110.Text) - (0)) >= 180 Then
                ElbowLeftangle.Text = 180
Else
                ElbowLeftangle.Text =
TextBox110.Text
EndIf

If ((TextBox111.Text) - (0)) <= 0 Then
                ElbowRightangle.Text = 0
ElseIf ((TextBox111.Text) - (0)) >= 180 Then
                ElbowRightangle.Text = 180
Else
                ElbowRightangle.Text =
TextBox111.Text
EndIf

If ((TextBox112.Text) - (0)) <= 0 Then
                Shoulderleftangle.Text = 0
ElseIf ((TextBox112.Text) - (0)) >= 180 Then
                Shoulderleftangle.Text = 180
Else
                Shoulderleftangle.Text =
TextBox112.Text
EndIf

If ((TextBox113.Text) - (0)) <= 0 Then
                ShoulderRightangle.Text = 0
ElseIf ((TextBox113.Text) - (0)) >= 180 Then
                ShoulderRightangle.Text =
180
Else
                ShoulderRightangle.Text =
TextBox113.Text
EndIf

If ((TextBox114.Text) - (0)) <= 0 Then
                ShoulderCenterangle.Text = 0
ElseIf ((TextBox114.Text) - (0)) >= 180 Then
```

```
                ShoulderCenterangle.Text =
180
Else
                ShoulderCenterangle.Text =
TextBox114.Text
EndIf


Dim i = 1, bb = 2
Dim strout AsString
If bb = save.Text Then
strout =
DateTime.Now.TimeOfDay.Seconds.ToString

DoUntil i = 2


Dim con AsNewSqlConnection(constr)



con.Open()
Dim mySQL AsString
                    mySQL = "insert into
vb(ID,date,Time,ShoulderC,ShoulderL,Shoul
derR,ElbowL,ElbowR,HipL,HipR,kneeL,kneeR,
AnkleL,AnkleR,HandL,HandR,HipC,PositionHi
pL,PositionHipR)
values(@ID,@da,@Ti,@SC,@SL,@SR,@EL,@ER,
@HL,@HR,@kL,@kR,@AL,@AR,@haL,@haR,@H
C,@PHL,@PHR)"
Dim cmd AsNewSqlCommand(mySQL, con)

cmd.Parameters.Add("ID",
System.Data.SqlDbType.NVarChar).Value =
TextBox8.Text
cmd.Parameters.Add("da",
System.Data.SqlDbType.NVarChar).Value =
DateTime.Now.ToLongDateString.ToString
cmd.Parameters.Add("Ti",
System.Data.SqlDbType.NVarChar).Value =
DateTime.Now.ToLongTimeString.ToString
cmd.Parameters.Add("SC",
System.Data.SqlDbType.NVarChar).Value =
ShoulderCenterangle.Text
cmd.Parameters.Add("SL",
System.Data.SqlDbType.NVarChar).Value =
Shoulderleftangle.Text
cmd.Parameters.Add("SR",
System.Data.SqlDbType.NVarChar).Value =
ShoulderRightangle.Text
cmd.Parameters.Add("EL",
System.Data.SqlDbType.NVarChar).Value =
ElbowLeftangle.Text
cmd.Parameters.Add("ER",
System.Data.SqlDbType.NVarChar).Value =
ElbowRightangle.Text
cmd.Parameters.Add("HL",
System.Data.SqlDbType.NVarChar).Value =
Hipleftangle.Text
cmd.Parameters.Add("HR",
System.Data.SqlDbType.NVarChar).Value =
HipRightangle.Text
cmd.Parameters.Add("kL",
System.Data.SqlDbType.NVarChar).Value =
KneeLAngle.Text
cmd.Parameters.Add("kR",
System.Data.SqlDbType.NVarChar).Value =
KneeRAngle.Text
```

```
cmd.Parameters.Add("AL",
System.Data.SqlDbType.NVarChar).Value =
Ankleleftangle.Text
cmd.Parameters.Add("AR",
System.Data.SqlDbType.NVarChar).Value =
Anklerightangle.Text
cmd.Parameters.Add("haL",
System.Data.SqlDbType.NVarChar).Value =
HandLeftangle.Text
cmd.Parameters.Add("haR",
System.Data.SqlDbType.NVarChar).Value =
HandRightangle.Text
cmd.Parameters.Add("HC",
System.Data.SqlDbType.NVarChar).Value =
HipCenterangle.Text
cmd.Parameters.Add("PHL",
System.Data.SqlDbType.NVarChar).Value =
HipL.Text
cmd.Parameters.Add("PHR",
System.Data.SqlDbType.NVarChar).Value =
HipR.Text

                    i = (i + 1)
cmd.ExecuteNonQuery()
con.Close()
Loop

EndIf
EndIf

EndSub

PrivateSubSetEllipsePosition(ByVal ellipse
AsFrameworkElement, ByVal kjoint AsJoint,
ByVal JID AsJointType)

Dim scaledJoint =
kjoint.ScaleTo(ScreenMaxX, ScreenMaxY,
0.5F, 0.2F)

Canvas.SetLeft(ellipse,
scaledJoint.Position.X)
Canvas.SetTop(ellipse,
scaledJoint.Position.Y)

EndSub

PrivateSub Window_Closed(ByVal sender
AsObject, ByVal e AsEventArgs)
sensor.Stop()
sensor.Dispose()


EndSub

PrivateSub
DiscoverKinectSensor()
ForEach iterSensor
AsKinectSensorInKinectSensor.KinectSensors
If iterSensor.Status =
KinectStatus.Connected Then
sensor = iterSensor
Exit For
EndIf
Next

If sensor IsNothingThen
MsgBox("Could not find any valid Kinect
Connected. Please restart the program",
vbCritical)
```

```
Exit Sub
EndIf

SelectCase sensor.Status
CaseKinectStatus.Connected

CaseKinectStatus.Disconnected
MsgBox("Kinect is Disconnected",
vbExclamation)


CaseKinectStatus.NotPowered
MsgBox("Kinect is Not Powered. Connect to
Power", vbExclamation)

Case Else
MsgBox("Unkown Kinect Status",
vbExclamation)
EndSelect

If (sensor.Status = KinectStatus.Connected)
Then
InitializeKinect()
EndIf
EndSub

PrivateSubInitializeKinect()

Dim parameters =
NewTransformSmoothParametersWith
{.Smoothing = 0.8F, .Correction = 0.3F,
.Prediction = 0.4F, .JitterRadius = 1.0F,
.MaxDeviationRadius = 0.5F}
sensor.SkeletonStream.Enable(parameters)
sensor.Start()
EndSub

PrivateFunctionHipCenterEllipse()
AsFrameworkElement
EndFunction

PrivateSub RadioButton1_Checked(ByVal
sender As System.Object, ByVal e As
System.Windows.RoutedEventArgs)

EndSub

PrivateSubTouchFrameEventHandler(ByVal
sender AsObject, ByVal e
AsColorImageFrameReadyEventArgs)
ThrowNewNotImplementedException
EndSub

PrivateSub
Shoulderleftangle_TextChanged(ByVal
sender As System.Object, ByVal e As
System.Windows.Controls.TextChangedEven
tArgs) Handles
Shoulderleftangle.TextChanged

EndSub


PrivateSub save_load() HandlesMe.Loaded
Dim strout AsString = ""
strout =
DateTime.Now.TimeOfDay.Minutes.ToString
Dim i = 1
Dim con AsNewSqlConnection(constr)
```

```
If strout = 1 Then
con.Open()
Dim mySQL AsString
mySQL = "insert into history(name)
values(@da)"
Dim cmd AsNewSqlCommand(mySQL, con)
cmd.Parameters.Add("da",
System.Data.SqlDbType.NVarChar).Value =
DateTime.Now.ToShortTimeString.ToString
cmd.ExecuteNonQuery()

con.Close()
EndIf



EndSub



PrivateSub Button1_Click_1(ByVal sender As
System.Object, ByVal e As
System.Windows.RoutedEventArgs) Handles
Button1.Click
Dim strout AsString
strout =
DateTime.Now.TimeOfDay.Seconds.ToString

Dim con AsNewSqlConnection(constr)
con.Open()
Dim mySQL AsString
        mySQL = "insert into
History(ID,name,lastname,Age,sex,Weight,Hei
ght,symptom)
values(@ID,@na,@la,@ag,@se,@we,@he,@sy)"
Dim cmd AsNewSqlCommand(mySQL, con)

cmd.Parameters.Add("ID",
System.Data.SqlDbType.NChar).Value =
TextBox8.Text
cmd.Parameters.Add("na",
System.Data.SqlDbType.NChar).Value =
TextBox1.Text
cmd.Parameters.Add("la",
System.Data.SqlDbType.NChar).Value =
TextBox2.Text
cmd.Parameters.Add("ag",
System.Data.SqlDbType.NChar).Value =
TextBox3.Text
cmd.Parameters.Add("se",
System.Data.SqlDbType.NChar).Value =
TextBox4.Text
cmd.Parameters.Add("we",
System.Data.SqlDbType.NChar).Value =
TextBox5.Text
cmd.Parameters.Add("he",
System.Data.SqlDbType.NChar).Value =
TextBox6.Text
cmd.Parameters.Add("sy",
System.Data.SqlDbType.NVarChar).Value =
TextBox7.Text


cmd.ExecuteNonQuery()
con.Close()
```

```
EndSub


Dim frmuse AsWindow2 = NewWindow2

PrivateSub Button2_Click(ByVal sender As
System.Object, ByVal e As
System.Windows.RoutedEventArgs) Handles
Button2.Click


frmuse.ShowDialog()


EndSub

PrivateSub Button3_Click(ByVal sender As
System.Object, ByVal e As
System.Windows.RoutedEventArgs) Handles
Button3.Click
        save.Text = 2

EndSub


PrivateSub TextBox8_TextChanged(ByVal
sender As System.Object, ByVal e As
System.Windows.Controls.TextChangedEven
tArgs) Handles TextBox8.TextChanged
EndSub

PrivateSub Button4_Click(ByVal sender As
System.Object, ByVal e As
System.Windows.RoutedEventArgs) Handles
Button4.Click
        save.Text = 1
EndSub


PrivateSub VideoImage_ImageFailed(ByVal
sender As System.Object, ByVal e As
System.Windows.ExceptionRoutedEventArgs)
Handles VideoImage.ImageFailed

EndSub


PrivateSub HipL_TextChanged(ByVal sender
As System.Object, ByVal e
AsSystem.Windows.Controls.TextChangedEv
entArgs) Handles HipL.TextChanged

EndSub
EndClass



ClassMainWindow
PrivateSub MainWindow_Closed(ByVal
sender AsObject, ByVal e As
System.EventArgs) HandlesMe.Closed
If Sensor IsNotNothingThen
If Sensor.IsRunning Then
Sensor.Stop()
Sensor.Dispose()
EndIf
EndIf
EndSub
```

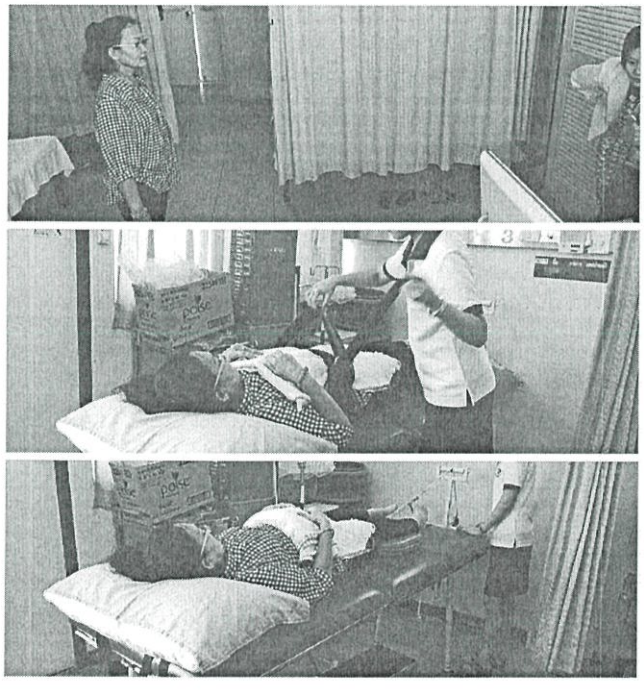
```
PrivateSub MainWindow_Loaded(ByVal
sender AsObject, ByVal e As
System.Windows.RoutedEventArgs)
HandlesMe.Loaded

IfKinectSensor.KinectSensors.Any() Then
                Sensor =
KinectSensor.KinectSensors(0)
If Sensor.Status = KinectStatus.Connected
Then
AddHandler Sensor.ColorFrameReady,
AddressOf ColorFrameReadyHandler
Sensor.ColorStream.Enable(ColorImageForm
at.RgbResolution640x480Fps30)
Try
Sensor.Start()
Catch ex AsException

EndTry
EndIf

If Sensor.IsRunning Then
                sldElevation.Minimum =
Sensor.MinElevationAngle
                sldElevation.Maximum =
Sensor.MaxElevationAngle
                sldElevation.Value =
Sensor.ElevationAngle
EndIf
EndIfEndSub

PrivateSubColorFrameReadyHandler(ByVal
sender AsObject, ByVal e
AsColorImageFrameReadyEventArgs)

Using colorImageFrame =
e.OpenColorImageFrame()
If colorImageFrame IsNotNothingThen
DimimageArray() =
NewByte(colorImageFrame.PixelDataLength -
1) {}
colorImageFrame.CopyPixelDataTo(imageArr
ay)
Dim stride = colorImageFrame.Width *
colorImageFrame.BytesPerPixel
Me.VideoImage.Source =
BitmapSource.Create(colorImageFrame.Widt
h, colorImageFrame.Height,

96, 96,
PixelFormats.Bgr32, Nothing,
imageArray, stride)
EndIf
EndUsing
EndSub

PrivateSub btnElevation_Click(ByVal sender
As System.Object, ByVal e As
System.Windows.RoutedEventArgs)
If Sensor.IsRunning Then
Dim value = sldElevation.Value
                Sensor.ElevationAngle =
CInt(value)
EndIf
EndSub
EndClass

EndNamespace
```

ภาคผนวค ข.

รูปภาพการทำกายภาพ

รูปการดำเนินงาน ณ โรงพยาบาลการุณเวช

รูปแสดงการกายที่มีการปวดบริเวณสะโพกซ้าย



รูปการกายภาพบำบัดผู้ป่วย

รูปรวมการทำงานร่วมกับนักกายภาพบำบัด