

รายงานโครงการวิจัยประจำปีงบประมาณ 2545

เรื่อง

การศึกษาอัลกอริทึมในการคำนวณค่าระดับการซูมของกล้องถ่ายภาพวีดีโอ
A Study of Algorithms for Video Camera Zooming-Level Estimation

โดย
ผศ. ดร. นพพร โชติกกำธร
และ
นาย วิชิต นันทรัตนพงศ์

RCH
TR
889
M1768

เลขหมู่.....
เลขทะเบียน.....50564
วัน,เดือน,ปี 17 พ.ค. 2547

คณะเทคโนโลยีสารสนเทศ

b.113466.6.8
i.....

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง กรุงเทพฯ 10520

พ.ศ. 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทคัดย่อ

ในการแทรกวัตถุเสมือน (virtual object) ลงในภาพจริง โดยเฉพาะภาพเคลื่อนไหว เช่น ภาพวิดีโอ สิ่งสำคัญที่จะทำให้ภาพวิดีโอที่ปรากฏมีความสมจริงคือตำแหน่งของกล้องเมื่อเปรียบเทียบกับตำแหน่งของวัตถุเสมือนในภาพ และค่าระดับการซูมหรือความยาวโฟกัสของกล้อง โดย ทั้ง 2 องค์ประกอบนี้มีผลต่อขนาดและมุมมองของวัตถุจริงในภาพ ดังนั้นเพื่อให้การแทรกวัตถุเสมือนลงในภาพจริงมีความสมจริงเป็นธรรมชาติ จำเป็นต้องทราบข้อมูลเหล่านี้เพื่อใช้ในการสร้างภาพ (rendering) วัตถุเสมือนให้สอดคล้องกับมุมมองและความยาวโฟกัสที่อาจเปลี่ยนแปลงได้ ตลอดเวลาในการถ่ายวิดีโอทั่วไป รายงานวิจัยฉบับนี้เป็นผลการศึกษาเทคนิคการวัดค่าระดับการซูมของกล้องวิดีโอ โดยอาศัยวัตถุรูปทรงกลมที่ปรากฏในภาพที่ถ่ายได้ โดยผู้วิจัยได้ทำการพัฒนาวิธีการหาค่าระดับการซูมของกล้องด้วยวิธีการประมวลผลสัญญาณภาพ วิธีการดังกล่าววัดรูปทรงกลมในภาพจะถูกแยกออกจากส่วนอื่น ๆ ของภาพด้วยเทคนิค image segmentation จากนั้นโดยอาศัยการทำ feature extraction ทำให้สามารถหาขนาดเส้นผ่าศูนย์กลางของทรงกลมที่ปรากฏอยู่ในภาพได้ อันจะนำไปสู่การคำนวณค่าระดับการซูมสัมพันธ์ของกล้อง ณ. ขณะเวลาใด ๆ ได้

รายงานได้นำเสนอผลการทดลองเบื้องต้นเพื่อวัดค่าระดับความถูกต้องของวิธีการที่พัฒนาขึ้น โดยในการทดลอง ผู้วิจัยได้ทำการพัฒนาโปรแกรมบนระบบปฏิบัติการ Windows ด้วยภาษา C และ Microsoft Vision SDK เพื่ออ่านข้อมูลจากกล้องวิดีโอ และทำการประมวลผลภาพที่ได้เพื่อหาค่าระดับการซูม จากนั้นจึงทำการซ้อนทับส่วนของวัตถุรูปทรงกลมในภาพด้วยวัตถุเสมือนที่ทำการสร้างขึ้น โดยใช้ OpenGL โปรแกรมที่พัฒนาขึ้นในส่วนของการอ่านและแสดงผลภาพจากกล้องวิดีโอสามารถทำงานได้ในแบบ real-time แต่เนื่องจากอัลกอริทึมที่ใช้ในการวัดระดับการซูมยังมิได้ทำการปรับปรุงสำหรับการทำงานแบบ real-time โดยเฉพาะ ทำให้การทำงานโดยรวมทั้งระบบยังทำในแบบ real-time โดยสมบูรณ์ไม่ได้ ซึ่งปัญหาดังกล่าวจะเป็นส่วนหนึ่งของงานวิจัยต่อไปในอนาคต

ABSTRACT

To insert a virtual object onto a real image, specially a moving image, it is important that both the camera position relative to the virtual object and zooming level (or camera focal length) are known. Both parameters affect the size and perspective of a real object within the scene. Therefore, to make the added virtual object look natural, this information is required for proper rendering of a virtual object as the perspective and focal length change over time. This research report is the result of a study of camera zooming level estimation methods using a known spherical object within a scene. A simple zooming estimation algorithm, using a known-size spherical object has been developed. The method makes use of standard image processing operations, such as image segmentation and feature extraction. Once the spherical object is identified (by image segmentation), the object diameter is estimated. With the estimated size of the spherical object, the relative camera zooming level can be obtained.

This report describes some preliminary experimental results on the accuracy of the developed method. As part of the experiment, we have developed a MS Windows program using C and Microsoft Vision SDK, for reading picture data in real-time from a video camera. From the data received, the program processes each picture to obtain zooming level estimate. A virtual object created by OpenGL is subsequently added to the scene at the location of the spherical object. The program can be used to read and display video data from a camera in real-time. However, because current algorithm for zooming-level estimation has not yet been devised for real-time operation, an overall performance of the program is not yet at a real-time level. This problem will be part of the future research work.

กิตติกรรมประกาศ

โครงการการศึกษาอัลกอริทึมในการคำนวณค่าระดับการชมของกล้องถ่ายภาพวิดีโอได้รับความร่วมมือจากบุคคลหลายฝ่ายดังนี้

ขอขอบคุณ ดร.ธนรัตน์ ชลิตาพงศ์ ที่ได้ให้คำแนะนำเบื้องต้นเกี่ยวกับการใช้งานชุดพัฒนา Microsoft Vision SDK ขอขอบคุณนักศึกษาที่ Multimedia and Virtual Research Laboratory อันได้แก่ คุณ วรวิทย์ วีระพันธ์ และ คุณ พรชัย กาญจนสุภักดิ์ ที่ได้สละเวลาอันมีค่าในการช่วยทำให้งานสำเร็จลุล่วงไปได้ด้วยดี

โครงการวิจัยนี้ได้รับการสนับสนุนทุนวิจัยจากเงินรายได้ของคณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ประจำปีงบประมาณ 2545



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และตั้งกฏอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
บทที่	
1. บทนำ	1
1.1 ความเป็นมา	1
1.2 วัตถุประสงค์ของการวิจัย	2
1.3 ประโยชน์คาดว่าจะได้รับ	2
2. ทฤษฎีการพัฒนาและใช้งานระบบ Augmented Reality	3
2.1 เทคโนโลยี Augmented reality และการประยุกต์ใช้	3
2.2 ค่าความยาวโฟกัสและการคำนวณโดยใช้วัตถุทรงกลม	6
2.2.1 ความยาวโฟกัส (focal length)	6
2.2.2 การใช้วัตถุทรงกลมสำหรับการหาค่าความยาวโฟกัสจากภาพที่ถ่ายได้	8
2.3 การติดต่อกับกล้องวีดีโอด้วย Microsoft Vision System Development Kit	11
2.4 OpenGL และ การสร้างภาพสามมิติ	13
3. การคำนวณค่าระดับการซูมโดยใช้วัตถุทรงกลมและการสร้าง/ซ้อนทับวัตถุ เสมือน	14
3.1 วิธีการคำนวณค่าระดับการซูมโดยใช้วัตถุทรงกลม	14
3.1.1 การแยกรูปวัตถุทรงกลมในภาพ (Segmentation)	15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และตัด IV อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 การหาเส้นผ่าศูนย์กลางของวัตถุทรงกลมในภาพที่ถ่ายได้	16
3.1.3 การคำนวณหาความยาวโฟกัส	18
4. ผลการทดลอง	19
5. สรุปและแนวทางในการพัฒนาโปรแกรมในอนาคต	30
เอกสารอ้างอิง	31



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมา

Augmented reality คือการผสมวัตถุเสมือน (virtual object) ที่ได้จากการสังเคราะห์โดยเครื่องคอมพิวเตอร์ลงไปในภาพที่ถ่ายจากสภาพแวดล้อมจริง เสมือนวัตถุนั้นมีอยู่ในสิ่งแวดล้อมจริง โดยได้มีการทำเทคโนโลยีดังกล่าวไปใช้ประโยชน์กันอย่างแพร่หลาย เช่น ภาพโฆษณาที่ซ้อนทับลงบนพื้นสนามฟุตบอลในระหว่างการแข่งขัน การสร้างภาพอวัยวะซ้อนทับภาพที่เกิดจากการทำ CT scan ที่ใช้ในทางการแพทย์ เป็นต้น

จุดมุ่งหมายสำคัญสำหรับการซ้อนทับวัตถุเสมือนในงาน Augmented reality คือจะอย่างไรให้ผู้ใช้งาน หรือ ผู้ดูภาพที่ผ่านการซ้อนทับแล้วไม่เห็นความแตกต่างระหว่างภาพเสมือนกับภาพจริงที่ซ้อนทับกันได้ กล่าวคือดูเสมือนหนึ่งเป็นภาพเดียวกัน เพราะฉะนั้นนอกจากจะต้องใช้คอมพิวเตอร์ประสิทธิภาพสูงสำหรับการสร้างวัตถุเสมือนสามมิติเพื่อให้ได้ภาพเสมือนที่ดูสมจริงแล้ว วัตถุเสมือนจะต้องสามารถตอบสนองต่อสิ่งแวดล้อมในแบบทันทีทันใด (real-time response) เช่น การเปลี่ยนขนาดและมุมมองของวัตถุเสมือนเมื่อกล้องวิดีโอมีการเคลื่อนที่ไปมา และการเปลี่ยนขนาดเมื่อมีการเปลี่ยนความยาวโฟกัสของกล้อง เป็นต้น

จากปัญหาดังกล่าวโดยเฉพาะอย่างยิ่งในงานที่เกี่ยวข้องกับการซ้อนภาพเสมือนลงไปในการบินที่ภาพวิดีโอ การที่จะทำให้ได้ภาพมีความสมบูรณ์จึงต้องมีระบบตรวจสอบตำแหน่งของกล้องวิดีโอ และ ระบบตรวจสอบความยาวโฟกัส (focal length) มาช่วยในการกำหนดขนาด และ คุณสมบัติในทาง perspective ของภาพเสมือนที่จะปรากฏบนจอภาพ เพื่อให้ทั้งภาพวัตถุจริงและวัตถุเสมือนมีการปรับเปลี่ยนขนาดและมุมมองที่สอดคล้องกัน เพื่อให้ได้ภาพที่ดูเหมือนจริงมากที่สุด

การคำนวณค่าพารามิเตอร์ของกล้องถ่ายภาพ/วิดีโอ (เช่น ความยาวโฟกัส ตำแหน่ง แกนหมุน) เป็นสิ่งจำเป็นสำหรับการประยุกต์ใช้งานในด้านการสร้างภาพซ้อนทับระหว่างภาพที่ถ่ายจริงกับภาพที่สร้างขึ้น หรือที่เรียกว่า Virtual Studio Applications การได้มาซึ่งพารามิเตอร์ดังกล่าวกระทำได้หลายวิธี ทั้งโดยการใช้อุปกรณ์เชิงกล อุปกรณ์กำเนิดและตรวจจذبคลื่นแม่เหล็กไฟฟ้า และโดยการคำนวณจากภาพที่ถ่ายได้โดยใช้เทคนิคการประมวลผลภาพ (image processing techniques) ([1-4] และเอกสารอ้างอิงภายใน) ซึ่งวิธีการคำนวณโดยใช้เทคนิคการประมวลผลภาพนี้มิได้จำกัดอยู่เฉพาะในกรณีของ Virtual Studio Applications เท่านั้น หากแต่เป็นงานวิจัยที่เกี่ยวข้องเนื่องกับการประยุกต์ใช้งานในหลายลักษณะ อาทิ เช่น ความต้องการในการทำ automatic/semi-

automatic camera calibration หรือความต้องการในการสร้างภาพหลายมุมมอง (Multi-view system) จากภาพถ่ายจำนวนจำกัด รวมถึงความต้องการในการแทรกภาพที่สังเคราะห์ขึ้นลงไปในพื้นที่ถ่ายได้จากกล้องตั้งที่กล่าวไว้ในข้างต้น วิธีการในการได้มาซึ่งค่าพารามิเตอร์ของกล้องตั้งกล่าว โดยทั่วไปจะอาศัยข้อกำหนด (constraints) ที่เกิดจากวัตถุในภาพ เช่น เป็นวัตถุที่อยู่ในระนาบเดียวกัน เป็นวัตถุที่มีโครงสร้างแน่นอน (calibration pattern) หรือโดยอาศัยข้อมูลที่ทราบล่วงหน้าเกี่ยวกับตำแหน่งของกล้องที่ใช้ถ่ายภาพหลายมุมมอง ประกอบในการคำนวณดังกล่าว ในขั้นตอนนี้ งานวิจัยจะทำการศึกษาวิธีการคำนวณค่าความยาวโฟกัสโดยใช้รูปภาพที่มีโครงสร้างเฉพาะ [4, 8-9] เป็นหลัก เนื่องจากจะทำให้ได้ค่าที่ถูกต้องที่สุดเมื่อเปรียบเทียบกับแนวทางอื่น และใช้การคำนวณที่ซับซ้อนน้อยที่สุดอันจะทำให้สามารถทำงานในรูปแบบ real-time อันเป็นเป้าหมายของโครงการวิจัย นอกจากนี้ผลจากการทดลองด้วยวิธีการดังกล่าวจะเป็นค่ามาตรฐานที่ใช้ในการทดสอบวิธีการอื่น ๆ ที่ไม่มีข้อจำกัดด้านโครงสร้างของวัตถุในภาพที่ใช้สำหรับการคำนวณค่าความยาวโฟกัสในโอกาสต่อไป ในส่วนของค่าพารามิเตอร์อื่น ๆ ของกล้อง (เช่น ตำแหน่ง) จะกำหนดเป็นสมมติฐานว่ามีได้มีการเปลี่ยนแปลงตลอดการบันทึกภาพ โดยในทางปฏิบัติจะแก้ปัญหาข้อจำกัดดังกล่าวโดยการใช้อุปกรณ์ประเภท 6-DOF electromagnetic tracking device เพื่อให้ได้มาซึ่งค่าต่าง ๆ ดังกล่าวแทน

1.2 วัตถุประสงค์ของการวิจัย

1. เพื่อศึกษาการประยุกต์และปรับปรุงวิธีการปรับแต่งคุณสมบัติของกล้อง (camera calibration) โดยอาศัยวัตถุทรงกลมเพื่อใช้ในการหาค่าระดับการหมุนของกล้องวิดีโอ
2. เพื่อพัฒนาโปรแกรมประยุกต์ที่มีความสามารถในการจับภาพต่อเนื่องเป็นเฟรมในแบบทันทีทันใด (real time) บนเครื่องคอมพิวเตอร์ทั่วไปที่ใช้ระบบปฏิบัติการวินโดวส์
3. เพื่อศึกษาการพัฒนาโปรแกรมซึ่งนำข้อมูลจากที่ได้จากข้อ 1 ไปทำการสร้างวัตถุสังเคราะห์ที่มีการปรับเปลี่ยนมุมมองและขนาดที่สอดคล้องกับทิศทาง ระยะห่าง และความยาวโฟกัส ของกล้องวิดีโอได้โดยอัตโนมัติ

1.3 ประโยชน์คาดว่าจะได้รับ

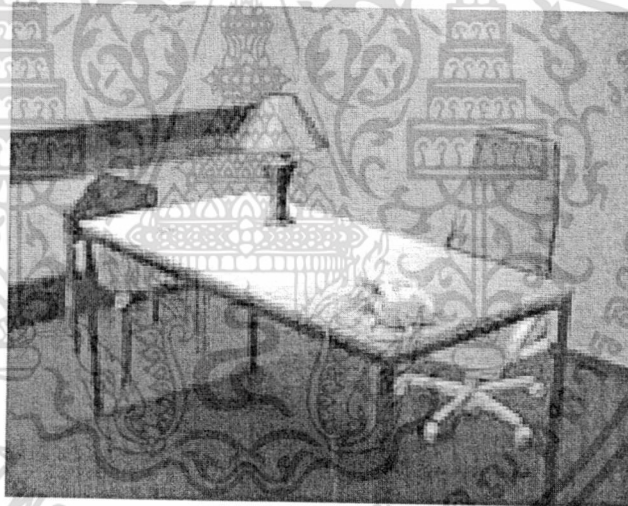
1. นำไปสู่การพัฒนากระบวนการสร้างภาพแบบ Augmented Reality เพื่อการผลิตสื่อวิดีโอ
2. เป็นส่วนหนึ่งของการศึกษาในระดับปริญญาโทของนักศึกษาคณะเทคโนโลยีสารสนเทศ

บทที่ 2

ทฤษฎีการพัฒนาและใช้งานระบบ Augmented Reality

2.1 เทคโนโลยี Augmented reality และการประยุกต์ใช้

Augmented reality เป็นวิทยาการแขนงหนึ่งที่มีผู้ศึกษาควบคู่ไปกับ Virtual Reality ในขณะที่ Virtual Reality คือการที่จะพยายามที่จะผสมวัตถุจริง (real object) โดยเฉพาะผู้ใช้งานให้ลงไปอยู่ในสถานะแวดล้อมที่สังเคราะห์ (synthetic environment) ขึ้น โดยคอมพิวเตอร์ เช่น โครงการมหาวิทยาลัยเสมือน (Virtual University) เป็นต้น แต่ Augmented reality คือการผสมผสานวัตถุสังเคราะห์ (synthetic object) ลงไปในสภาพแวดล้อมจริง ตัวอย่างเช่น ในงานภาพยนตร์เรื่อง “Who Framed Roger Rabbit” ที่พยายามสร้างตัวการ์ตูนที่ปรากฏอยู่ในฟิล์มภาพยนตร์เสมือนหนึ่งตัวการ์ตูนนั้นมีอยู่จริง เป็นต้น



ภาพที่ 1 แสดงภาพโต๊ะทำงานจริงที่ถูกซ้อนด้วย โคมไฟเสมือนและเก้าอี้เสมือนสองตัว

นักวิจัยหลายท่านที่ทำงานด้าน Augmented reality มักจะจำกัดการใช้งานที่เกี่ยวข้องกับการใช้อุปกรณ์ Head-Mounted Displays แต่ขอบเขตการใช้งานของ Augmented reality มีมากกว่า จึงอาจให้คำจำกัดความถึงลักษณะเฉพาะของ Augmented reality ได้ดังนี้

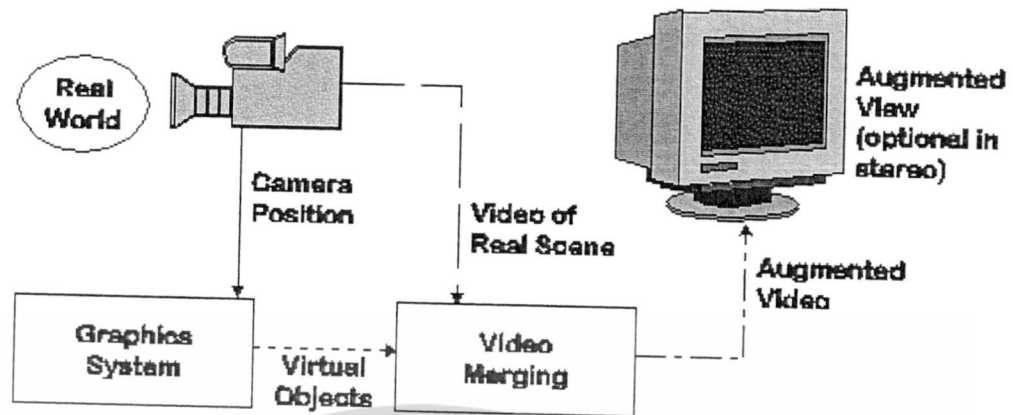
1. เป็นการทำงานที่เกี่ยวข้องกับการผสมระหว่างภาพจริง และ ภาพเสมือน
2. โปรแกรมประยุกต์ที่สร้างขึ้นต้องมีการตอบสนองในแบบทันทีทันใด (real time)
3. วัตถุสังเคราะห์ที่สร้างต้องมีลักษณะเป็นสามมิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการประยุกต์ใช้งานเทคโนโลยี Augmented reality มีดังนี้

1. ด้านการแพทย์ เช่น การทำภาพจำลองเพื่อช่วยในการฝึกฝนในห้องผ่าตัด หรือ อย่างเช่น การทำภาพจำลองซ้อนทับลงในภาพที่ได้จากการทำ CT Scan เป็นต้น เนื่องจากคุณสมบัติในแง่ที่ว่าวัตถุสังเคราะห์ที่ปรากฏมีลักษณะเป็นสามมิติ ดังนั้น แพทย์ที่ใช้งานสามารถมีอิสระในการมองในมุมต่างๆ ได้โดยไม่จำกัด
2. ด้านการผลิต และ การดูแลซ่อมแซมเครื่องจักร ประโยชน์ที่จะได้ในงานนี้คือ ใน การซ่อมแซมเครื่องจักรที่มีความซับซ้อนการใช้คู่มือที่เป็นสองมิติอยู่ในกระดาษ อาจทำให้ช่างที่ดูแลเสียเวลาในการทำความเข้าใจมาก เพราะฉะนั้นถ้าสามารถ แสดงข้อมูล (ข้อความหรือรูป) ลงไปบนอุปกรณ์จริงก็จะทำให้การทำงานมีประสิทธิภาพมากยิ่งขึ้น
3. Annotation และ Visualization ปัจจุบันมีการใช้งานอุปกรณ์ประเภทพกพาได้ (hand-held display) อย่างแพร่หลาย เช่นพนักงานใช้อุปกรณ์ช่วยบันทึกการใช้ไฟฟ้าตามบ้าน เป็นต้น การนำเอาเทคโนโลยี Augmented reality มาประกอบช่วยทำให้การใช้งานมีประสิทธิภาพสูงขึ้น
4. ด้านบันเทิง มีการใช้งานอย่างกว้างขวาง เช่น งานด้านเทคนิคภาพพิเศษ (visual effect) ที่ปรากฏในภาพยนตร์ทั้งในและต่างประเทศ มีการนำวัตถุสังเคราะห์ที่ สร้างจากเครื่องคอมพิวเตอร์ซึ่งมักเป็นวัตถุที่ไม่มีอยู่จริงหรือไม่สามารถนำมารวม ในภาพหลักได้ในเวลาเดียวกัน เช่น การรายงานอากาศที่มีการซ้อนภาพกราฟิกลง ไปด้วยในขณะรายงานสด
5. ด้านการทหาร มีการใช้งานเทคโนโลยีดังกล่าวเพื่อจำลองแบบฝึกในทางการทหาร เช่น การจำลองการบิน (flight simulator) เป็นต้น

เทคโนโลยี Augmented reality อาจจำแนกตามอุปกรณ์ที่ใช้ในการนำเสนอได้เป็น 2 ประเภท ได้แก่ การนำเสนอภาพผ่านอุปกรณ์ Head-Mounted Displays (HMD-based augmented reality system) และ การนำเสนอภาพผ่านจอมอนิเตอร์ (monitor-based augmented reality system) ในงานวิจัยนี้จะทำการศึกษา Augmented Reality ประเภทหลัง ซึ่งเป็นรูปแบบที่ใช้งานในด้านอุตสาหกรรมบันเทิงและสื่อประสม อันเป็นสาขาประยุกต์เป้าหมายหลักของงานวิจัย องค์ประกอบของระบบดังกล่าวแสดงได้ดังรูปที่ 2



ภาพที่ 2 แสดงองค์ประกอบของระบบ monitor-based Augmented reality system

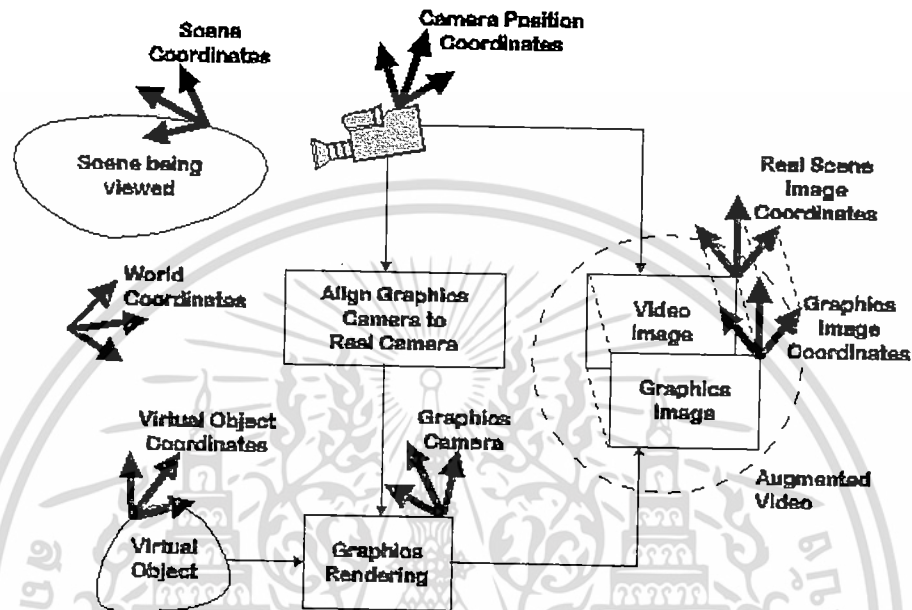
จากรูป ส่วนประกอบที่สำคัญได้แก่

- กล้องวิดีโอ เป็นส่วนที่นำสัญญาณภาพจากสิ่งแวดล้อมจริง
- ระบบตรวจหาตำแหน่งของกล้องวิดีโอ (Camera Position) เพื่อบอกตำแหน่งของกล้องให้ระบบสังเคราะห์ภาพกราฟิก (หรือ วัตถุเสมือน)
- ระบบการสังเคราะห์ภาพกราฟิก เพื่อนำเสนอไปผสมรวมกับภาพจริง (Graphics System) ระบบนี้ยังมีส่วนที่ทำการหาขนาดความยาวโฟกัสของกล้องวิดีโอในแบบ real-time ด้วย
- ระบบการผสมภาพที่ได้จากวิดีโอกับภาพที่ได้จากการสังเคราะห์ขึ้น (Video Merging) ผลที่ได้คือ ภาพวิดีโอที่นำเสนอทางจอคอมพิวเตอร์ (Augmented Video)
- มอนิเตอร์ที่แสดงภาพที่ผ่านขบวนการทั้งหมดแล้ว

เพื่อให้บรรลุวัตถุประสงค์ในการประยุกต์ใช้งาน ภาพที่ผ่านระบบ Augmented reality ส่วนที่เป็นวัตถุสังเคราะห์จะต้องมีความสอดคล้องสัมพันธ์กับส่วนที่เป็นภาพจริง เสมือนเป็นวัตถุที่ปรากฏอยู่จริงในจุดที่มีการบันทึกภาพ เช่น ขนาดและมุมมองของภาพสังเคราะห์จะต้องมีการเปลี่ยนแปลงสัมพันธ์กับตำแหน่งหรือค่าระดัการซูมของกล้องที่อาจมีการเปลี่ยนแปลงได้ขณะที่มีการบันทึกภาพ เป็นต้น เพื่อให้ส่วนของการสร้างภาพสังเคราะห์สามารถทำการปรับเปลี่ยนภาพได้ตามความเหมาะสมดังกล่าว จำเป็นต้องทราบค่าพารามิเตอร์ (parameters) ของตัวอุปกรณ์กล้องถ่ายภาพ ซึ่งแบ่งได้เป็น 2 ส่วน คือพารามิเตอร์ที่เกี่ยวกับปัจจัยภายใน (intrinsic parameters) ได้แก่ ความยาวโฟกัส (focal length) ความเพี้ยนของเลนส์ (lens distortion) และ พารามิเตอร์ที่เกี่ยวกับปัจจัยภายนอก (extrinsic parameter) ได้แก่ ตำแหน่งของกล้อง (position) มุมกล้อง (pose) ซึ่งใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และผู้ใช้อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบ Augmented reality จำเป็นต้องทำการตรวจวัดข้อมูลเหล่านี้แบบ real-time เพื่อนำไปป้อนให้กับส่วนที่ทำการสร้างภาพสังเคราะห์ต่อไป โดยในงานวิจัยนี้จะพิจารณาวิธีการในการได้มาซึ่งค่าความยาวโฟกัส อันเป็นพารามิเตอร์หนึ่งที่เกี่ยวข้องกับปัจจัยภายใน



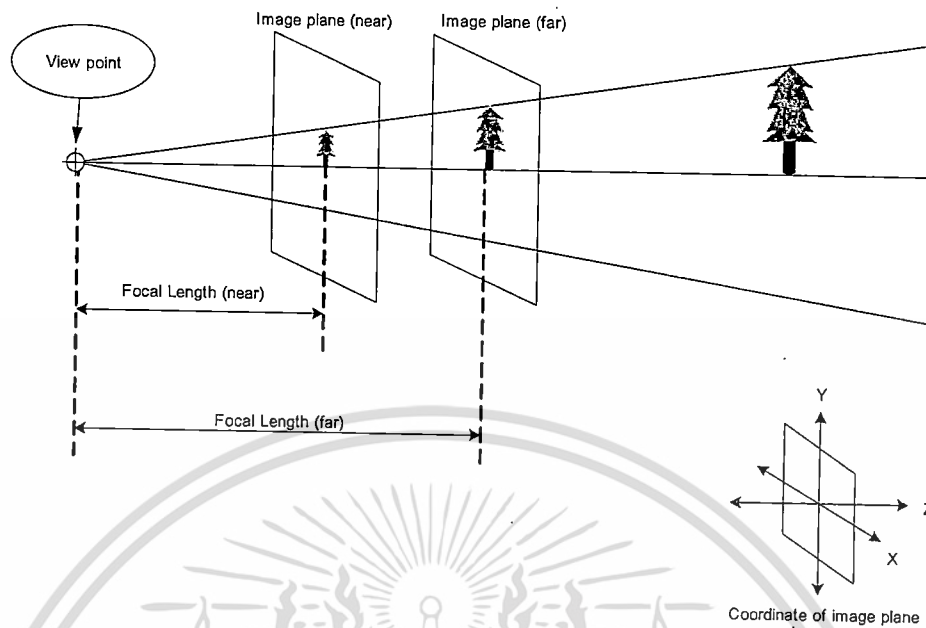
รูปที่ 3 แสดงส่วนประกอบของระบบ Augmented reality และตำแหน่งของวัตถุในโลกจริง และในโลกเสมือนที่ซ้อนทับกัน

2.2 ค่าความยาวโฟกัสและการคำนวณโดยใช้วัตถุทรงกลม

ในหัวข้อนี้จะกล่าวถึง ความหมายของค่าความยาวโฟกัส และการในการหาค่าความยาวโฟกัสของกล้องวิดีโอจากภาพที่ถ่ายได้ด้วยวิธีการประมวลผลสัญญาณภาพ (image processing)

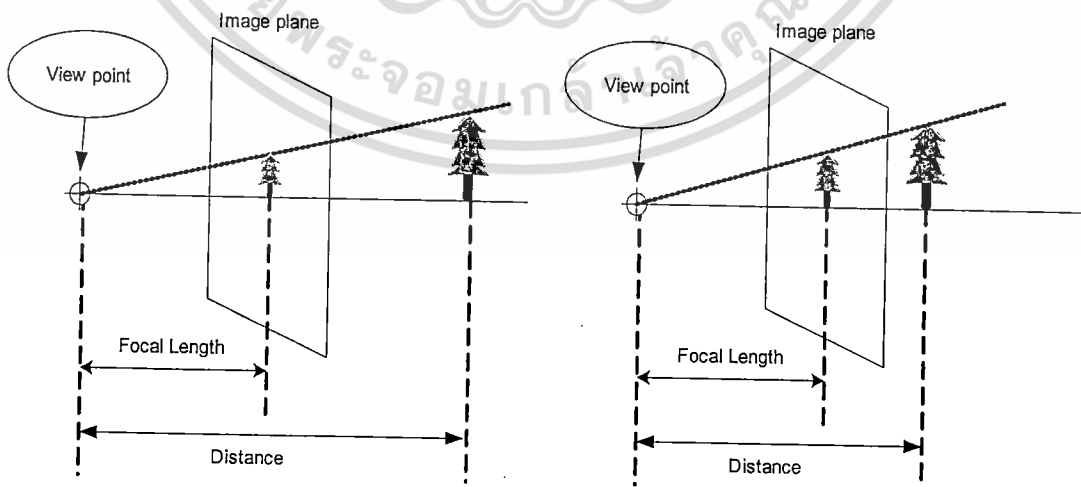
2.2.1 ความยาวโฟกัส (focal length)

ความยาวโฟกัสคือ ระยะทางจากจุดรวมแสงของเลนส์ที่เป็นจุดสมมุติอยู่ด้านหลังเลนส์ (View point) ถึงบริเวณฉากรับภาพ (ถ้าเป็นกล้องถ่ายภาพแบบเดิมก็คือตำแหน่งของฟิล์มที่ใช้บันทึกรูปภาพ ส่วนในกรณีของกล้องถ่ายภาพรูปดิจิทัลหรือกล้องวิดีโอ จะเป็นตำแหน่งที่ติดตั้งแผงเซ็นเซอร์ไวแสงรับภาพ) ดังรูปที่ 4



รูปที่ 4 แสดงค่าความยาวโฟกัสที่มีผลต่อขนาดของวัตถุบนภาพที่ฉายลงบนฉากรับภาพ

เมื่อผู้ใช้กล้องวิดีโอมีการซูมกล้อง เข้า-ออก ผลที่ได้คือความยาวโฟกัสมีการเปลี่ยนแปลง ดังรูปที่ 5 เมื่อฉากรับภาพ (Image Plane) เคลื่อนที่ออกจาก view point ผลที่ได้คือภาพจะมีขนาดใหญ่ขึ้น ในทางตรงข้ามถ้าฉากรับภาพเคลื่อนที่เข้าหา view point ภาพจะมีขนาดเล็กลง ในทำนองเดียวกันเมื่อระยะทางจาก view point ถึงวัตถุเปลี่ยนไปอันเป็นผลมาจากการที่กล้องวิดีโอเคลื่อนที่เข้าหาหรือไกลออกไปจากวัตถุที่ภาพถูกบันทึกลงบนฉากรับภาพส่งผลให้ภาพที่ได้มีขนาดใหญ่ขึ้น (เล็กลงในกรณีเคลื่อนที่ห่างออกไปจากวัตถุ) ตามไปด้วย หากแต่ในกรณีของการย้ายตำแหน่งกล้องจะทำให้ลักษณะทาง perspective ของภาพมีการเปลี่ยนแปลงตามไปด้วย



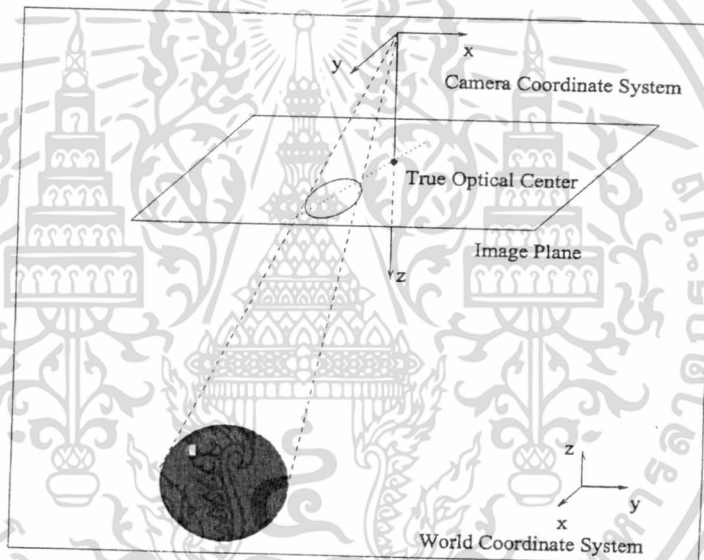
รูปที่ 5 แสดงผลของระยะห่างระหว่างกล้องกับวัตถุที่ถ่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 การใช้วัตถุทรงกลมสำหรับการหาค่าความยาวโฟกัสจากภาพที่ถ่ายได้

ในเบื้องต้นผู้วิจัยได้ทำการศึกษาวิธีการต่างๆ ที่ได้มีการพัฒนาขึ้นเพื่อการหาค่าระดับการซูมของกล้องวีดีโอ โดยได้ทำการศึกษาวิธีการที่เสนอโดย [3] ในรายละเอียด วิธีดังกล่าวซึ่งมีวัตถุประสงค์เพื่อใช้ในการทำ Camera Calibration มีส่วนของอัลกอริทึมในการหาความยาวโฟกัส (focal length) โดยอาศัยภาพวัตถุทรงกลมจากภาพที่ถ่ายได้ อันมีหลักการโดยย่อดังนี้

ภายใต้สมมติฐานของแบบจำลองกล้องรูเข็ม จากวัตถุทรงกลมที่ปรากฏในบริเวณที่กล้องจับภาพได้ ภาพวัตถุทรงกลมที่ปรากฏลงบนฉากรับภาพ จะเป็นภาพวงกลมหรือวงรี ขึ้นอยู่กับทิศทางของวัตถุดังกล่าวเทียบกับ view point ของกล้องถ่ายภาพ ดังแสดงในรูปที่ 6



รูปที่ 6 แสดงวัตถุทรงกลมที่ปรากฏบน image plane และเกิดเป็นรูปวงรี

จากรูปที่ 6 ตำแหน่งของ view point กำหนดได้โดยใช้ 3 แกนดังรูป โดยฉากรับภาพตัดตั้งฉากกับแกน Z แต่วัตถุทรงกลมไม่ได้ในอยู่ในแนวแกน Z ภาพที่ปรากฏบนฉากรับภาพจะปรากฏเป็นรูปวงรี แต่ถ้าวัตถุอยู่ในแนวแกน Z รูปวัตถุที่ปรากฏจะเป็นรูปวงกลมที่มีจุดศูนย์กลางอยู่ที่ทับกับจุดศูนย์กลางของฉากรับภาพ จากคุณสมบัติบางประการของรูปวงรี ทำให้สามารถหาความยาวโฟกัสโดยไม่จำเป็นต้องทราบระยะห่างระหว่างกล้องกับวัตถุจากสมการต่อไปนี้ [3]

$$f = \frac{1}{2} \sqrt{\frac{4x_2^2 K_1^2 (1 + K_2^2)}{K_2^2 (1 - K_1^2)} - L_{MnA}^2} \quad (1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยค่า K_1 คือค่าความยาวของแกน minor axis หาค่าด้วยค่าความยาวของแกน major axis ตามสมการ

$$K_1 = \frac{L_{MMA}}{L_{MJA}} \quad (2)$$

ส่วนค่า K_2 มีค่าเท่ากับ C_1 หาค่าด้วย C_2 ซึ่งมีค่าเท่ากับ X_c หาค่าด้วย Y_c โดยที่ค่า X_c และ ค่า Y_c คือจุดศูนย์กลางของวงรีที่เกิดขึ้น

$$K_2 = \frac{c_1}{c_2} \quad (3)$$

จากวิธีการดังกล่าวข้างต้น ทำให้สามารถหาความยาวโฟกัสโดยไม่ต้องอาศัยระยะห่างระหว่างกล้องและวัตถุทรงกลมได้ โดยใช้เทคนิคการประมวลผลรูปภาพเพื่อให้ได้มาซึ่งค่าพารามิเตอร์ต่าง ๆ ในสมการข้างต้น

ผู้วิจัยได้ทำการทดลองวิธีการข้างต้น โดยใช้กล้องถ่ายภาพดิจิทัลที่สามารถซูมสูงสุดได้หกเท่า ทำการถ่ายภาพวัตถุทรงกลมสี่ดวงบนพื้นสีขาว ในมุมและขนาดความยาวโฟกัสต่างๆ กัน จากนั้นนำรูปที่ได้มาทำการวิเคราะห์ด้วยโปรแกรม MATLAB[®] โดยมีกระบวนการในการทดลองดังนี้

- 1) ทำการบันทึกภาพวัตถุทรงกลมด้วยกล้องดิจิทัล (ดูรูปที่ 7 ประกอบ)

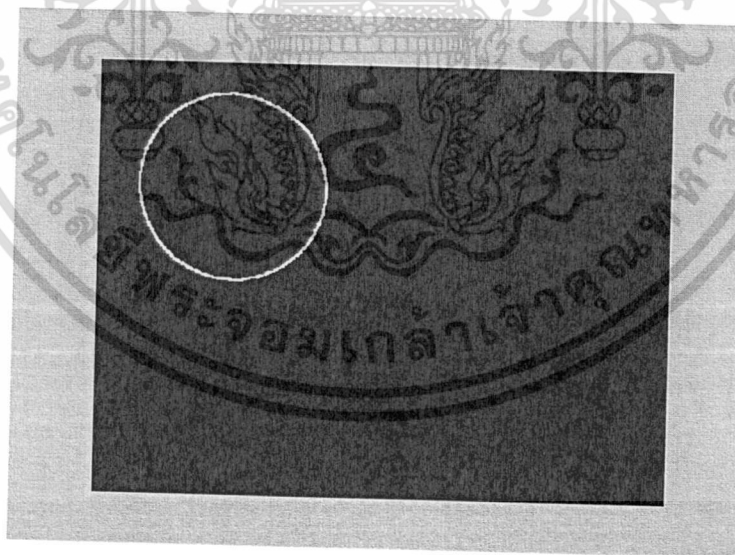


รูปที่ 7 แสดงตัวอย่างภาพที่ถ่ายด้วยกล้องดิจิทัลเพื่อนำมาทำการทดลอง

- 2) ทำการตรวจหาเส้นขอบของวัตถุสีแดง และทำการเปลี่ยนทุก pixels ที่ไม่ใช่เส้นขอบให้กลายเป็นสีดำ โดยอาศัยอัลกอริทึมที่แสดงดังในรูปที่ 8 ตัวอย่างผลลัพธ์ที่ได้แสดงดังในรูปที่ 9

```
function OutputImg = OutlineObj(InputImg)
    Img = double(InputImg);
    Dim = size(Img);
    AnsImg = zeros([Dim(1) Dim(2)]);
    for y = 1:Dim(2)
        for x = 1:Dim(1)
            if Img(x,y,1) > Img(x,y,2) & Img(x,y,1) >
                Img(x,y,3) &
                    std(Img(x,y,1:3)) > 50
                AnsImg(x,y) = 1;
            end
        end
    end
    AnsImg = double(bwmorph(AnsImg,'dilate',20));
    AnsImg = double(bwmorph(AnsImg,'erode',20));
    ErodeImg = double(bwmorph(AnsImg,'erode',2));
    OutputImg = AnsImg-ErodeImg;
```

รูปที่ 8 อัลกอริทึมที่ใช้ในการทดลองหาขอบเขตวัตถุสีแดง (MATLAB Syntax)



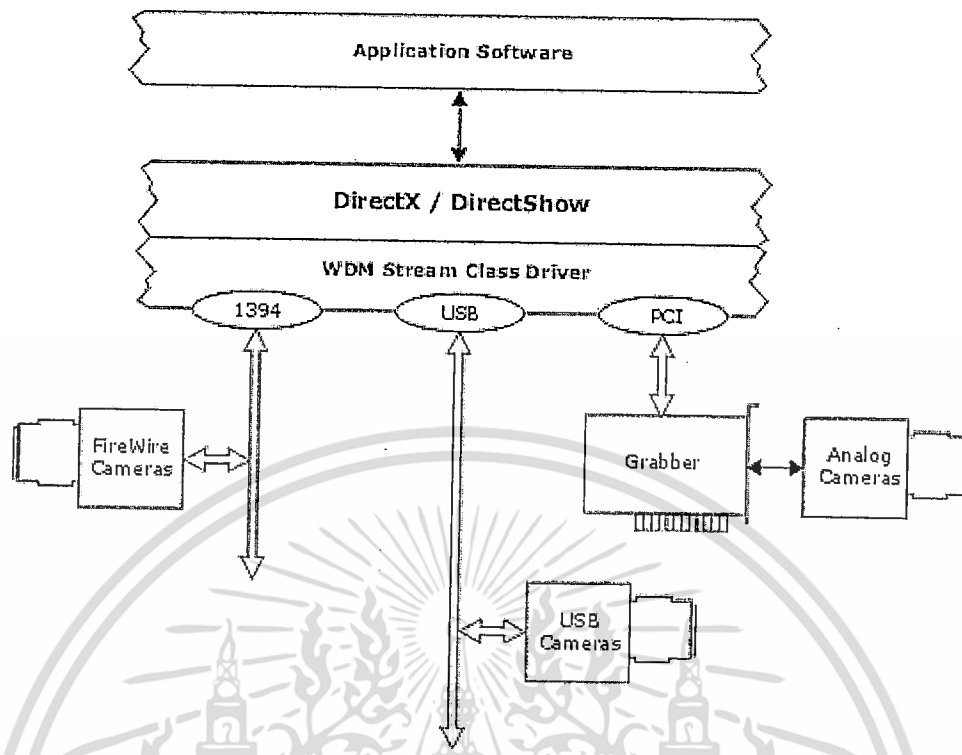
รูปที่ 9 ตัวอย่างผลลัพธ์ที่ได้จากการหาขอบเขตของรูปวงรี

จากการทดลองถ่ายภาพที่ระดับการซูมและมุมกล้องต่าง ๆ พบว่าภาพวัตถุทรงกลมที่ถ่ายได้ ไม่สามารถทำการตรวจพบลักษณะของวัตถุที่เป็นวงรีแต่อย่างใด แม้ในกรณีของภาพที่วัตถุทรง เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

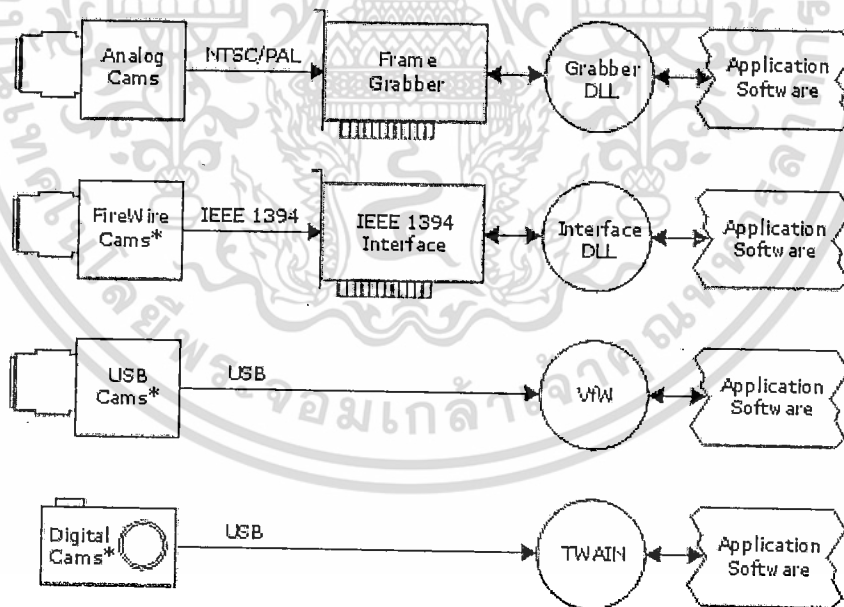
กลมปรากฏอยู่ห่างจากจุดศูนย์กลางภาพ เป็นผลทำให้ไม่สามารถทำการหาค่าระดับการซูมได้ตาม
ทฤษฎี จากการศึกษพบว่าสาเหตุมาจากการที่กล้องถ่ายภาพที่ใช้มีความยาวโฟกัสไม่สั้นพอ
ตลอดระยะห่างจากกล้องไปยังวัตถุไม่ใกล้พอ ที่จะทำให้เกิดผลดังกล่าวชัดเจน เนื่องจากข้อจำกัด
ของรูปแบบการประยุกต์ใช้งานที่เป็นเป้าหมายของงานวิจัย ทำให้ไม่สามารถกำหนดให้เงื่อนไขดัง
กล่าวข้างต้นเพื่อการนำวิธีการที่กล่าวมาข้างต้นมาใช้ได้ ผู้วิจัยจึงได้ทำการพัฒนาวิธีการหาค่า
ระดับการซูมขึ้น โดยยังคงอาศัยวัตถุรูปทรงกลมประกอบในการคำนวณ รายละเอียดจะได้กล่าวถึง
ในบทถัดไป

2.3 การติดต่อกับกล้องวิดีโอด้วย Microsoft Vision System Development Kit

เพื่อให้บรรลุวัตถุประสงค์ในการสร้างระบบ Augmented Reality จำเป็นต้องสามารถทำ
การอ่านข้อมูลจากอุปกรณ์ถ่ายภาพวิดีโอได้ในแบบ real-time จากการศึกษพบว่าบริษัท Microsoft
ได้ทำการพัฒนา Microsoft Vision System Development Kit (MS Vision SDK) ซึ่งเป็น library ที่
ช่วยในการสร้าง โปรแกรมสำหรับการจัดการและวิเคราะห์ข้อมูลภาพที่ทำงานบนระบบปฏิบัติการ
วินโดวส์ Software library ดังกล่าวถูกพัฒนาโดย Vision Technology Research Group ซึ่งเป็นส่วน
หนึ่งของ Microsoft Research เพื่อสนับสนุนการวิจัยในงานทางด้าน Image Processing , real-time
Image Processing และ Computer Vision ที่ต้องการสร้าง โปรแกรมประยุกต์บนระบบปฏิบัติการ
วินโดวส์ MS Vision SDK ได้เตรียมคลาส และ ฟังก์ชันในภาษา C++ สำหรับการทำงานที่เกี่ยวข้อง
กับงานด้านภาพ ผู้สร้าง โปรแกรมประยุกต์ที่ติดตั้ง MS Vision SDK สามารถพัฒนาโปรแกรม
ประยุกต์โดยใช้ MS Visual C++ ในชุด Visual Studio ได้อย่างอิสระโดยไม่ขึ้นกับอุปกรณ์ต่อเชื่อม
ที่น่าสัญญาณเข้าเครื่องคอมพิวเตอร์ เพราะ MS Vision SDK อาศัยหลัก Windows Driver Model
(WDM) ที่ทำหน้าที่เป็นตัวกลางในการติดต่อระหว่าง application program กับ อุปกรณ์เชื่อมต่อ
เพราะฉะนั้นอุปกรณ์ที่เป็นตัวนำข้อมูลภาพทั้งภาพนิ่ง และ ภาพเคลื่อนไหว เช่น กล้องถ่ายรูป
ดิจิทัล กล้องถ่ายวิดีโอดิจิทัล ที่ต่อกับอุปกรณ์คอมพิวเตอร์ผ่านทาง USB port หรือ ผ่านทาง port
IEEE 1394 (Fire Wire) และ รวมทั้งกล้องวิดีโออนาล็อกที่ต่อผ่านทาง capture card (เช่น Metrox
card , Meteor card เป็นต้น) ที่ต่อกับคอมพิวเตอร์ผ่านทาง PCI bus ดังแสดงในภาพที่ 10 ส่วนภาพ
ที่ 11 แสดงการสร้าง โปรแกรมประยุกต์ที่ไม่มี WDM ผู้สร้าง โปรแกรมประยุกต์ต้องสร้างส่วนที่ติด
ต่อกับอุปกรณ์ต่อเชื่อมแต่ละชนิดเอง และเมื่อมีการนำอุปกรณ์ใหม่มาใช้จะต้องมีการแก้ไข
โปรแกรมเพื่ทำให้ระบบสามารถทำงานได้



รูปที่ 10 Windows Driver Model (WDM)



* FireWire and USB cameras which do NOT comply with the WDM standard

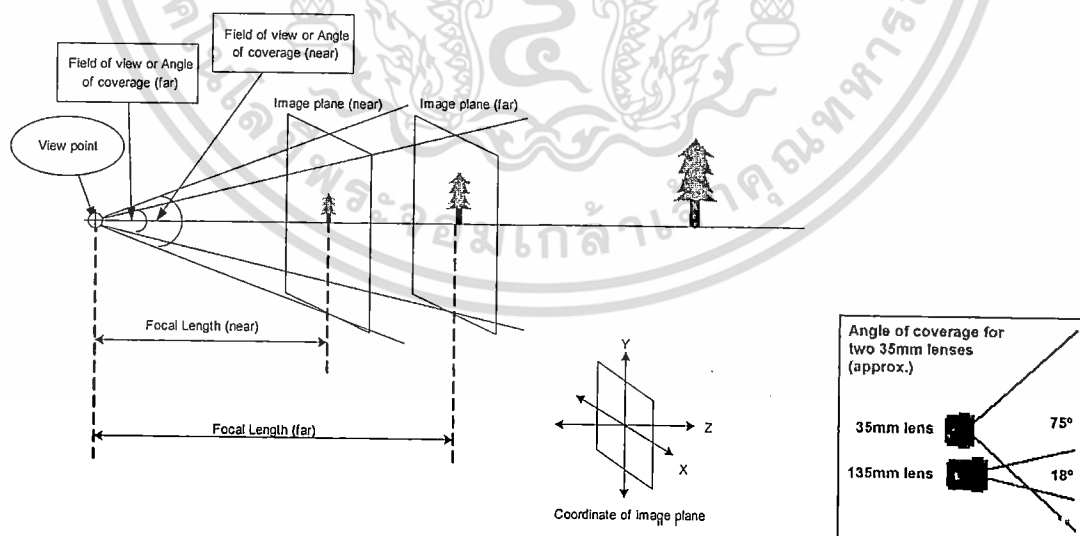
รูปที่ 11 การสร้างโปรแกรมประยุกต์ที่ไม่สนับสนุน WDM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และแจ้งอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 OpenGL และ การสร้างภาพสามมิติ

Open GL เป็นมาตรฐานของ Application programming interface (API) ที่ใช้ในการสร้างวัตถุสังเคราะห์สามมิติที่มีความสามารถในการตอบสนอง (interactive) ถูกสร้างขึ้น โดยบริษัท Silicon Graphic Inc. ในแบบ Open Source และ สามารถรองรับกับคอมพิวเตอร์ที่มีแพลตฟอร์มต่าง ๆ กัน เช่น สามารถสร้างโปรแกรมประยุกต์ที่ใช้ OpenGL บนระบบปฏิบัติการ วินโดว์ หรือระบบยูนิกซ์ เป็นต้น ในงานวิจัยนี้เลือกใช้ OpenGL ในการสร้างวัตถุสามมิติ เนื่องจากวัตถุสังเคราะห์ที่ได้จากการสร้างโดย OpenGL มีลักษณะที่เป็นวัตถุที่มีความสามารถในการตอบสนอง (interactive) ได้ กล่าวคือผู้ใช้สามารถเปลี่ยนแปลงมุมมองวัตถุสังเคราะห์ที่สร้างขึ้นได้โดยการปรับเปลี่ยนตำแหน่งของกล้องเสมือน (Virtual Camera) ตลอดจนการซูมเข้าหรือออกจากวัตถุสังเคราะห์ได้ โดยการผ่านค่าพารามิเตอร์ในหัวข้อ 2.5 และ 2.6 เข้าไปยังฟังก์ชันของ OpenGL นอกจากนี้ข้อดีอีกประการหนึ่งคือ วัตถุสังเคราะห์ที่สร้างจาก OpenGL และ ผ่านขบวนการตรวจสอบหาขนาดที่เกิดจากการเคลื่อนที่ของกล้องวีดีโอ และการซูมของกล้องวีดีโอ รวมถึงทิศทาง และ มุมของกล้องวีดีโอ สามารถจะนำไปรวมกับภาพที่อ่านได้จากกล้องวีดีโอด้วย MS Vision SDK โดยใช้ฟังก์ชันที่มีอยู่ใน OpenGL ได้

ในส่วนของการเปลี่ยนอัตราการซูมใน OpenGL กระทำได้โดยอาศัยหลักการในการเปลี่ยนแปลงค่า Field of view (Angle of coverage) จากรูปที่ 13 เมื่อทำการเปลี่ยนแปลงค่าความยาวโฟกัส เช่น เพิ่มความยาวโฟกัส (ซึ่งทำให้ image plane เคลื่อนที่เข้าใกล้วัตถุ) ค่า Field of view จะมีค่าลดลง



รูปที่ 13 ความสัมพันธ์ระหว่าง Field of view กับความยาวโฟกัส

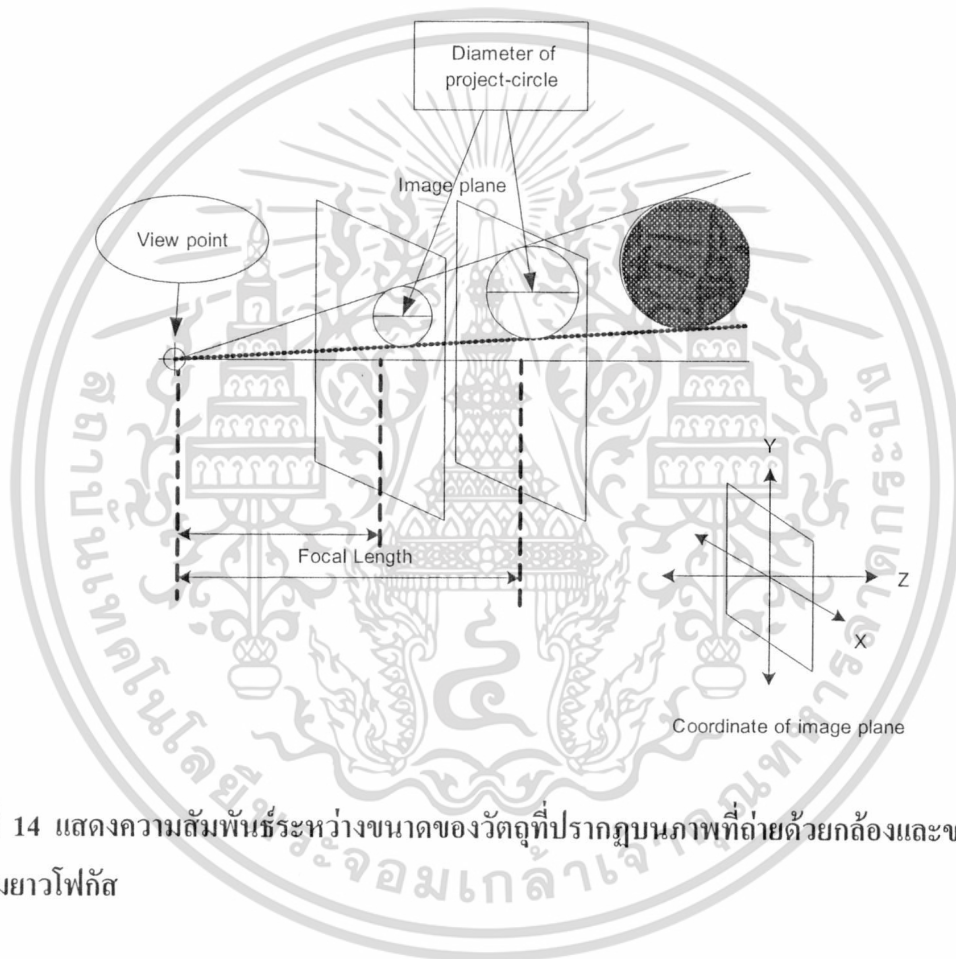
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และ 13 องค์อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การคำนวณค่าระดับการชมโดยใช้วัตถุทรงกลมและการสร้าง/ซ้อนทับวัตถุเสมือน

3.1 วิธีกรคำนวณค่าระดับการชมโดยใช้วัตถุทรงกลม

วิธีการคำนวณค่าระดับการชมที่พัฒนาขึ้นอันเป็นส่วนหนึ่งของการงานวิจัยนี้ อาศัยความสัมพันธ์พื้นฐานระหว่างขนาดของวัตถุในภาพกับค่าระดับการชมของกล้อง ดังแสดงในรูปที่ 14



รูปที่ 14 แสดงความสัมพันธ์ระหว่างขนาดของวัตถุที่ปรากฏบนภาพที่ถ่ายด้วยกล้องและขนาดความยาวโฟกัส

จากรูปขนาดของวัตถุที่ปรากฏบนภาพที่ถ่ายได้ ณ ตำแหน่งของกล้องและวัตถุเดียวกัน จะแปรผกผันกับค่าความยาวโฟกัสของกล้อง โดยในกรณีของวัตถุทรงกลม ขนาดของวัตถุที่ปรากฏในภาพจะแปรผันโดยตรงกับเส้นผ่าศูนย์กลางของ projected image ของวัตถุนั้น จากความสัมพันธ์ข้างต้นทำให้สามารถทำการหาค่าระดับการชมหรือความยาวโฟกัสของกล้องได้โดยการหาเส้นผ่าศูนย์กลางของภาพทรงกลมที่ถ่ายได้ โดยค่าระดับการชมดังกล่าวเป็นค่าสัมพันธ์กับค่าระดับการชมค่าใดค่าหนึ่งที่ทราบเส้นผ่าศูนย์กลางของภาพทรงกลมแล้ว

กระบวนการในการคำนวณค่าเส้นผ่าศูนย์กลางของภาพทรงกลม (อันจะนำไปสู่การหาค่าระดับการขมต่อไป) สามารถทำได้โดยอาศัยกระบวนการประมวลผลสัญญาณภาพแบบดิจิทัลซึ่งมีรายละเอียดดังต่อไปนี้

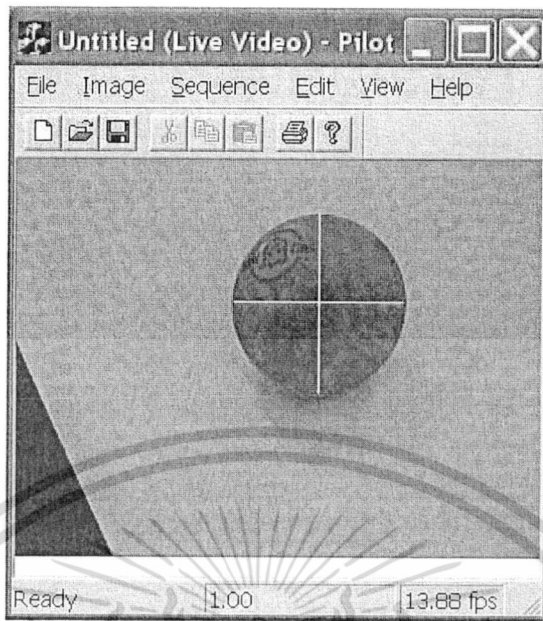
3.1.1 การแยกวัตถุทรงกลมในภาพ (Segmentation)

ในระบบที่ทำการวิจัยและพัฒนา แต่ละเฟรมของภาพต่อเนื่องที่ได้จากกล้องวิดีโอจะถูกอ่านเข้ามาโดยอาศัย Microsoft Vision SDK จากนั้นจะต้องนำภาพที่ได้ในแต่ละเฟรมมาทำการหาตำแหน่งขอบเขตของวัตถุทรงกลมในภาพที่ถ่ายได้ สาเหตุที่เลือกใช้วัตถุทรงกลมเพื่อการหาความยาวโฟกัสก็เนื่องจากเป็นรูปทรงที่มีลักษณะเฉพาะ คือมีความสมมาตรทุกทิศทาง เพราะฉะนั้นผู้ใช้สามารถเคลื่อนไหวกล้องวิดีโอไปมาได้ทุกทิศทางโดยมีข้อจำกัดเพียงที่วัตถุดังกล่าวต้องอยู่ในภาพเต็ม ในการตรวจหาวัตถุทรงกลมนั้น ในงานวิจัยนี้ได้อาศัยของสีของวัตถุในการแยกวัตถุทรงกลมออกจากวัตถุอื่นหรือภาพพื้น โดยในการทดลองใช้วัตถุทรงกลมสีน้ำเงินหรือสีแดง ที่มีความเข้มตัวของสีสูง โดยในแต่ละเฟรม จะทำการตรวจสอบค่าความสว่างของสีแต่ละสีเทียบกับในแต่ละ pixel เพื่อตรวจหาตำแหน่งของวัตถุดังกล่าว

```
int ir = (int)image.Pixel(pixelx, pixely).R();
int ig = (int)image.Pixel(pixelx, pixely).G();
int ib = (int)image.Pixel(pixelx, pixely).B();
float imean = (float)(ir+ig+ib)/3;
float ivar = (float)((ir-imean)*(ir-imean) +
((ig-imean)*(ig-imean))+((ib-imean)*(ib-imean)))/3;
```

รูปที่ 15 แสดงอัลกอริทึมที่ใช้ในการแยกส่วนของรูปภาพวัตถุทรงกลม

จากอัลกอริทึมข้างต้น สมมุติฐานมีอยู่ว่าถ้าจุดนั้นเป็นสีน้ำเงิน (หรือแดง) ค่าความสว่างของสีน้ำเงินเมื่อเทียบกับค่าความสว่างของสีแดงและสีเขียวจะต้องมีค่ามากกว่า ในการแยกได้อาศัยอัตราส่วนค่าความแปรปรวน (Variance) ระหว่างสีน้ำเงินกับค่าความแปรปรวนของทุกสีเป็นตัววัด จากการทดลองพบว่า ค่าความแปรปรวนที่ขนาดสูงกว่า 50 ให้ผลที่ถูกต้องเมื่อผ่านกระบวนการแยกวัตถุที่ใช้สำหรับการหาความยาวโฟกัส ตัวอย่างผลการทดลองแสดงในรูปที่ 16



รูปที่ 16 แสดงตำแหน่งของวัตถุทรงกลมสีน้ำเงินที่โปรแกรมตรวจจับได้ (กากบาทสีขาวในภาพ)

3.1.2 การหาเส้นผ่านศูนย์กลางของวัตถุทรงกลมในภาพที่ถ่ายได้

จากขอบเขตของรูปวัตถุทรงกลมที่หาได้จากข้อ 3.1.1 ขั้นตอนต่อไปคือการหาค่าเส้นผ่านศูนย์กลางของทรงกลมที่ปรากฏในภาพ เนื่องจากในงานวิจัยนี้ปัจจัยที่มีความสำคัญคือความเร็วในการประมวลผล เพราะโปรแกรมประยุกต์ที่สร้างขึ้นมาต้องสามารถใช้งานได้แบบทันทีทันใด (real time) เพราะฉะนั้นนอกจากประสิทธิภาพของอุปกรณ์นำมาใช้งานแล้ว เช่น หน่วยประมวลผลกลางของคอมพิวเตอร์ (CPU) และ การ์ดจอสามมิติ เป็นต้น อัลกอริทึมสำหรับการคำนวณก็มีเป็นสิ่งสำคัญในการคำนวณหาเส้นผ่านศูนย์กลางของวงกลมในภาพมีหลายวิธีในการหา ซึ่งแต่ละวิธีก็มีข้อดีข้อเสียที่แตกต่างกันไป เช่น บางวิธีที่มีความถูกต้องสูง หากแต่ใช้เวลาในการคำนวณมาก วิธีที่ใช้หาเส้นผ่านศูนย์กลางของวงกลมในภาพในที่นี้ มีขั้นตอนดังต่อไปนี้

- 1) ทำการแบ่งภาพที่ได้ ออกเป็นสี่เหลี่ยมที่มีพื้นที่เท่า ๆ กัน โดยการแบ่งจะกระทำเป็นลำดับขั้น โดยเริ่มจากการแบ่งอย่างหยาบก่อน
- 2) จากแต่ละส่วนของภาพที่แบ่งในขั้นที่ 1 ทำการหาค่าสี ณ จุดศูนย์กลางของแต่ละส่วน เมื่อพบสีที่ต้องการ (เช่น ในการทดลองใช้สีน้ำเงิน) จากนั้นทำการตรวจจุดที่ติดกันทั้งทางด้านซ้ายและขวาว่ามีค่าสีเท่าหรือใกล้เคียงกับสีที่จุดกึ่งกลางหรือไม่ ถ้าใช่ให้ทำการตรวจสอบในลักษณะเดียวกันต่อไปในทิศทางนั้น ๆ จนกว่าจะพบจุดที่มีค่าสีต่างออกไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และ 16 อ่างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการวัดระยะระหว่างจุดสุดท้ายที่มีค่าสีตามที่กำหนดทางด้านซ้ายกับจุดสุดท้ายที่มีค่าสีตามที่กำหนดทางด้านขวา

- 3) กระทำในลักษณะเดียวกันกับทุก ๆ ส่วนของภาพย่อยที่ได้จากการแบ่งในขั้นที่ 1
- 4) ทำการเปรียบเทียบว่าเส้นตรงที่ได้จากข้อ 2-3 เพื่อหาเส้นที่ยาวที่สุด โดยตัดเส้นที่มีค่าความยาวต่ำกว่าที่กำหนด ทั้งนี้เพราะในบางครั้งผลของสัญญาณรบกวนอาจทำให้การตรวจสอบค่าเส้นผ่านศูนย์กลางผิดพลาดได้ เช่น ในการทดลองกำหนดให้ค่าที่ได้ต้องมากกว่า 10 ซึ่งได้จากค่าเส้นผ่านศูนย์กลางที่ระดับการชุมนุมน้อยที่สุด (ความยาวโฟกัสสั้นที่สุด) และกล้องอยู่ในตำแหน่งที่ห่างจากวัตถุมากที่สุดที่กำหนดให้กระทำได้
- 5) นำค่าที่ได้จากข้อ 4 มาหาจุดกึ่งกลาง จากนั้นหาเส้นตรงที่ตั้งฉากกับเส้นตรงดังกล่าว (เส้นตรงตามแนวตั้ง) ซึ่งลากจากขอบของวงกลมด้านหนึ่งไปยังอีกด้านหนึ่ง เส้นตรงดังกล่าวแทนเส้นผ่านศูนย์กลางของวงกลมนั้น ๆ (รูปที่ 17 ประกอบ)
- 6) กรณีที่หากทำตามขบวนการที่กล่าวทั้งหมดแล้วไม่พบจุดสีหรือเส้นที่มีค่าความยาวต่ำกว่าค่า Threshold ที่ได้ตั้งไว้ ให้ทำการแบ่งภาพออกเป็นส่วนย่อย ๆ โดยให้มีขนาดของแต่ละส่วนย่อยที่เล็กลง (แบ่งที่ให้ละเอียดขึ้น) จากนั้นทำการตามขบวนการทั้งหมดที่กล่าวมาข้างต้นกับแต่ละส่วนของภาพที่แบ่งใหม่



รูปที่ 17 แสดงผลของอัลกอริทึมในการหาเส้นผ่านศูนย์กลางของวงกลมในภาพ

3.1.3 การคำนวณหาความยาวโฟกัส

ความยาวโฟกัสที่ได้จากค่าเส้นผ่าศูนย์กลางในหัวข้อ 3.1.2 เป็นค่าสัมพัทธ์ (Relative) โดยจากค่าที่ได้ เพื่อให้สามารถหาค่าโฟกัสที่แท้จริงต้องทราบตั้งค่าเริ่มต้นก่อน จากนั้นทำการวัดค่าที่มีการเปลี่ยนแปลงเทียบกับค่าเริ่มต้นจากสมการต่อไปนี้

$$\theta' = \text{Tan}^{-1}(\text{Tan}(\theta) * (\text{Dia} * D' / \text{Dia}' * D)) \quad (4)$$

ค่า θ เป็นค่า Field of view เริ่มต้นที่ผู้ใช้ต้องตั้งค่าไว้เมื่อเริ่มใช้งาน (สาเหตุที่ต้องใช้ค่า Field of view เพราะใน OpenGL กล้องเสมือนภายในมีการเปลี่ยนแปลงอัตราการซูมโดยใช้ค่า Field of view ซึ่งค่านี้แปรผันตรงกับค่าความยาวโฟกัส) ส่วนค่า Dia คือ ค่าเส้นผ่าศูนย์กลางของรูป วงกลมที่เป็นค่าเริ่มต้นที่ได้ทำการบันทึกไว้ก่อนแล้ว ค่า D คือ ค่าระยะห่างระหว่างจุด origin (ซึ่งเป็นตำแหน่งเริ่มต้นของกล้องวิดีโอ) กับตำแหน่งของวัตถุทรงกลมที่ใช้เป็นวัตถุ สำหรับใช้ในการคำนวณหาความยาวโฟกัส ส่วนค่า D' ซึ่งเป็นระยะห่างระหว่างกล้องกับวัตถุทรงกลม ณ. ขณะปัจจุบัน ในทางปฏิบัติหาได้จากวิธีการอื่น เช่น จากอุปกรณ์ ISOTRAK® II เป็นต้น ส่วนค่าเส้นผ่าศูนย์กลาง Dia' สามารถหาได้จากวิธีที่ได้กล่าวมาแล้วข้างต้น

ผลการทดลองที่นำเสนอในบทนี้ใช้วิธีการดังที่ได้นำเสนอในบทที่ 3 โดยมีรายละเอียดของอัลกอริทึมที่แสดงในรูปแบบที่ 19

```
function Output = CalcDia(Img)
% Calculate Diameter for Circle
InputImg = OutlineObj(Img);
[x y] = size(InputImg);
ScanY = y ;
turn = 1;
sw = 0;

while turn == 1 & ScanY > 0
    if sum(InputImg(1:x,ScanY)) > 0 & sw == 0
        y1 = ScanY;
        sw = 1;
    elseif sum(InputImg(1:x,ScanY)) == 0 & sw == 1
        y2 = ScanY + 1;
        turn = 0;
    end
    ScanY = ScanY - 1;
end

ScanX = x ;
turn = 1;
sw = 0;

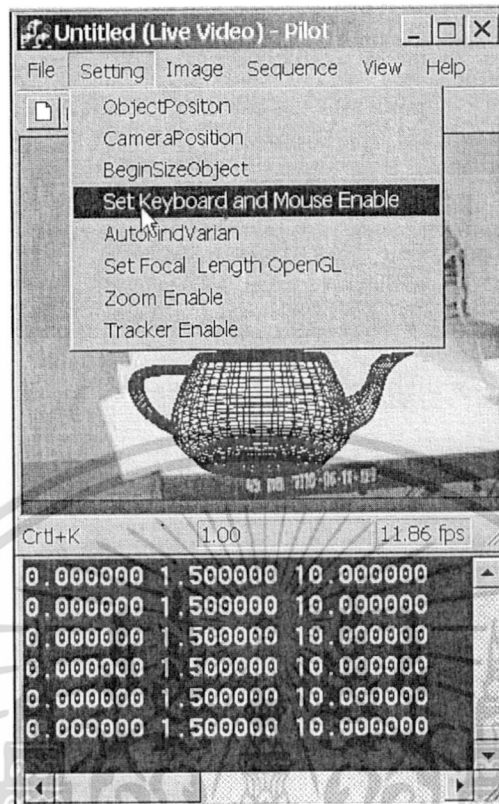
while turn == 1 & ScanX > 0
    if sum(InputImg(ScanX,1:y)) > 0 & sw == 0
        x3 = ScanX;
        sw = 1;
    elseif sum(InputImg(ScanX,1:y)) == 0 & sw == 1
        x4 = ScanX + 1;
        turn = 0;
    end
    ScanX = ScanX - 1;
end

LenX = abs(x3-x4);
LenY = abs(y1-y2);
dia = mean([LenX LenY]);
xc = round((x3+x4)/2);
yc = round((y1+y2)/2);
Output = [xc yc LenX LenY dia];
```

รูปที่ 19 อัลกอริทึมที่ใช้ในการทดลอง

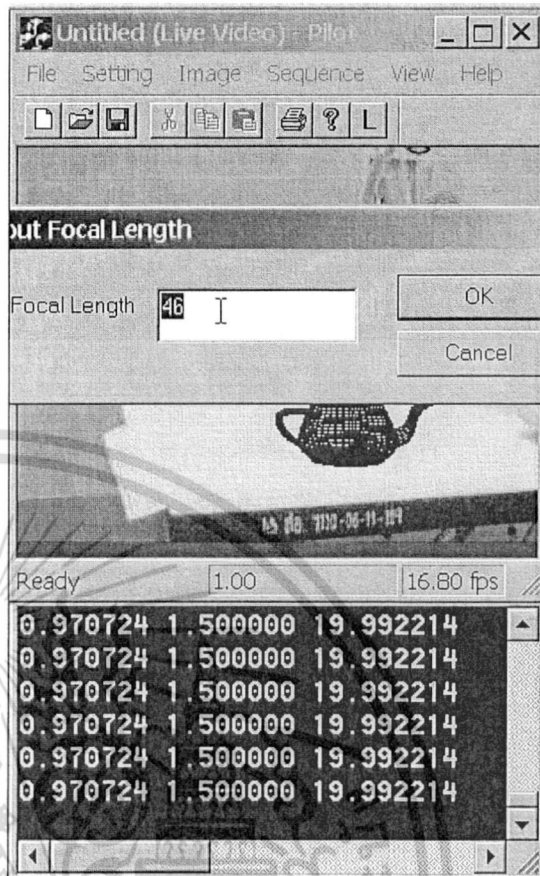
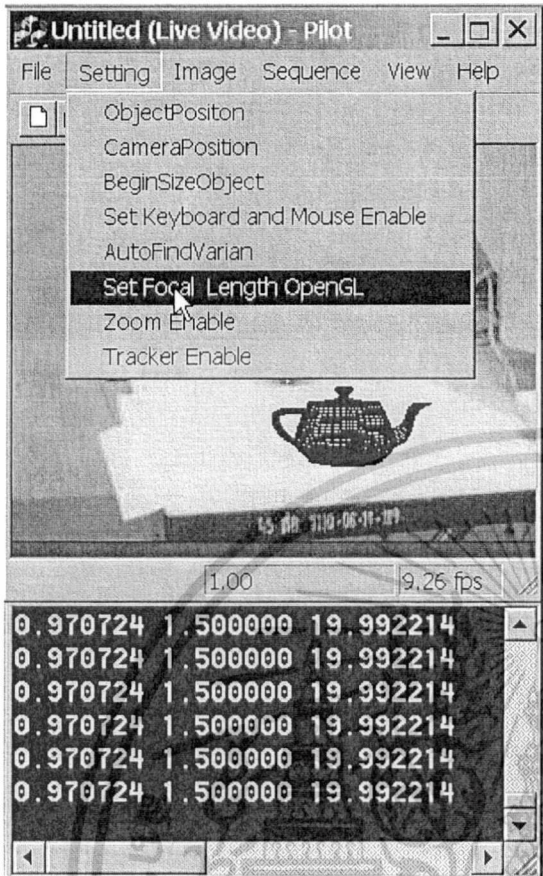
1) **ผลการทดลองเมื่อมีการเปลี่ยนเฉพาะความยาวโฟกัส**

เมื่อเริ่มต้นใช้งานหรือเริ่มการทดลอง จะต้องมีการกำหนดตำแหน่งและขนาดเริ่มต้นของวัตถุสังเคราะห์ก่อน รูปที่ 20 แสดงการกำหนดตำแหน่งและขนาดของวัตถุสังเคราะห์ให้เหมาะสม



รูปที่ 20 แสดงการเรียกใช้ฟังก์ชัน Set Keyboard สำหรับการกำหนดขนาด และ ตำแหน่งของวัตถุ
สังเคราะห์เริ่มต้นให้เหมาะสม

จากนั้นทำการกำหนดค่า Field of view (ความยาวโฟกัส) เริ่มต้นให้กับกล้องเสมือนใน
OpenGL รูปที่ 21 ก) แสดงการเลือกเมนูในการกำหนดค่า Field of view เริ่มต้น และ รูปที่ 21 ข)
แสดงการป้อนค่า Field of view (46 องศา ซึ่งเป็นค่าปกติของเลนส์ที่มีความยาวโฟกัสขนาด 35
มิลลิเมตร)

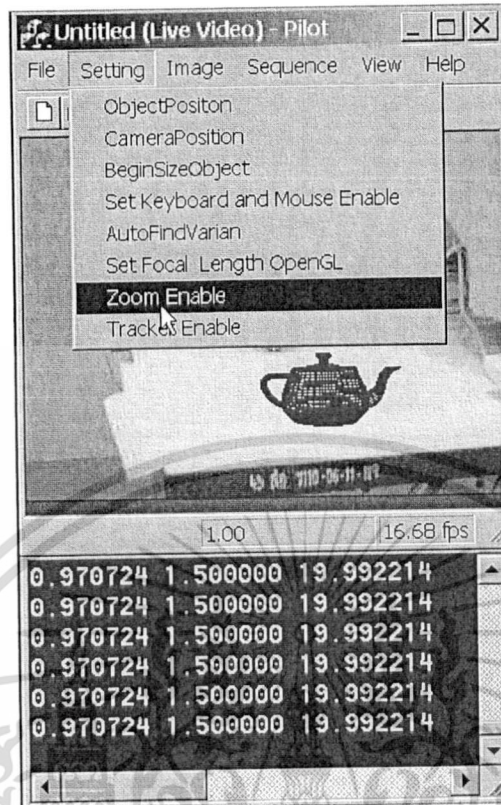


ก)

ข)

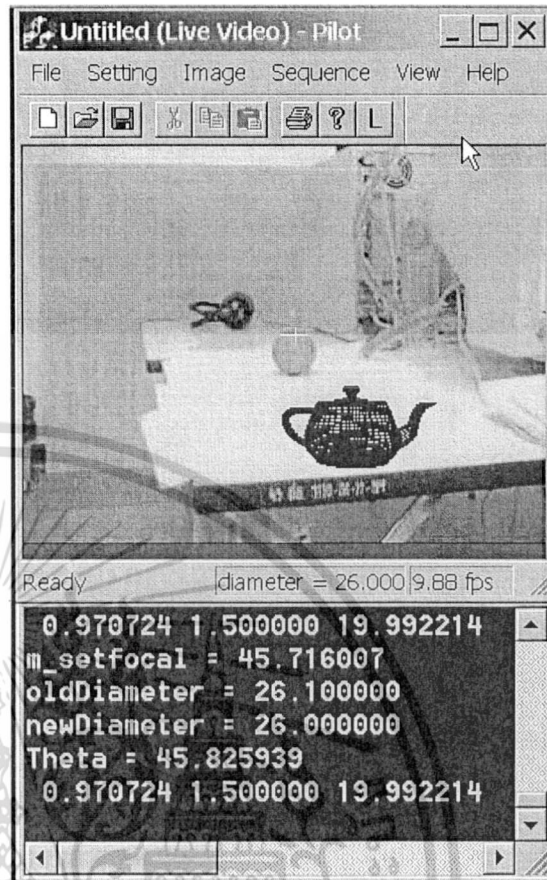
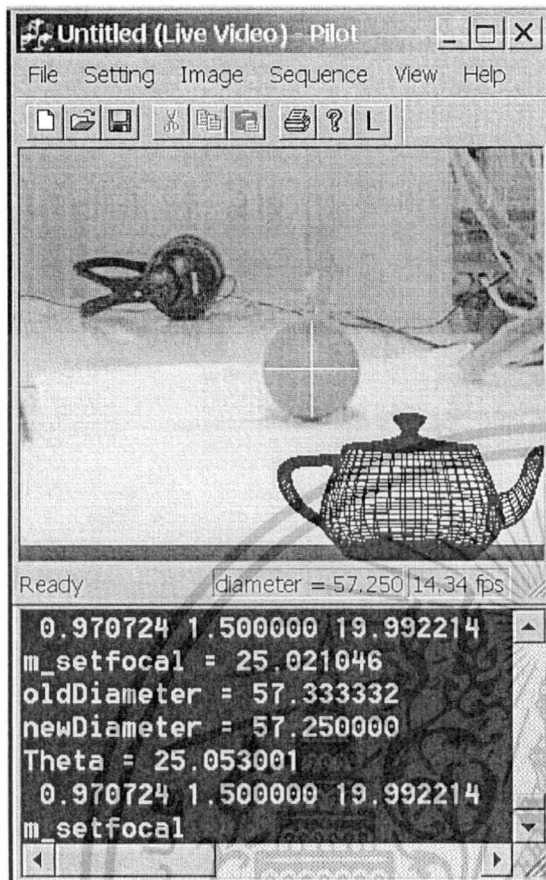
รูปที่ 21 การกำหนดค่าความยาวโฟกัสเริ่มต้น

ขั้นตอนต่อไปทำการสั่งให้ระบบทำงานในแบบ real-time เพื่อให้สามารถตอบสนองต่อการเปลี่ยนความยาวโฟกัสของกล้องวิดีโอ (ซึ่งเกิดจากการซูม) ได้ ตามรูปที่ 22



รูปที่ 22 แสดงการกำหนดให้โปรแกรมทำงานในแบบ Real-time เพื่อให้สามารถตอบสนองต่อการเปลี่ยนแปลงความยาวโฟกัส

ผลที่ได้จากการซุ่มกล้องออกปรากฏดังรูปที่ 23 ก) ภาพกาน้ำมีขนาดใหญ่ขึ้น และที่หน้าต่างล่างแสดง ค่าเส้นผ่าศูนย์กลางที่ตรวจพบมีขนาดใหญ่ ซึ่งมีผลทำให้ค่าความยาวโฟกัสมีขนาดมากขึ้น (แต่เนื่องจากค่าที่แสดงเป็นค่า Field of view มีค่าเป็นองศา และแปรผกผันกับค่าความยาวโฟกัส เพราะฉะนั้นจึงปรากฏว่า ค่า Field of view มีขนาดลดลง) ส่วนรูปที่ 23 ข) ให้ผลที่ตรงกันข้าม



ก)

ข)

รูปที่ 23 ผลที่ได้เมื่อมีการเปลี่ยนแปลงความยาวโฟกัสของกล้องวิดีโอ

ตารางที่ 1 แสดงผลในเชิงตัวเลขเปรียบเทียบค่าพารามิเตอร์ต่างๆ เมื่อมีการเปลี่ยนแปลงระดับการซูม

ค่า Parameter ต่างๆเมื่อมีการซูมออก	
OldDiameter	57.333332
NewDiameter	57.250000
Theta	25.053001
Frame rate	14.34 fps

ค่า Parameter ต่างๆเมื่อมีการซูมเข้า	
OldDiameter	26.100000
NewDiameter	26.000000
Theta	45.825939
Frame rate	9.88 fps

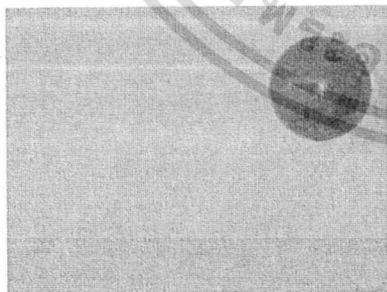
ตารางที่ 1 เปรียบเทียบผลเมื่อมีการเปลี่ยนแปลงความยาวโฟกัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางข้างต้น ค่า oldDiameter คือค่าเส้นผ่าศูนย์กลางของเฟรมก่อน และ ค่า newDiameter คือ ค่าเส้นผ่าศูนย์กลางของเฟรมปัจจุบัน (หน่วยเป็นพิกเซล) ในการทดลองได้ใช้การเฉลี่ยเชิงเวลา (moving average filter) ในการแก้ปัญหาความไม่เสถียรของค่าที่วัดได้ จะเห็นได้ว่า เมื่อมีการเพิ่มขนาดความยาวโฟกัส กล่าวคือ ขนาดของวัตถุจะมีขนาดใหญ่ขึ้น ตามรูป 23 ก) และ ค่าเส้นผ่าศูนย์กลางที่วัดได้มีขนาดใหญ่ด้วยคือ เท่ากับประมาณ 57 พิกเซล ตามตารางซ้ายมือ ซึ่งส่งผลให้ค่า Theta (Field of view) ที่คำนวณได้มีค่าลดลง ซึ่งถูกต้องตามทฤษฎีที่กล่าวมา และ ผลที่ได้ นี้เปรียบเทียบกับผลการลดความยาวโฟกัสในตารางด้านขวามือ

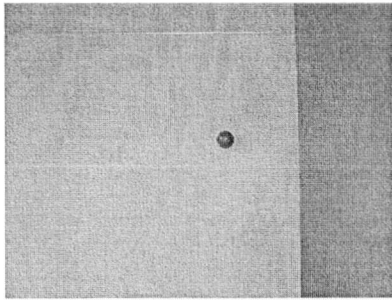
มีผลที่น่าสนใจอีกประการหนึ่งคือ ค่า frame rate (ในที่นี้คืออัตราการประมวลผลของอัลกอริทึมที่พัฒนาขึ้น มิใช่อัตราการถ่ายภาพของกล้อง) มีการเปลี่ยนแปลงเมื่อมีการลดขนาดความยาวโฟกัส กล่าวคือ เมื่อมีการลดขนาดความยาวโฟกัสรูปทรงกลมจะมีขนาดเล็กลงทำให้การตรวจหาต้องกระทำหลายรอบมากขึ้นจากอัลกอริทึมในหัวข้อที่ 3.31 frame rate ลดลงจาก 14.34 frame per second ไปเป็น 9.88 frame per second

การทดลองข้างต้นเป็นการทดสอบการทำงานในแบบ real-time จากการสังเกตด้วยสายตา พบว่าการเปลี่ยนแปลงขนาดของภาพสังเคราะห์ยังไม่สอดคล้องสัมพันธ์กับภาพจริงนัก ทั้งนี้เป็นผลเนื่องมาจากการประมวลผลข้อมูลบางช่วงไม่สามารถกระทำได้ในแบบ real-time และอีกส่วนหนึ่งอาจเป็นผลมาจากความผิดพลาดของตัวอัลกอริทึมเอง ดังนั้นเพื่อทดสอบความถูกต้องเฉพาะในส่วนของอัลกอริทึมที่ใช้ในการหาค่าระดับการซุม จึงได้ทำการทดลองเพิ่มเติมโดยใช้การถ่ายภาพนิ่งด้วยกล้องถ่ายภาพดิจิทัลที่ระดับการซุมที่ทราบค่าล่วงหน้า 2 ค่า เพื่อทดสอบความถูกต้องของวิธีการที่เสนอ รูปที่ 24 แสดงผลที่ได้จากการถ่ายภาพที่ค่าระดับการซุมที่มีความแตกต่างกัน 6 เท่า โดยจัดวางวัตถุทรงกลมไว้ในตำแหน่งต่าง ๆ กันในภาพ

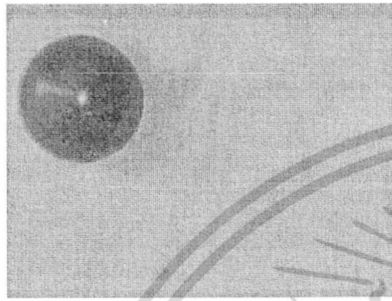


Xc	140
Yc	528
Horizontal Dia	175
Vertical Dia	168
Averaged Dia	171.5

รูปที่ 24 รูปภาพและผลการทดลองหาค่าความถูกต้องของวิธีการวัดค่าระดับการซุม



Xc	225
Yc	362
Horizontal Dia	29
Vertical Dia	28
Averaged Dia	28.5



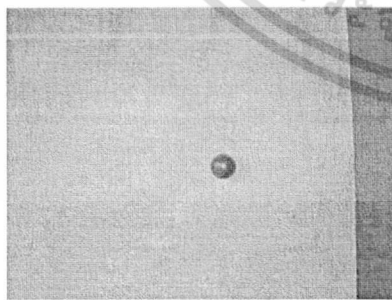
Xc	156
Yc	125
Horizontal Dia	208
Vertical Dia	203
Averaged Dia	205.5



Xc	230
Yc	292
Horizontal Dia	34
Vertical Dia	34
Averaged Dia	34



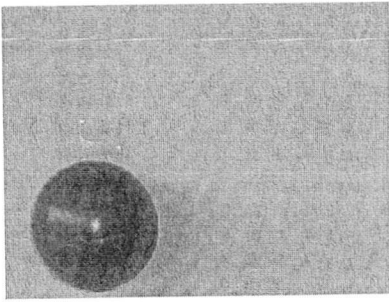
Xc	317
Yc	488
Horizontal Dia	243
Vertical Dia	239
Averaged Dia	241



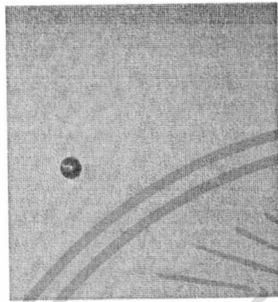
Xc	259
Yc	354
Horizontal Dia	41
Vertical Dia	41
Averaged Dia	41

รูปที่ 24 รูปภาพและผลการทดลองหาค่าความถูกต้องของวิธีการวัดค่าระดับการชุม (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Xc	359
Yc	150
Horizontal Dia	203
Vertical Dia	206
Averaged Dia	204.5



xc	264
yc	296
Horizontal Dia	34
Vertical Dia	33
Averaged Dia	33.5

รูปที่ 24 รูปภาพและผลการทดลองหาค่าความถูกต้องของวิธีการวัดค่าระดับการชุม (ต่อ)

จากรูปที่ 24 ภาพที่ใช้ในการทดลองทั้งแปดภาพ ทั้งหมดที่ถ่ายเป็นคู่ ต่างที่การวางตำแหน่งของวัตถุทั้งสี่มุมคือ N_E (North-East) N_W (North-West) S_E (South-East) และ S_W (South-West) ภาพทุกภาพวางบนขาตั้งกล้องจึงมีระยะห่างจากกล้องถึงวัตถุทรงกลมเท่ากัน จากการทดลองได้ค่าสัมพัทธ์ที่เทียบอัตราการชุมของภาพแต่ละคู่ที่ถูกบันทึกโดยใช้ระดับการชุมต่างกัน 6 เท่าดังนี้

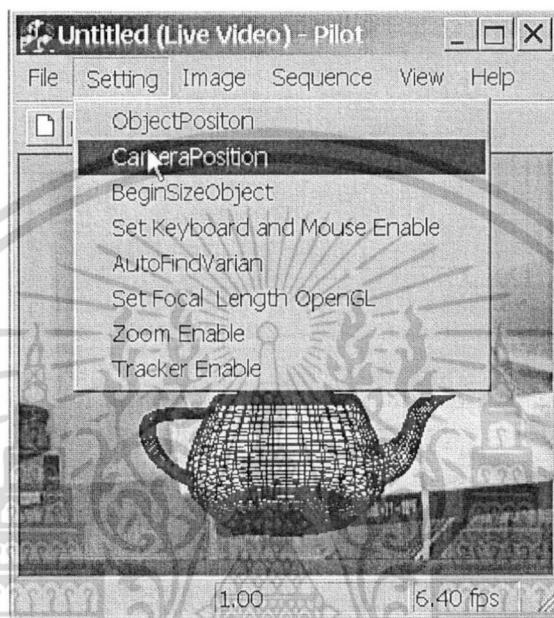
รูปที่	อัตราส่วนการชุม	ค่าความคลาดเคลื่อน
1-2	6.1754	2.9 %
3-4	6.0441	0.7 %
5-6	5.8780	2.0 %
7-8	6.1045	1.7 %
เฉลี่ย	6.0505	0.8 %

ตารางที่ 2 แสดงค่าความคลาดเคลื่อนในการวัดอัตราส่วนการชุม

3) ผลการทำงานเมื่อสั่งให้ระบบ tracker ทำงาน

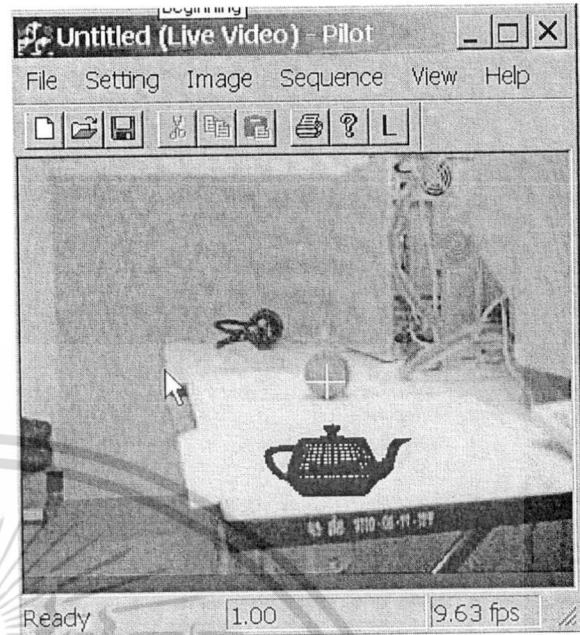
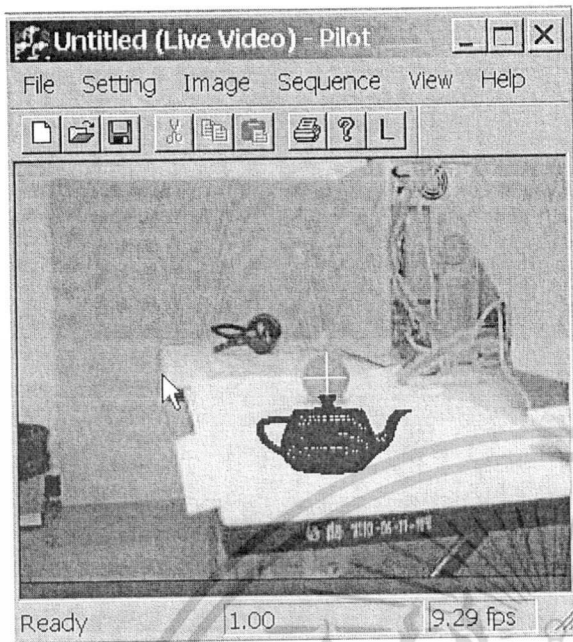
ในการทดลองนี้ได้นำค่าตำแหน่งของกล้องที่ได้จากอุปกรณ์ 6-DOF Tracking Device (ISOTRAK® II) มาประกอบในการสร้างภาพ โดยนำอุปกรณ์ Receiver ไปวางไว้ที่วัตถุทรงกลมที่ใช้เป็นตัววัดค่าระดับการชุม เพื่อบันทึกค่าตำแหน่งของวัตถุทรงกลมเทียบกับอุปกรณ์ Transmitter

ของ ISOTRAK II จากนั้นนำ Receiver ไปติดตั้งที่กล้องวิดีโอแล้วทำการบันทึกตำแหน่งเริ่มต้นซึ่งเป็นระยะห่างของกล้องเทียบกับ Transmitter (และวัตถุทรงกลม โดยอาศัยค่าตำแหน่งสัมพันธ์กับอุปกรณ์ Transmitter ที่ได้บันทึกไว้แล้วมาคำนวณ) ค่าที่ได้คือ D และ ค่า Dia ที่จะนำไปคำนวณหาความยาวโฟกัสจากสูตรที่กล่าวมาแล้วนั่นเอง



รูปที่ 25 แสดงการบันทึกค่าระยะห่างระหว่าง กล้องและวัตถุเริ่มต้น

เมื่อทำการเปิดระบบ Tracker ผลที่ได้แสดงดังในรูปที่ 26 จากรูปตำแหน่งของกาน้ำซึ่งเป็นวัตถุสังเคราะห์มีการเปลี่ยนแปลงตำแหน่งตลอดเวลา ทุ้่งๆ กล้องถ่ายวิดีโออยู่นิ่งบนขาตั้งกล้อง ซึ่งสาเหตุมาจากการที่อุปกรณ์ ISOTRAK®II ส่งค่าที่มีการเปลี่ยนแปลงตลอดเวลา สังเกตได้จากหน้าต่างด้านล่างของรูปที่ 26 ค่าที่ได้มีการเปลี่ยนแปลงตลอดเวลาทุ้่งๆที่ กล้องถ่ายวิดีโออยู่บนขาตั้งกล้องตลอดเวลา เพราะฉะนั้นค่านี้จะถูกส่ง ไปให้กล้องเสมือนใน OpenGL ซึ่งจะทำให้กล้องเสมือนมีการเคลื่อนที่ตลอดเวลาซึ่งทำให้ภาพต่อเนื่องที่แสดงออกมามีความผิดพลาด ทั้งนี้เป็นผลมาจากการที่ระบบ Tracker ที่ใช้เป็นประเภทใช้คลื่นแม่เหล็กไฟฟ้า ทำให้ได้รับผลกระทบจากสัญญาณรบกวนจากกล้องวิดีโอและจอมอนิเตอร์ของเครื่องคอมพิวเตอร์ที่อยู่ข้างเคียง



```

D:\Projects\Copy of proj.\Debug\Pilot.exe
X -157.406067 Y -30.421892 Z 4.946381 RX 1.730583 RY 0.555400 RZ 2.487939
0.000000 1.500000 10.000000
X -157.469528 Y -29.714449 Z 6.856156 RX 1.746208 RY 0.552666 RZ 2.516101
0.000000 1.500000 10.000000
X -157.124374 Y -31.640720 Z 8.902978 RX 1.778289 RY 0.553374 RZ 2.546815
0.000000 1.500000 10.000000
X -156.810303 Y -32.679359 Z 8.695504 RX 1.804193 RY 0.553850 RZ 2.562221
0.000000 1.500000 10.000000
X -156.557785 Y -33.729420 Z 10.465694 RX 1.838459 RY 0.557341 RZ 2.591434
0.000000 1.500000 10.000000
X -156.449921 Y -33.681198 Z 11.828550 RX 1.854915 RY 0.553862 RZ 2.609806
  
```

รูปที่ 26 ผลลัพธ์ที่ได้จากการเปิดระบบ Tracker

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและแนวทางการพัฒนาในอนาคต

จากการทดลอง สามารถสรุปผลได้ดังต่อไปนี้

1. ค่าที่วัดได้จากการหาเส้นผ่านศูนย์กลางมีความแปรปรวนในระดับหนึ่งจะทำให้ค่าความยาวโฟกัสที่ได้ไม่มีเสถียรภาพเพียงพอ ทำให้แม็กล้องไม่ได้มีการเปลี่ยนแปลงความยาวโฟกัส พบว่าภาพของวัตถุสังเคราะห์จะมีการเปลี่ยนแปลงขนาด โดยเป็นผลมาจากสัญญาณรบกวนต่าง ๆ เช่น สัญญาณรบกวนในเซ็นเซอร์รับแสงของกล้อง สภาพแสงภายในห้องทดลองมีการเปลี่ยนแปลง เป็นต้น ซึ่งในการพัฒนาระบบงานนี้ เราแก้ไขด้วยวิธีการเฉลี่ยเชิงเวลา พบว่าแก้ปัญหาได้ในระดับหนึ่ง แต่มีข้อเสียที่ตามมาคือ frame rate ในการแสดงผลจะตกลงซึ่งเป็นข้อควรคำนึงที่สำคัญอย่างหนึ่งในงานแบบ real-time จากการทดลองพบว่าค่าการเฉลี่ยเชิงเวลาที่เหมาะสมคือ ประมาณ 5 ค่า (เฟรม)
2. ปัญหาที่เกิดจากอุปกรณ์ ISOTRAK®II ให้ค่าไม่คงที่ สาเหตุน่าจะมาจากหลักการทำงานของอุปกรณ์ที่ใช้คลื่นแม่เหล็กไฟฟ้า ดังนั้นการใช้อุปกรณ์หลายๆ อย่างในห้องปฏิบัติการวิจัยอาจทำให้เกิดการรบกวนกันของสัญญาณได้

ข้อเสนอแนะเพื่อการปรับปรุงแก้ไขในอนาคตมีดังนี้

1. ศึกษาทดลองใช้ระบบการหาตำแหน่งระบบอื่นๆ เช่น ระบบที่ใช้แสงพิเศษในการติดตามตำแหน่ง
2. วิธีการที่นำเสนอต้องอาศัยกระบวนการประมวลผลสัญญาณภาพสูง ทำให้ต้องเสียเวลาในการคำนวณ ดังนั้นต้องอาศัยอุปกรณ์ เช่น คอมพิวเตอร์ที่มีประสิทธิภาพสูง เป็นต้น โดยแนวทางแก้ปัญหา มีอาทิ เช่น การใช้คอมพิวเตอร์หลายเครื่องมาช่วยในการทำงานโดยการประมวลผลแบบกระจาย เช่น แบ่งหน้าที่ของคอมพิวเตอร์กันทำงาน ให้เครื่องหนึ่งทำหน้าที่รับภาพจากกล้องและคำนวณระดับการซูม เครื่องที่ 2 ทำหน้าที่สร้างวัตถุสามมิติ และเครื่องสุดท้ายทำหน้าที่รวมภาพและแสดงผล เป็นต้น

เอกสารอ้างอิง

- [1] R. T. Azuma, "A Survey of Augmented Reality." *Teleoperators and Virtual Environments*. vol. 6, no. 4, 1997, pp. 355-385.
- [2] K. N. Kutulakos, and J. R. Vallino, "Calibration-Free Augmented Reality." *IEEE Transaction on visualization and computer graphics*, vol. 4, no. 1, 1998, pp. 1-20.
- [3] G. Shivaram, and G. Seetharaman, *A New Technique for Finding the Optical Center of Cameras*. [Online]. Available: <http://citeseer.nj.nec.com/345537.html>
- [4] Vision Technology Group, Microsoft Research. *The Microsoft Vision SDK Version 1.2*, May 2000. [Online]. Available: <http://research.microsoft.com/projects/VisSDK/VisSDK.doc>.
- [5] G. Baxes, *Digital Image Processing : Principles and Applications*, John Wiley & Sons, 1994.
- [6] D. J. Kruglinski, *Inside Visual C++*. 4th Edition, Microsoft Press, 1997.
- [7] S. Gibbs, C. Arapis, C. Breiteneder, V. Lalioti, S. Mostafawy, J. Speier, "Virtual studios: An overview", *IEEE Multimedia*, vol.5, no.1, 1998, pp.18- 35.
- [8] L. Blonde, M. Buck, R. Galli, W. Niem, Y. Paker, W. Schmidt, G. Tomas, "A virtual studio for live broadcasting: The Mona Lisa project", *IEEE Multimedia*, vol.3, no.2, 1996, pp. 18- 29.
- [9] T. S. Huang, A. N. Netravali, "Motion and structure from feature correspondences: A review", *Proc. IEEE*, vol.82, 1994, pp.252- 269.
- [10] Y. S. Xirouhakis, A. I. Drosopoulos, A.I, and A. N. Delopoulos, "Efficient optical camera tracking in virtual sets, " *IEEE Trans. Image Processing*, Vo. 10. Issue 4, April 2001, pp. 609 -622.
- [11] Jun-Sik Kim; and In-So Kweon, "A new camera calibration method for robotic applications," *Proc. 2001 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Vol. 2, 2001, pp. 778 -783.
- [12] E. Malis, and R. Cipolla, "Self-calibration of zooming cameras observing an unknown planar structure," *Proc.15th Int. Conf. Pattern Recognition*, Vol. 1 , 2000, pp. 85 -88.
- [13] Y. Seo, and K.S. Hong, "Theory and practice on the self-calibration of a rotating and zooming camera from two views, " , *IEE Proceedings on Vision, Image and Signal Processing*, Vol.148 Issue 3 , June 2001, pp.166 -172.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และ 31อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [14] G. Shivaram, and G. Seetharaman, "A new technique for finding the optical center of cameras," Proc. Int. Conf. Image Processing, Vol. 2, 1998, pp. 167 -171.
- [15] Won Pil Yu, Young Jin Lee, and Myung Jin Chung , "Self calibration of focal length and rotation parameters using cross ratio, " Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, Vol. 3, 1998, pp. 1828 -1833.

