สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การพัฒนาสื่อแสดงผลในรูปแบบเสมือนจริงผสม

Implementation of Mixed Reality Presentation Media

นายคีตะพล บุญประจักษ์
นางสาวภาวิณี ลิขสิทธิ์ศุภการ

เลขหมู่.....
เลขทะเบียน 62445
วัน,เดือน,ปี 18 ส.ค. 2549

b 11625387
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาสื่อแสดงผลในรูปแบบเสมือนจริงผสม

Implementation of Mixed Reality Presentation Media

โดย

นายคีตะพล บุญประจักษ์

นางสาวภาวิณี ลิขสิทธิ์ศุภการ

อาจารย์ที่ปรึกษา

ดร.สมศักดิ์ วลัยรัชต์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2548

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาสื่อแสดงผลในรูปแบบเสมือนจริงผสม

Implementation of Mixed Reality Presentation Media

ผู้จัดทำ

1. นายคีตะพล บุญประจักษ์ รหัสนักศึกษา 46015338
2. นางสาวภาวิณี ลิขสิทธิ์ศุภการ รหัสนักศึกษา 46015365

_____ อาจารย์ที่ปรึกษา

(ดร.สมศักดิ์ วลัยรัชต์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาสื่อแสดงผลในรูปแบบเสมือนจริงผสม

นายคีตะพล บุญประจักษ์ 46015338
นางสาวภาวิณี ลิขสิทธิ์ศุภการ 46015365
ดร.สมศักดิ์ วลัยรัชต์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2548

บทคัดย่อ

ในวิทยานิพนธ์ฉบับนี้ เสนอการออกแบบและพัฒนาโปรแกรม เพื่อใช้เป็นสื่อในการนำเสนอผลงานในรูปแบบเสมือนจริงผสม โดยใช้เทคนิค Chroma Keying สำหรับการนำภาพที่ถ่ายจากฉากหลังสีน้ำเงิน มารวมกับสภาพแวดล้อม 3 มิติที่จำลองขึ้นมา (Three-dimensional environment) มีการนำเทคโนโลยีของ VideoOCX เข้ามาประยุกต์ใช้ในการประมวลผลเก็บข้อมูลภาพถ่ายวิดีโอ ในส่วนของโปรแกรม Virtual C#.NET ถูกใช้เป็นเครื่องมือพัฒนา ที่ออกแบบให้ทำงานตามสถาปัตยกรรมของ .NET Framework ในการทดลอง ระบบสามารถรวมภาพวิโอวัตถุจริงเข้าไปในสภาพแวดล้อมกราฟิกแบบ 3 มิติ ได้อย่างถูกต้อง สอดคล้องกับตำแหน่งต่างๆของวัตถุในสภาวะแวดล้อมที่สร้างขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Implementation of Mixed Reality Presentation Media

Mr. Keetapol Boonprachak 46015338
Miss Pawinee Likasitsupakarn 46015365
Dr. Somsak Walairacht Advisor

Academic Year 2004

ABSTRACT

This thesis proposes an implementation of Mixed Reality Presentation Media System. The chroma keying technique is used to extract video images of real objects from the blue screen background. VideoOCX is employed in the process of video image acquisition. The system is implemented by using Visual C#.NET. In the experiments, we show that a real-world object can be correctly merged into three-dimensional virtual world according to the spatial locations.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้อย่างดี ด้วยคำแนะนำ และคำปรึกษาจาก ดร. สมศักดิ์ วลัยรัชต์ ซึ่งเป็นอาจารย์ผู้ควบคุมวิทยานิพนธ์ ข้าพเจ้ารู้สึกทราบซึ้งในความอนุเคราะห์จากท่านอาจารย์ และขอขอบพระคุณเป็นอย่างสูง

ขอกราบขอบพระคุณคณาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุก ๆ ท่านที่ได้ประสิทธิ์ประสาทวิชาให้กับข้าพเจ้า

ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ในภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกคนที่ให้คำแนะนำต่างๆ และคอยให้กำลังใจเสมอมา

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็นกำลังใจ และให้การสนับสนุนในทุกๆ เรื่อง ทำให้ข้าพเจ้าสามารถทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี

คุณค่าและประโยชน์อันพึงมาจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบแด่ผู้มีพระคุณทุกท่าน

คีตะพล บุญประจักษ์

ภาวิณี ลิขสิทธิ์ศุภการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VI

สารบัญรูป VII

บทที่ 1 บทนำ 1

1.1 ความสำคัญและที่มาของโครงงาน 1

1.2 วัตถุประสงค์ของโครงงาน 1

1.3 ขอบเขตของโครงงาน 2

1.4 วิธีการดำเนินการ 2

1.5 ประโยชน์ที่คาดว่าจะได้รับ 2

1.6 ส่วนประกอบของปริญญานิพนธ์ 2

บทที่ 2 ทฤษฎีพื้นฐานที่ใช้ในการพัฒนาระบบ 4

2.1 Color Model 4

2.1.1 RGB Model 4

2.1.2 HSL Model 7

2.2 Software Chroma Keying 8

2.2.1 Virtual Environment 9

2.2.2 Chroma Keying Exact Solutions 11

2.2.3 *Chroma Keying: Approximation* 12

2.2.4 Algorithm 12

2.2.5 Hue-base approaches 13

2.2.6 An improved approach 14

2.2.7 Image quality 17

2.2.8 Performance 19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.3 VideoOCX 20

2.3.1 ความต้องการพื้นฐานสำหรับการใช้งาน VideoOCX 20

2.3.2 ความสามารถของ VideoOCX 20

2.3.3 Method ต่างๆสำหรับใช้งาน VideoOCX 21

2.4 Visual C#.NET 26

บทที่ 3 การออกแบบและพัฒนา 28

3.1 การออกแบบระบบโดยภาครวม 28

3.2 การนำภาพจากกล้องเข้ามาใช้งาน 29

3.3 การนำภาพฉากเข้ามาใช้งาน 30

3.4 การนำภาพมาผสมผสานกัน 31

บทที่ 4 การใช้งานระบบ 33

4.1 ติดตั้งอุปกรณ์ฮาร์ดแวร์ของระบบ 33

4.2 การนำภาพจากไฟล์เข้ามาใช้ในโปรแกรม 33

4.3 การกำหนดค่าพารามิเตอร์ต่างๆเกี่ยวกับกล้อง 36

4.3.1 *Video Format* 36

4.3.2 *Video Source* 36

4.3.3 *Video Driver* 37

4.4 การบันทึกภาพเป็นไฟล์วิดีโอ 37

4.5 การนำภาพถ่ายจากกล้องวิดีโอเข้ามาใช้งาน 39

บทที่ 5 บทวิจารณ์และสรุป 41

5.1 บทสรุป 41

5.2 วิจารณ์สิ่งที่ได้จากโครงงาน 41

5.3 ปัญหาอุปสรรคและแนวทางแก้ไข 41

5.4 แนวทางการพัฒนาต่อ 42

บรรณานุกรม 43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

2.1 Software HLS vs Improved Algorithm, FPS 19

2.2 Software HLS vs Improved Algorithm, Mp/s 19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

2.1 RGB cube 6

2.2 Additive Color 7

2.3 Subtractive color 7

2.4 HSL Color Model 8

2.5 A model of the Responsive Workbench 9

2.6 Collaborative Responsive Workbench 10

2.7 HSL hexagon 13

2.8 C++ code for the HSL algorithm 14

2.9 An RGB cube showing the skew pyramid OBTPQ 15

2.10 An RGB cube showing a plane S intersecting the skew pyramid OBTPQ 15

2.11 An RGB cube showing a plane S of constant d 16

2.12 Two sample alpha functions 17

2.13 Two sample alpha functions 17

2.14 Original Image 18

2.15 a) Hardware keyed, b) Software keyed 18

2.16 Blow-up of elbow region: a) Hardware keyed, b) Software keyed 19

2.17 แสดงหน้าต่าง Video Format 23

2.18 แสดงหน้าต่าง Video Source 24

2.19 แสดงหน้าต่าง Driver Selection 24

2.20 แสดงหน้าต่างบันทึก 25

2.21 แสดงหน้าต่าง Choose Code 25

2.22 แสดงหลักการทำงานของตัวแปลภาษา Visual C# 26

3.1 แสดงภาพฉากหลังที่เตรียมไว้ 28

3.2 แสดงภาพฉากหน้า 28

3.3 แสดงภาพรวมของการนำมาผสมผสานกัน 29

3.4 แสดงการนำภาพจากกล้องเข้ามาใช้งาน 29

3.5 แสดงภาพฉากหลัง 30

3.6 แสดงภาพ Depth-key ของพื้น และลักษณะของมุมกล้อง 30

3.7 แสดงภาพ Depth-key ของวัตถุในฉาก 31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

3.8 แสดงกระบวนการผสมผสานภาพในหนึ่งเฟรม 32
4.1 แสดงภาพอุปกรณ์ที่ใช้งาน 33
4.2 เมนูสำหรับเลือก folder 34
4.3 เลือก folder สำหรับใช้เป็น background 34
4.4 แสดงเมนูเช็คเพื่อ capture from file 35
4.5 เลือกไฟล์ที่นำมาเป็นฉากหน้า 35
4.6 แสดงหน้าต่างการปรับค่า Video Format 36
4.7 แสดงหน้าต่างการปรับค่า Video Source 36
4.8 แสดงหน้าต่างการเลือก Video Driver 37
4.9 แสดงเมนูเช็คเพื่อ record 37
4.10 แสดงหน้าต่างสำหรับตั้งชื่อไฟล์ที่จะบันทึก 38
4.11 แสดงหน้าต่างปรับค่าความละเอียดและวิธีการบีบอัดภาพ 38
4.12 แสดงภาพ Preview ที่ได้จากกล้อง 39
4.13 แสดงการรวมภาพจากไฟล์ผสมกับภาพจากกล้อง 39
4.14 แสดงการผสมผสานภาพในกรณีต่างๆ 40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของโครงงาน

เนื่องจากในปัจจุบัน เทคโนโลยีด้านสารสนเทศเข้ามามีบทบาทในชีวิตประจำวันอย่างมาก และมีการการประยุกต์ใช้งานเทคโนโลยีทางด้านสารสนเทศกันมากขึ้น การนำเสนอผลงานโดยใช้เทคโนโลยีก็เป็นอีกแนวความคิดที่ได้รับความนิยม เช่น การทำโฆษณา การทำรายการโทรทัศน์ และยังถูกนำมาใช้งานในหน่วยงานอีกหลายๆ หน่วยงาน เช่น หน่วยงานทางด้านการศึกษา เป็นต้น การนำเสนอผลงานโดยใช้สื่อที่น่าสนใจนั้นให้ผลประโยชน์ต่างๆมากมายทั้งทางด้านธุรกิจ การศึกษา ถึงแม้ว่าจะมีการใช้งานกันอย่างแพร่หลาย แต่ก็ยังมีปัญหาในด้านของอุปกรณ์ และความชำนาญเพื่อให้สื่อการนำเสนอที่ได้มีความสมจริง รวมทั้งอุปกรณ์ที่นำมาใช้งานนั้นก็ต้องเป็นอุปกรณ์ที่มีประสิทธิภาพ และมีความละเอียดในการทำงานที่สูงด้วยเช่นกัน ดังนั้นหากสามารถพัฒนาระบบให้มีความสมจริงนั้น จึงตามมาด้วยการลงทุนที่สูง

ด้วยเหตุผลเหล่านี้จึงทำให้หน่วยงานต่างๆ ภายในประเทศ ไม่ได้นำเทคโนโลยีเหล่านี้มาใช้ให้เกิดประโยชน์เท่าที่ควร ดังนั้นหากสามารถพัฒนาระบบที่สามารถสร้างสื่อการนำเสนอผลงานที่นำตัวผู้บรรยายเข้าไปอยู่ในสภาพแวดล้อมจำลองได้อย่างเหมาะสมมีความสวยงามและดูเป็นธรรมชาติ และมีต้นทุนที่ถูกได้ จะทำให้มีการนำเทคโนโลยีนี้มาใช้งานในหน่วยงานต่างๆมากขึ้น ซึ่งจะทำให้เกิดความก้าวหน้าทางด้านเทคโนโลยีตามมาด้วย

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อพัฒนาระบบบันทึกและประมวลผลภาพที่สามารถนำไปใช้เป็นตัวละครในสภาพแวดล้อมจำลองได้
2. เพื่อนำภาพตัวละครที่ได้จากการประมวลผลมารวมกับสภาพแวดล้อมจำลองที่สังเคราะห์ขึ้นมาได้
3. เพื่อเสริมสร้างความก้าวหน้าในการนำเทคโนโลยีมาประยุกต์ใช้ในการนำเสนอผลงาน
4. เพื่อส่งเสริมการพัฒนาเทคโนโลยีที่สามารถใช้งานได้ โดยใช้ต้นทุนต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของโครงงาน

พัฒนาระบบการสื่อการนำเสนอผลงานในสภาวะแวดล้อมผสม ซึ่งเป็นระบบที่นำมาใช้เพื่อสามารถสร้างสื่อการนำเสนอผลงานโดยสามารถประมวลผลภาพตัวละครที่ได้จากการถ่ายในฉากบลูสกรีน และสามารถสร้างสภาพแวดล้อมจำลองที่จะนำมารวมกับภาพตัวละครที่ได้จากการประมวลผล แล้วสามารถบันทึกเก็บไว้ได้ในรูปแบบของไฟล์วิดีโอ และสามารถแสดงการบรรยายสดในสภาวะแวดล้อมผสมได้

1.4 วิธีดำเนินการ

1. ศึกษาเกี่ยวกับกระบวนการการประมวลผลภาพ
2. ศึกษาโครงสร้างของไฟล์ที่ใช้บันทึก
3. ศึกษาเทคนิคและเทคโนโลยีที่นำมาใช้ในการพัฒนาระบบ
4. จัดหาวัสดุอุปกรณ์ที่จำเป็นต้องใช้ในการพัฒนา
5. วิเคราะห์และออกแบบระบบ
6. พัฒนาโปรแกรมในส่วนของการประมวลผลภาพและการสร้างสภาพแวดล้อมจำลอง
7. พัฒนาโปรแกรมในส่วนการรวมภาพตัวละครกับสภาพแวดล้อมจำลอง
8. พัฒนาโปรแกรมที่ใช้ในการบันทึกเพื่อเก็บไว้มนรูปแบบของไฟล์
9. วิเคราะห์ระบบและแก้ไขข้อผิดพลาดเพื่อทำให้สื่อมีความสวยงามและดูสมจริงมากที่สุด

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้รับความรู้ความเข้าใจเกี่ยวกับกระบวนการประมวลผลภาพ
2. ได้รับความรู้ ความเข้าใจเกี่ยวกับกระบวนการทำภาพเคลื่อนไหว
3. ได้รับความรู้ ความเข้าใจเกี่ยวกับกระบวนการนำภาพมาผสมผสานกัน
4. ได้รับความรู้ ความเข้าใจ เกี่ยวกับกระบวนการสร้างสื่อการนำเสนอ ในรูปแบบสภาวะแวดล้อมผสม
5. สามารถสร้างสื่อการนำเสนอผลงานในสภาวะแวดล้อมผสมที่มีต้นทุนต่ำ

1.6 ส่วนประกอบของปริญญานิพนธ์

ปริญญานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 5 บทด้วยกันคือ

บทที่ 1 กล่าวถึงความสำคัญและที่มาของโครงงาน วัตถุประสงค์ของโครงงาน ขอบเขตของโครงงาน วิธีการดำเนินการ ประโยชน์ที่คาดว่าจะได้รับ และส่วนประกอบของปริญญานิพนธ์

เอกสารนี้มีลิขสิทธิ์ ห้ามนำไปใช้ในการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2 กล่าวถึงทฤษฎีพื้นฐานที่ใช้ในโครงงาน ซึ่งประกอบด้วยทฤษฎีอะไรบ้าง ให้บรรยายทฤษฎีทั้งหมดโดยละเอียด ได้แก่

2.1. ความรู้เบื้องต้นเกี่ยวกับ Color Model อธิบายถึงลักษณะโดยทั่วไปของ Model เพื่อนำมาใช้ในการพัฒนาโปรแกรม

2.2. อธิบายถึงเทคนิคการพัฒนา Software Chroma keying และการนำมาประยุกต์ใช้

2.3. รายละเอียดคุณสมบัติของเทคโนโลยี VideoOCX ที่นำมาประยุกต์ใช้ร่วมกันกับการพัฒนาระบบ

2.4. อธิบายถึงคุณลักษณะโดยทั่วไปของเครื่องมือที่ใช้ในการพัฒนา(Visual C#.NET)

บทที่ 3 อธิบายส่วนของการออกแบบและพัฒนาระบบนี้ซึ่งประกอบไปด้วย 3 ส่วนใหญ่ๆ คือ ส่วนของการนำภาพมาประมวลผล, ส่วนของการรวมภาพกับสภาพแวดล้อมจำลอง และส่วนของการจัดเก็บในรูปแบบไฟล์

บทที่ 4 กล่าวถึงการทดลองการติดตั้งการใช้งานและผลการทดลองของระบบ เพื่อแสดงผลที่ได้จากโครงงาน

บทที่ 5 เป็นการสรุปโครงงานทั้งหมดและข้อเสนอแนะสำหรับเป็นแนวทางในการพัฒนาต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีพื้นฐานที่ใช้ในโครงงาน

ในหัวข้อนี้จะกล่าวถึงทฤษฎีพื้นฐานต่างๆ ที่เกี่ยวข้องในการทำโครงงาน ซึ่งเนื้อหาในบทนี้จะกล่าวถึงเทคนิคและทฤษฎีขั้นพื้นฐาน ซึ่งเนื้อหาทั้งหมดนี้จำเป็นสำหรับการศึกษา เพื่อนำไปใช้ในการพัฒนาโครงงาน

2.1 Color Model

Color model ใช้สำหรับอ้างอิงถึงสีต่างๆโดยไม่ใช้ analytical model เหมือนกับที่ใช้ทางวิทยาศาสตร์ซึ่งใช้วิธีการวัดให้อยู่ในรูปของพลังงานตลอดช่วงของสเปคตรัมแต่จะเป็น empirical model ที่ได้ความสัมพันธ์ของค่าที่ใช้อ้างอิงกับสีใดๆ จากการทดลองที่ศึกษาแบบ phychophysical ที่มีการรับรู้ของมนุษย์เข้ามาเกี่ยวข้องสื่อสารนั้น โดยมีรายละเอียด model แบบต่างๆ ดังนี้

2.1.1 RGB Model

RGB Model เป็น color model ที่เฉพาะเจาะจงกับอุปกรณ์ใดอุปกรณ์หนึ่ง ซึ่งในที่นี้ก็คือจอภาพตัวใดตัวหนึ่งของคอมพิวเตอร์เท่านั้น เนื่องจาก RGB model ได้ทำการสร้างสีต่างๆขึ้นโดยการใช้แหล่งกำเนิดแสงจำนวนสามสีได้แก่ Red Green Blue ที่เกิดจากสารเรืองแสงที่มีคุณสมบัติที่แตกต่างกันตามลำดับ ซึ่ง Red Green และ Blue นี้ก็จะไม่เท่ากันในแต่ละอุปกรณ์นอกเสียจากว่าจะมีคุณสมบัติของสารเรืองแสงและการ setup จอและสภาพแวดล้อมที่จอภาพตั้งอยู่ที่เหมือนกันทุกประการ ซึ่งโดยปกติแล้วก็จะมีความแตกต่างกันออกไป

การที่เราเห็นภาพจากจอภาพเหล่านั้นได้ก็เพราะการทำงานของคลื่นแม่เหล็กไฟฟ้าเรียงตัวกันตามลำดับ (แดง-เขียว-น้ำเงิน) หรือที่เรียกว่า Spectrum โดยอัตราของสีที่แตกต่างกัน และเมื่อทับซ้อนกัน (Colors Overlap) ก็จะเกิดเป็นสีอื่นๆ อีกมากมาย

เมื่อทำการแสดงสีที่มีค่าเท่ากับ [1.0,0.0,0.0] ซึ่งเป็นสีแดงจะหมายถึงสีแดงของจอภาพนั้นๆ และเมื่อนำไปเปรียบเทียบกับ [1.0,0.0,0.0] จอภาพอื่นๆก็จะได้สีแดงที่มีค่าไม่เท่ากับจอภาพสีแดงตัวแรก ดังนั้นเมื่อทำการเปรียบเทียบสี RGB ของจอภาพมากกว่าสองตัวขึ้นไปจะต้องเปรียบเทียบโดยการใช้ XYZ Model ที่ไม่ขึ้นกับอุปกรณ์ โดยการแปลง RGB ของจอภาพตัวที่หนึ่งไปเป็น XYZ จากนั้นจึงทำการแปลง RGB ของจอภาพตัวที่สองอีกครั้งที่ต้องทำเช่นนี้เนื่องจากไม่สามารถที่จะทำการเปรียบเทียบระหว่างจอภาพทั้งสองได้โดยตรง เนื่องจากไม่มีข้อมูลความสัมพันธ์ของสีระหว่างจอภาพทั้งสองแต่เรามีความสัมพันธ์ของข้อมูลสีของจอภาพแต่ละตัวกับระบบสี XYZ ที่เป็นมาตรฐานซึ่งค่าดังกล่าวสามารถทราบได้จากผู้ผลิตจอภาพนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นระบบสี RGB จึงเป็นระบบสีที่ขึ้นอยู่กับอุปกรณ์แสดงผล(device dependent) อีกทั้งยังไม่สามารถที่จะแสดงสีบนสเปคตรัมได้ทั้งหมดแต่ก็นับว่ามี color gamut ที่เพียงพอในการคำนวณในงานคอมพิวเตอร์กราฟริกสามมิติ เนื่องจากอยู่ในรูปของ intensity ซึ่งสมการในการคำนวณของ illumination model จะคำนวณในรูปของ intensity แต่การกำหนด color model แบบ RGB นี้ก็ใช่ว่าจะเหมาะสมกับทุกกรณีเสมอไปจึงได้มีการคิดค้น color model อื่นที่ช่วยให้มีความเหมาะสมกับงานอื่นๆมากยิ่งขึ้น

ระบบสี RGB ประกอบด้วยข้อมูลจำนวนสามส่วนคือค่า intensity ของสีใน 3 channel ได้แก่ Red Green และ Blue ตามลำดับ โดยสีดังกล่าวก็จะแตกต่างกันไปในแต่ละจอภาพดังที่กล่าวไว้ข้างต้น การที่เราจะสร้างสีหรือ reproduce สีต่างๆ โดยการใช้สี RGB ดังกล่าวนี้สามารถที่จะกระทำได้โดยการแสดงสีทั้งสามสีที่มี intensity ที่แตกต่างกันออกไปทำให้เรามองเห็นเป็นสีต่างๆ เนื่องจาก RGB ที่มี intensity ที่แตกต่างกันนี้จะไปกระตุ้น cone cell แต่ละชนิดมากน้อยต่างกันไปและทำให้มีการรับรู้ว่าเป็นสีต่างๆ โดยกระบวนการประสาทต่อไป

สัญลักษณ์ที่ใช้แทน RGB ในงานคอมพิวเตอร์กราฟริกนั้นสามารถที่จะแสดงได้โดยการใช้วงเล็บเพื่อแสดงค่าเช่น (R,G,B) หรือ [R,G,B] เป็นต้น ค่าที่แสดงในวงเล็บดังกล่าวสามารถเป็นได้ทั้ง floating point และ fixed point ในกระบวนการ render นั้น สี RGB จะถูกกำหนดชนิดของข้อมูลในการคำนวณด้วยคอมพิวเตอร์ให้เป็นแบบ floating point หรือเลขทศนิยม เนื่องจากต้องการความละเอียดสูง เช่น อาจถูกกำหนดเป็น [0.23564f,0.56546f,0.98465f] เป็นต้นโดยค่า intensity ในแต่ละ channel จะอยู่ระหว่าง 0 ถึง 1 แต่เนื่องจากในการกำหนดแหล่งกำเนิดแสงนั้น intensity ของแหล่งกำเนิดแสงสามารถที่จะมีค่าได้มากกว่า 1 ดังนั้นภายหลังจากการคำนวณค่า intensity อาจจะมีค่ามากกว่า 1 ได้ในแต่ละ channel ซึ่งค่าใน channel ใดๆที่มีค่าเกิน 1 ก็จะถูกตัด (clamp) ให้มีค่าไม่เกิน 1 แหล่งกำเนิดแสงสามารถที่จะถูกกำหนดให้เป็นค่าลบได้เช่นเดียวกัน และทำนองเดียวกันค่า intensity ที่น้อยกว่า 1 ก็จะถูกตัดให้มีค่าที่ไม่ต่ำกว่าศูนย์เช่นกัน โดยงานที่มีการจัดการเรื่องแสงเป็นอย่างดีจะไม่นิยมให้เกิดการ clamp ในลักษณะนี้เพราะอาจจะทำให้ภาพมีการไล่ tone ของแสงไม่ถูกต้องสวยงาม และไม่สามารถที่จะแก้ไขได้ในภายหลังเนื่องจากไม่สามารถทราบข้อมูลสีที่โดนตัดไป

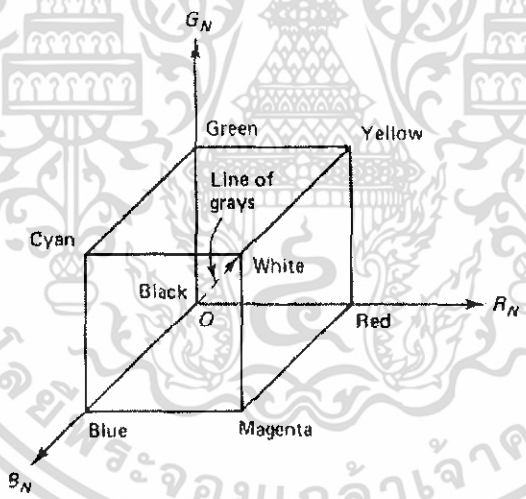
การเก็บข้อมูลสี RGB ให้อยู่ในรูป floating point นี้มีข้อดีเมื่อเราต้องการที่จะดำเนินการใดๆบนภาพที่เราสร้างขึ้น เนื่องจากมี precision ที่สูงทำให้ข้อมูลสีมีการบิดเบือนที่น้อยแต่ต้องใช้เนื้อที่ในการเก็บที่มาก เช่น 1 pixel ถ้าจัดเก็บในรูป RGBA ด้วยข้อมูลชนิด float ต้องใช้เนื้อที่เท่ากับ 4x4=16 bytes/pixel หรือ 32bits/channel ซึ่งปริมาณที่มากเกินความจำเป็นถ้าเราต้องการที่จะเก็บภาพเพื่อทำการแสดงผลเพียงอย่างเดียว เนื่องจากจอภาพทั่วไปนั้นสามารถแสดงสีที่แตกต่างกันได้เพียง 10 ถึง 12 bit เท่านั้น อีกทั้งตาของมนุษย์สามารถที่จะแยกแยะสีต่างๆได้ทั้งหมดประมาณ 10 ล้านสีและจอภาพก็ไม่สามารถที่จะแสดงสีเหล่านั้นได้ทั้งหมด ดังนั้นการใช้ข้อมูลสีที่มีขนาดเท่ากับ 8bit/channel หรือ 256 ระดับในแต่ละ channel ที่สามารถแสดงสีที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แตกต่างกันได้ทั้งหมด 16.7 ล้านสีนั้น ก็เพียงพอที่จะแสดงสีที่มนุษย์เห็นได้อย่างเป็นธรรมชาติ และใช้เนื้อที่ในการเก็บข้อมูลสีเพียง 4bytes/pixel เท่านั้น

ด้วยเหตุนี้การใช้ระบบ RGB ทั่วไปจึงกำหนดค่า intensity เป็น 8 bits ซึ่งเท่ากับ 256 ระดับในแต่ละ channel เช่น [0,0,0] หมายถึง intensity มีค่าเท่ากับ 0 ในทุก channel หรือสีดำและ [255,255,255] ซึ่งเท่ากับสีขาว ดังนั้นเมื่อพูดถึงค่าที่เป็นข้อมูลสีในแต่ละ channel ของ color model ใดๆในบางครั้งจึงอาจจะใช้คำว่าสูงสุดหรือต่ำสุดแทน เนื่องจากค่าสูงสุดอาจเป็นได้ทั้ง 1.0 เมื่อมีช่วงข้อมูลเท่ากับ 0 ถึง 1 หรืออาจจะเป็น 255 เมื่อมีช่วงข้อมูลเท่ากับ 0 ถึง 255 หรืออาจจะเป็น 100% เมื่อมีช่วงข่อมูล 0 ถึง 100%

เพื่อช่วยให้เกิดความเข้าใจใน RGB model และสามารถมองเห็นภาพได้ดียิ่งขึ้นจึงได้มีการสร้าง RGB cube ขึ้นโดยเป็น cube ที่ถูกสร้างในพิกัด 3 มิติ โดยให้แกนทั้งสามเป็นค่าของ intensity ในแต่ละ channel ตามลำดับ ดังนั้นจึงทำให้เกิด cube ที่มีมุมทั้งหมด 8 มุมที่เป็นตำแหน่งของ Red Green Blue White และ Cyan Magenta Yellow Black จำนวน 8 สีตามลำดับ โดยชุดแรก RGB เป็น primary color ของการผสมสีแบบ additive และชุดหลัง CYM คือ primary color ของการผสมสีแบบ subtractive



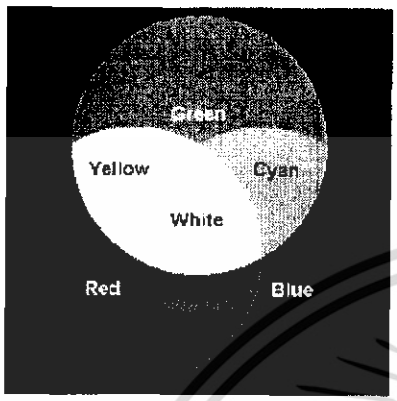
รูปที่ 2.1 RGB cube

ใน RGB cube นี้นี้มีเส้นที่ลากจุด [0,0,0] ไปยัง [255,255,255] เรียกว่า ซึ่งเป็นเส้นที่ลากจากสีดำไปจนถึงสีขาว ซึ่งสีที่อยู่บนเส้นนี้จะเป็นสีเทาที่มี intensity ที่แตกต่างกันออกไปโดยค่า intensity ของสีบน line of gray ดังกล่าวจะมีค่า intensity ที่เท่ากันทั้งสาม channel ดังแสดงในรูปที่ 2.1

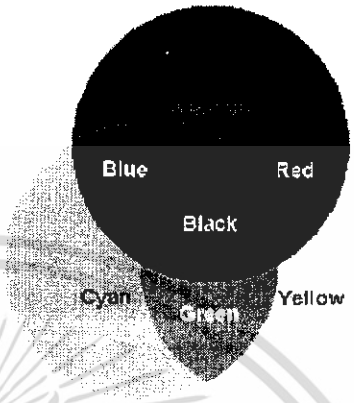
การสร้างสีทำได้ 2 วิธีคือ การผสมสีแบบ additive ได้แก่ การผสมสีโดยใช้แสงสีโดยเริ่มจากสีดำซึ่งก็คือความมืดหรือไม่มีแสง และทำการเติมหรือเพิ่มแสงสีทั้งสามสีได้แก่ RGB ลงไปเพื่อให้ได้สีที่ต้องการไปจนกระทั่งถึงสีขาว ดังแสดงในรูปที่ 2.2 ส่วนการผสมสีแบบ subtractive

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งการผสมสารสีนั้นจะเริ่มต้นจากสีขาวโดยใช้แสงสีขาวส่องลงบนกระดาษก่อนจากนั้นจึงทำการกำจัดหรือลบสีที่ไม่ต้องการออกจากแสงสีขาวนั้นโดยการใช้สารสี CYM เพื่อทำการดูดกลืนสีที่ไม่ต้องการออกไปทำให้เกิดเป็นสีต่างๆและไปสิ้นสุดที่สีดำ ดังที่แสดงในรูปที่ 2.3



รูปที่ 2.2 Additive Color



รูปที่ 2.3 Subtractive color

2.1.2 HSL Model

HSL color model เป็น color model ที่นิยมใช้กันมากโดยมีคุณสมบัติหลายอย่างที่เหมือน HSV โดยคำนึงถึงการใช้ในงานศิลปะเช่นเดียวกัน ระบบสี HSL นี้ประกอบด้วยสามส่วนคือ

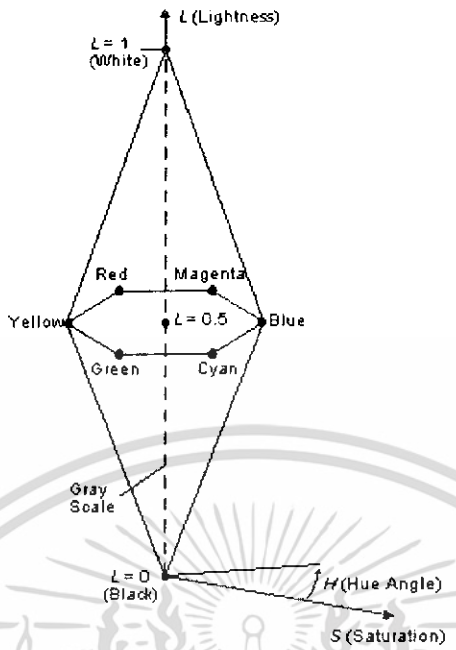
H หมายถึง hue หรือสีที่มีค่าแตกต่างกันออกไปตามความถี่ของแสงเช่น แดง น้ำเงิน ม่วง เขียว เป็นต้น

S หมายถึง saturation หรือ ความอิ่มตัวของ hue นั้นๆ เช่น สีแดง และสีชมพูก็คือสีแดงเพียงแต่สีชมพูมีความอิ่มตัวที่น้อยกว่าหรือมีสีขาวเข้ามาผสม

L หมายถึง lightness หรือ ค่าความสว่างของสีที่มีค่า hue และ saturation เท่าใดโดยที่ lightness ต่ำสุดหมายถึงสีดำ ไม่ว่าจะมี hue และ saturation เท่าใด และ lightness สูงสุดหมายถึงสีขาว ไม่ว่าจะมี hue และ saturation เท่าใด

เมื่อนำ HSL model มาสร้างเป็น model ในพิกัด 3 มิติจะทำให้ได้รูปร่างที่มีลักษณะของ hexagonal cone สองอันประกบกันโดยใช้ด้านที่เป็นฐานในการประกบและมีจุดยอดอยู่ที่ปลายด้านบน และด้านล่างตามลำดับ ปลายด้านบนที่มีค่า lightness เท่ากับ 1 ก็คือ สีขาวและปลายด้านล่างที่มีค่า lightness เท่ากับ 0 ก็คือสีดำ สีที่อิ่มตัวที่สุดจะอยู่บนระนาบที่ lightness มีค่าเท่ากับ 0.5 โดยอยู่ตรงกลางระหว่างปลายด้านทั้งสองด้าน นอกจากนั้นแล้วยังมี gray scale นั่นคือแกนกลางของ cone ทั้งสองอัน รูปร่าง double hexagonal cone ดังกล่าวคือ รูปร่างที่ transform มาจาก RGB cube ของRGB model ซึ่งมีมุมเท่ากัน 8 มุม แต่ละมุมจะมีสีที่เหมือนกับ RGB cube ได้แก่ RGBW และ CYMK ดังแสดงในรูปที่ 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 HSL Color Model

2.2 Chroma keying

เทคนิคการนำภาพ 2 ภาพมารวมกันซึ่งทำให้เกิดเป็นภาพใหม่ขึ้นมา โดยบางส่วนได้มาจากภาพแรกส่วนที่เหลือก็มาจากภาพที่สอง. A mask ในที่นี้คือส่วนหนึ่งของภาพที่ 1 มีหลายวิธีที่ทำให้ a mask ยังคงอยู่ วิธีหนึ่งที่ใช้บ่อยในการผลิตสื่อทางโทรทัศน์นั่นก็คือ Chroma Keying

ภาพ 1ภาพที่ทำโดยวิธีของ chroma keying คือจะถูกบันทึกใน studio ซึ่งจะมี background ที่เป็นสีเดียวกันโดยทั่วไปแล้วจะใช้สีน้ำเงิน นักแสดงจะต้องสวมเสื้อผ้าที่มีสีแตกต่างจากฉากพื้นหลังสำหรับรายการโทรทัศน์จะใช้อุปกรณ์พิเศษในการทำให้เกิด a mask ที่สอดคล้องกับส่วนของภาพที่เป็นสีน้ำเงินซึ่งสามารถใช้ในการรวมภาพของนักแสดงกับ background ที่มีความแตกต่างกันโดยใช้คอมพิวเตอร์เช่น รายการพยากรณ์อากาศซึ่งจะมีภาพแผนภูมิแสดงสภาพลมฟ้าอากาศมาแทนที่ background ซึ่งแผนที่นั้นถูกสร้างโดยคอมพิวเตอร์

ในบางที่จะใช้เทคนิคของ chroma key ในรูปแบบของ software เพื่อขจัดปัญหาในเรื่องของ hardware ชนิดพิเศษที่มีราคาแพงขั้นตอนที่นำเสนอนี้เป็นวิธีการที่ค่อนข้างง่าย,รวดเร็วและคุณภาพดีเมื่อเทียบกับภาพที่ได้จากการทำด้วย hardware

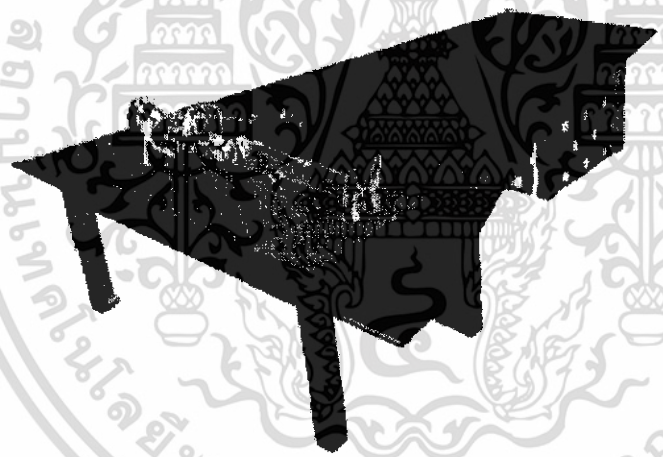
ในหัวข้อที่ 2.2.1 จะกล่าวถึงลักษณะโดยทั่งไปของ Immersive Telepresence และ chroma keying และในหัวข้อที่ 2.2.2 เป็นการบรรยายถึง algorithm ของ chroma keying ส่วนผลลัพธ์ที่ได้จะกล่าวถึงในหัวข้อที่ 2.2.3 และหัวข้อที่ 2.2.4 จะกล่าวถึงประโยชน์ของ software chroma keying

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 Virtual Environment

Projective Display System (PDS)เป็นลักษณะของศิลปะชั้นสูงของ Virtual Environment ระบบ PDS จะมีการทำงานที่หนักและส่งผลกระทบต่อ head-mounted display, ต้องมีความแน่นอนและต้องมีความรวดเร็วในการ rendering ซึ่งเหมาะที่จะใช้ hardware ลักษณะพิเศษของ projection-base VE system คือ ส่วนทั้งหมดของ real space ต่อ virtual space ทำงานร่วมกันกับ world co-ordinate system. Projection-based VR

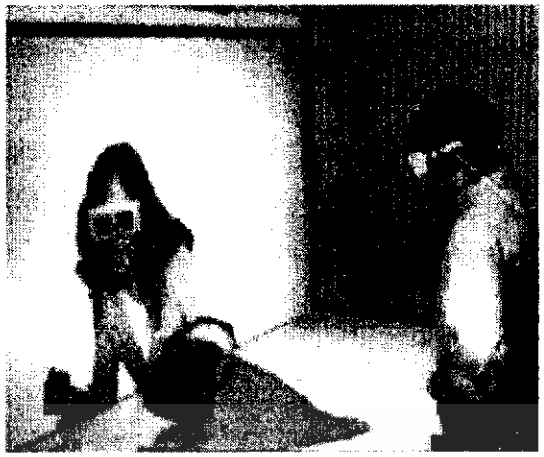
การปะติดปะต่อ virtual space ไปยัง real space ต้องใช้ projection-base virtual environments เช่น โต๊ะ หรือห้องที่ถูกสร้างขึ้น โดยทั่วไปโต๊ะหรือห้องที่ถูกสร้างขึ้น ที่เป็นมาตรฐานยอมรับโดยทั่วกันจะต้องผ่านการจดลิขสิทธิ์ Responsive Workbench(RWB คือ การจดทะเบียนการค้าของ German Nation Research center for Information Technology),Collaborative Responsive Workbench the CyberStage หรือ Teleport. RWB ถูกพัฒนาขึ้นเมื่อปี 1993 ที่ GMD และใช้โต๊ะในการทำงานซึ่งเหมาะที่จะแสดงเกี่ยวกับภาพจำลอง 3 มิติ และการก็ตอบสนองดูเป็นธรรมชาติมากกว่า แสดงในรูปที่ 2.5



รูปที่ 2.5 A model of the Responsive Workbench

และใน Collaborative Responsive Workbench นั้นได้เพิ่มเติมในส่วนของ Responsive Workbench โดยเพิ่มฉากแนวตั้งขึ้น1 อัน ดังนั้นการขยายภาพจะมีมุมมองในแบบ frustum(ส่วนที่เหลือของรูปกรวยที่ถูกตัดออกโดยขนานกับพื้น) ดังแสดงในรูปที่ 2.6 แต่ละภาพที่แสดงมีขนาด 1.80m x 1.20m และสังเกตว่า frustum อยู่รอบๆ 2m ต่อ 3m ใน cube นี้การจำลองวัตถุ 3 มิติ สามารถถูกจับต้องได้โยตรงโดยผ่านทางเข็ม,ทางข้อมูลของถุงมือหรืออุปกรณ์ input อื่นๆ Collaborative Responsive Workbench นอกจากจะมีมุมมองที่ใหญ่กว่าแล้ว ยังสมารถทำงานร่วมกันได้ในระยะไกลที่มากกว่า กล่าวคือการทำงานรอบโต๊ะนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 Collaborative Responsive Workbench

CyberStage คือ CAVE (การจดทะเบียนการค้าของ University of Illinois) มักจะมีระบบ room-size stereo display4 ด้าน ติดตั้งที่ GMD ผู้ใช้จะเห็น virtual space ขนาดใหญ่และได้ยินเสียงที่กระจายออกมา Projection systemนั้นคล้ายกับ CyberStage โดยสามารถเปลี่ยนแปลงได้ โดยตรงละคนที่เป็นศูนย์กลางจะติดต่อกับ virtual world ได้เป็นอย่างดี ขนาดของฉาก 3 ฉากที่ถูกติดตั้งขึ้นกับพื้นแต่ละอันมีขนาด 3x3 เมตร Crystal Eyes เป็นเลนส์ที่ใช้สำหรับ stereo ภาพที่แสดงมีขนาด 1024x768 pixel ที่ 10 Hz ระบบเสียง 8 ทิศทาง โดย IRCMP's room ซึ่งใช้ software Spatilisateur ในการบันทึกเสียง และรองรับเสียงต้นฉบับที่จำกัดเขตภายใน Virtual Environment วิธีการใหม่ๆของ CyberStage คือ acoustic floor ซึ่งรองรับการสร้างฉากที่สั่นสะเทือน

วิธีการแรกในการ Immersive Telepresence คือการ mask โดยใช้ Mask ของ Collaborative Responsive Workbench และ CyberStage ในการจัดทำ virtual space. Immersive Telepresence อาศัยหลักการทำงานของ stereo-video, hardware chromakey และองค์ประกอบการ mappingในการสั่งให้ videoรวมกับ virtual environment

Chroma-keying คือเทคนิคที่ใช้ร่วมกับ virtual studioผู้แสดงจะอยู่ในพื้นที่ของห้องที่มีสีน้ำเงิน chroma-keyer ใช้แยกภาพนักแสดงออกมาจาก background สีน้ำเงิน ทำโดยเพิ่ม alpha-channel ไปใน frame video ต้นฉบับ. นำ video มารวมกับ virtual environment ในเวลา real-time. Camera parameter คือ track และ virtual environment คือการ render ในเวลา real-time ซึ่งขึ้นอยู่กับเทคนิคการเขียนภาพของ real camera ถึงแม้ว่าอุปกรณ์ของ Chroma keying จะมีคุณภาพสูงสำหรับ virtual studio แต่มันยังมีราคาแพง จึงมีการเสนอวิธีการใช้ software chroma keying

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 Chroma Keying Exact Solutions

การเข้าถึงปัญหาเกี่ยวกับคณิตศาสตร์ของ chroma keying ถูกเสนอโดย Smith และ Blinn พวกเขาให้คำจัดกัดความและเรียกมันว่า " Matting Problem" ซึ่งต่อจากนั้นเขาได้อธิบายถึงการใช้วิธีการใหม่

ให้ C_f และ C_b อยู่ที่จุด corresponding และ C_k คือสีพื้นหลัง สมมุติว่า $C_f = C_o + (1-\alpha_0) C_k$ และให้ C_o เป็นส่วนประกอบของสี $C = C_o + (1-\alpha_0) C_b$ ที่จุด *corresponding, สำหรับทุกจุดนั้น* C_f และ C_b ใช้ร่วมกัน *Note* C_i คือ vectors $C_i = [R_i \; G_i \; B_i \; \alpha_i]$ ค่าของ α_i อยู่ในช่วง $[0 \ldots \alpha_i]$ โดยที่ $0 \leq \alpha_i \leq 1$

C_i คือสีวัตถุที่แสดงใน foreground ซึ่งรวมด้วยส่วนประกอบ alpha. นี่คือค่าที่ถูกกำหนดให้สีbacking C_k และสีที่แท้จริงของภาพที่จุดนั้นๆ, C_f จะยอมให้วัตถุรวมกับ background ที่กำหนด(แทนโดย C_b) การแก้ปัญหา matting problem เริ่มต้นโดยตัวประกอบภาพ(ซึ่งถูกบันทึกโดยกล้อง)และรู้ถึงสี backing (เช่นสีน้ำเงิน) ผลของระบบคือ

$$R_f = R_o + (1-\alpha_0) R_k$$
$$G_f = G_o + (1-\alpha_0) G_k$$
$$B_f = B_o + (1-\alpha_0) B_k$$

X_f เป็นตัวประกอบที่แทนด้วยค่าที่ถูกบันทึกโดยกล้อง และ X_k แทน background X_o จะมาจากการแก้ปัญหาของระบบนี้ *Note* บันทึกว่ามีสี่ค่าที่ไม่รู้, และเพียงสามสมการเท่านั้นที่ส่งผลภายใต้การเจาะจงระบบดังนั้นมันเป็นไปไม่ได้ที่จะรับการแก้ปัญหาในเรื่องปัญหา chroma keying เมื่อการใช้การเข้าถึงนี้โดยการบังคับการจัดวางบนระบบกลายเป็นการแก้ปัญหาของมัน เช่น ต้องการวัตถุ foreground ที่ไม่ใช่สีน้ำเงินเป็นส่วนประกอบ ผลคือ

$$C_o = \left[\; R_f \quad G_f \quad 0 \quad 1 - \frac{B_f}{B_k} \; \right]$$

โดยมีตัวเลือกในการจำกัดเป็นเฉดสีเทาคือ

$$C_o = \left[\; R_f \quad G_f \quad B_f - B_k + \alpha_o B_k \quad \frac{G_f - (B_f - B_k)}{B_k} \; \right]$$

Smith และ Blinn แสดงให้ดูเพียงอย่างเดียว การแก้ปัญหาทั่วไปคงอยู่ แต่วิธีของพวกเขาต้องการถ่ายวัตถุที่แตกต่างกับ background ซึ่งวิธีการนี้ไม่สามารถใช้การได้เมื่อมีนักแสดงเข้ามาร่วมด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 Chroma Keying: Approximation

สิ่งสำคัญในการทำงานบน chroma keying คือมันมีการป้องกันโดยการจดสิทธิบัตรส่วนมามันจะหมดอายุภายในเวลาไม่กี่ปีสิ่งที่กล่าวนี้ถูกค้นพบโดย Petro Vlahos ผู้ซึ่งรู้ดีเกี่ยวกับ Ultimatte™ ของเขา. วิธีการประมาณค่าต้องใช้มนุษย์ในการแก้ปัญหาทางคณิตศาสตร์(หรืออาจจะใช้ software intelligent) ในการปรับค่าตัวแปรบางตัวจนกระทั่งได้ผลลัพธ์ที่ถูกต้อง ส่วนหนึ่งที่ยากที่สุดในการประมาณค่าคือ

$$\alpha_o = 1 - a_1(B_f - a_2 G_f).$$

สำหรับการประมาณค่าของการทำงานนี้มีข้อสมมุติฐานหลายอย่างที่ต้องทำ เช่น การใช้ background สีน้ำเงิน,ผู้แสดงจะต้องไม่สวมเสื้อผ้าสีน้ำเงิน เป็นต้น ค่าที่ใกล้เคียง chroma key ลำดับที่สองอยู่ที่ความพอใจมากกว่า สมมุติฐานแรกก็คือ ช่วงของ background color c_k นั้น จะอยู่ในพื้นที่ขอบเขตของ 3D-RGB และเป็นส่วนที่ขาดหายไปจากสีที่พบในส่วนของ foreground color object รอบๆ ขอบเขตในพื้นที่ RGB จะมีรูปหลายเหลี่ยมมากมายถูกสร้างขึ้นมาซ้อนทับกันอันที่เล็กๆอันนึงจะประกอบกันเป็นสีของ background ที่จุดรวมภายในจะถูกกำหนดให้มีค่าอัลฟ่าของศูนย์ รูปหลายเหลี่ยมที่ใหญ่ที่สุดต้องอยู่บนหรืออยู่นอกเหนือพื้นที่ของสีที่พบใน foreground object นั่นคืออัลฟ่าเท่ากับหนึ่งสำหรับที่ทุกจุดที่อยู่นอกเหนือรูปหลายอันนี้ ในส่วนที่อยู่ระหว่างรูปหลายเหลี่ยมสองอันจะเกิดสิ่งที่แตกต่างกันขึ้น แต่ละอันจะถูกแทนที่ด้วยค่าคงที่อัลฟ่าของพื้นที่ผิวหน้าอยู่ระหว่าง0-1

อุปกรณ์ primatteTM ของบริษัท Photron Ltd จะขึ้นอยู่กับค่าพวกนี้โดยใช้128-faced polyhedra นี่เป็นส่วนของความรู้เบื้องต้นที่พูดถึงการทำงานของระบบ Smith และ Bill ได้พูดถึงปัญหาที่ยังคงเหลืออยู่สำหรับการประมาณค่าการเข้าใกล้นี้

สำหรับ algorithm ที่มีอยู่ในปัจจุบันนั้นต่อมาได้ถูกเก็บไว้ในลำดับของการคาดคะเนความต้องการของเราจะอยู่ระหว่างการปรับตั้งค่า จนกว่าจะได้ผลลัพธ์ที่พอใจ

2.2.4 Algorithm

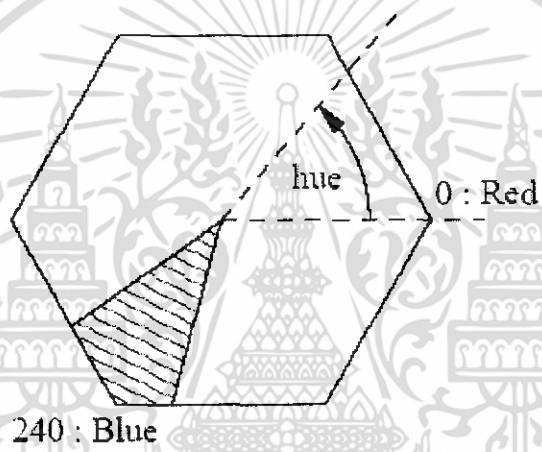
สิ่งที่สำคัญที่สุดของขั้นตอน chroma keying algorithm คือการแยก pixel ว่าเป็น key color หรือไม่ สำหรับการเปรียบเทียบ hue base ที่เกิดขึ้นนั้นมีการกล่าวถึงตั้งแต่แรกตามหัวข้อการปรับปรุง ส่วนที่แตกต่างระหว่าง 2 algorithm นี้ขึ้นอยู่กับการแบ่งแยกสีที่ทำสำเร็จ

สำหรับเรื่องนี้ key color นี้ถูกสมมุติให้เป็นสีน้ำเงินว่าเป็นสีที่นิยมใช้เป็น key color ซึ่งใกล้เคียงกับสีผิวมนุษย์ จะพบได้ว่าผู้คนทั้งหลายชอบที่จะทำงานในห้องสีน้ำเงินซึ่งต่างจากสีแดงกับสีเขียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.5 Hue-base approaches

คอมพิวเตอร์ส่วนใหญ่จะแสดงผลโดยใช้ RGB นี่คือสิ่งที่สามารถใช้แสดงแทนรูปได้ ที่การเปลี่ยนแปลงง่ายๆของ RGB สามารถที่จะแปลงไปจนใช้แทนที่ HSL ซึ่งประกอบไปด้วย สี, แสง, และ มีน้ำมีนวล ส่วนที่เป็นรูปทรงเลขาคณิตของ พื้นที่สี HSL นั้นคือ double-hexcone ถ้ามอง hexcone จากด้านบนแล้ว เราจะเห็น hexcone ที่มีสีเปลี่ยนแปลงไปตามมุมของเส้น วัดจากสีแดงที่มุม 0 องศา ในส่วนที่จัดไว้ สีน้ำเงิน จะอยู่ที่ 240 องศา ด้วยเหตุนี้ การกำหนดจุดสีน้ำเงินเราสามารถที่จะกำหนดคร่าวๆระหว่าง 230-255 องศา ดังในรูปที่ 2.7 เช่น รูป 6 เหลี่ยม HSL ส่วนที่เป็นเงานั้นจะถูกระบาย ส่วนที่เป็นสีนั้นจะเปลี่ยนไปตามมุม และควรที่จะถูกเลือกโดย ผู้ใช้ นำมาเทียบกับรายละเอียดของ ส่วนที่เป็นฉากสีน้ำเงิน



รูปที่ 2.7 HSL hexagon

ในระหว่างที่ใกล้จะเรียบร้อยดีนั้น จะเกิดอุปสรรคเมื่อใช้ software รูปภาพนั้นจะอยู่ในส่วนของ HSL ถ้าไม่จะต้องแปลงมันก่อนโดยใช้ slow down อัลกอริทึม

สำหรับการเปรียบเทียบการใช้ software HSL นั้นได้มีอยู่แล้ว อัลกอริทึมจะ มีประโยชน์มาเมื่อใช้สีน้ำเงินเท่านั้น นั่นหมายความว่า HSL เต็มขั้นนั้นห้ามทำ หัวใจของ อัลกอริทึมนั้นดูได้จาก รูปที่ 2.8 จำไว้ว่าการ implemention มีแค่ สอง ส่วนเท่านั้น คือ ส่วนที่โปร่งแสงและส่วนที่ทึบแสง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// s points to the interleaved RGBA image
unsigned char *rp=s,*gp=s+1,*bp=s+2,*ap=s+3;
int h,l,s;
for (int idx = 0; idx < imgsize; idx++)
{
  int max = (*rp > *bp) ? *rp : *bp;
  max = max > *gp ? max : *gp;
  int min = (*rp < *bp) ? *rp : *bp;
  min = min < *gp ? min : *gp;

  if (max == min) h = UNDEFINED; else
  {
    int delta = max - min;
    if (*bp == max) h = 240 +
         (60*(*rp-*gp))/delta;
    else h = 0;
    if (h < 0) h += 360;
  }

  if ((h < hue_max) && (h > hue_min))
   *ap = 0;
  else *ap = 255;
  rp += 4; bp += 4;
  gp += 4; ap += 4;
}
```

รูปที่ 2.8 C++ code for the HSL algorithm

2.2.6 An improved approach

ถ้ารามีภาพที่อยู่ในพื้นที่ *RGB* มันเป็นไปได้ที่จะเพิ่มความเร็วของอัลกอริทึมสูงสุดถึง *5 operation ต่อ pixel* และต้องพิจารณาให้ได้ว่าให้ได้ว่า pixel ไหนมีสีฟ้า อย่างแรกที่เราต้อง set ให้เป็นมาตรฐานคือ B>R และ B>G ซึ่ง RGB เป็นส่วนประกอบของภาพที่สามารถแยกออกมาได้ง่ายในระยะ [0...255] กล่าวอีกอย่างหนึ่ง, ถ้าจุดภาพ pixel จะดูเหมือนสีน้ำเงิน, ส่วนประกอบสีที่เป็นจุดเด่นต้องเป็นน้ำเงิน อย่างไรก็ตาม มันยังคงกว้างเกินไป เพราะเราสามารถมี B = x, R = x-1, G = x-1 ซึ่งเป็นเฉดของสีเทาจะถูกแยกผิดเป็นสีน้ำเงินแทน มีทางนึงโดยจำกัดการตั้งค่าให้แคบลงโดยเพิ่มการบังคับระยะ เช่น ถ้า

$$d = \sqrt{(B-R)^2 + (B-G)^2} > d_{max} \qquad (1)$$

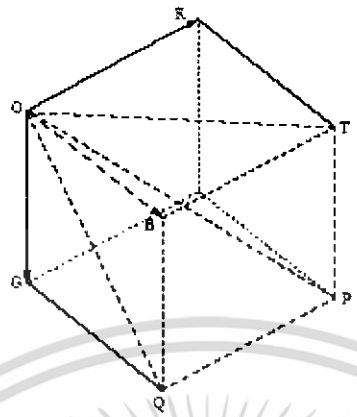
ต่อมาเราสามารถพูดสิ่งนั่นจุดภาพพิกเซลจะปรากฏสีน้ำเงินอย่างอย่างเห็นได้ชัด, และควรจะ key ออก การคูณและการถอดสแควร์รูตจะเสียเวลามาก, ดังนั้นทำให้ง่ายขึ้นโดยใช้สมการที่ (2)มาแทน

$$d = 2 \times B - R - G \qquad (2)$$

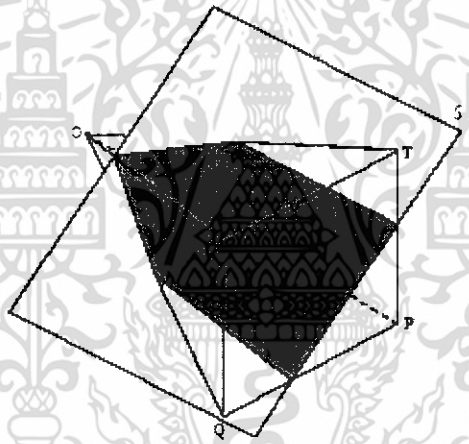
ในรูปที่ 2.9 นี้จะแสดง RGB cube ซึ่งภายในมีพีระมิดที่ลาดเอียง OBTPQ (เส้นที่อยู่ข้างใน) พีระมิดนี้จะจำกัดค่าของ cube ซึ่งคือ $B \geq R$ และ $B \geq G$ ในรูปที่ 2.10 จะแสดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พีระมิด OBTPQ ที่มีระนาบ S ตัดผ่านโดยระนาบ S นี้จะขนานกับเส้นทแยงมุมหลัก OP และแสดงพื้นผิวของค่าคงที่ d (ดังที่กำหนดไว้ในสมการที่ (2)) *Note* สมการนี้ถูกจำกัดเพียงแค่ภาพในพีระมิด(ส่วนของสีน้ำเงินเท่านั้น)



รูปที่ 2.9 An RGB cube showing the skew pyramid OBTPQ

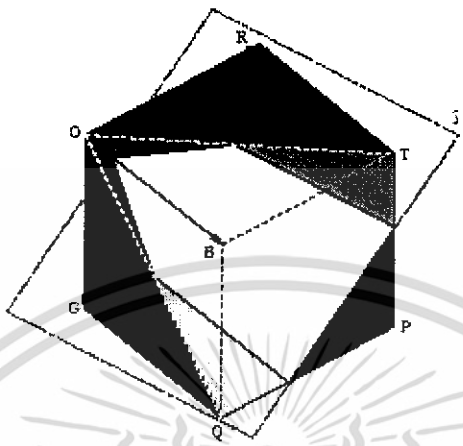


รูปที่ 2.10 An RGB cube showing a plane S intersecting the skew pyramid OBTPQ

Dark-gray polygon คือ พื้นผิวหน้าตัดของพีระมิดและระนาบ S จุดทั้งหมดภายใน polygon นี้มีลักษณะเช่นเดียวกับสีในพื้นที่ RGB space กับ ค่าคงที่ *d* (ค่า *d* ที่กำหนดในสมการที่(2)) เส้นทแยงมุม OP คือ "gray line" ของ cube , สีทั้งหมดบนเส้นนี้มีค่าเท่ากับ R,G และส่วนประกอบ B ระนาบ S จะขนานกับเส้นทแยงมุม OP ,ความจริงที่ได้มาจากสมการที่(2) สิ่งนี้ชี้แนะให้เห็นว่าจุดทั้งหมดใน dark-gray polygon มีค่าเท่ากับสีน้ำเงิน การประมาณค่าเท่ากับความเข้มของค่า gray และเท่ากับค่าความเข้ม ดังนั้นค่า *d* ของสีในพีระมิดสามารถแยกลำดับชั้นเริ่มต้นจากสีน้ำเงินหรือไม่ก็เลือกโดยจุดเริ่มต้นในค่า *d Note* ถึงแม้ค่า *d* จะมีระยะไม่ถูกต้องตามเมตริก, แต่ค่อนข้างจะคำนวณการประมาณมีประสิทธิภาพทำให้มองเห็นระดับที่เหนือกว่าระนาบ S ก่อนที่จะเคลื่อไปยังรูปที่ 2.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 2.11 จะแสดงถึง RGB cube อีกครั้ง ใน cube นี้พีระมิดยังคงมองเห็นเส้นทแยงมุม ทำให้มองเห็นรูปภาพที่ระดับของพีระมิดเหนือกว่าระนาบ S ที่ถูกย้าย สามเหลี่ยมสองด้านที่มีสีเทาเป็นด้านข้างของกำแพงพีระมิด(ที่มองเห็นเหนือระนาบ S)



รูปที่ 2.11 An RGB cube showing a plane S of constant d

ที่กล่าวมาก่อนหน้านี้, สี(pixel) สามารถแบ่งแยกประเภทเหนือกว่าระนาบ S (และในพีระมิด)หรือต่ำกว่าระนาบ S ที่ซึ่งตำแหน่งที่แน่นอนของระนาบ S คือตั้งค่าโดยเลือกจากค่า d_{max} ที่เป็นตัวแปรเริ่มต้น สำหรับแต่ละ pixel ค่าสำหรับ d จะถูกคำนวณ,และ mask ที่สร้างจะสอดคล้องกับผลลัพธ์ของการเปรียบเทียบของ d ถึง d_{max}

ในทางปฏิบัติ mask ที่เราสร้างนั้นจะไม่มี binary mask, แต่ค่าอัลฟ่าจะมีค่า 8-bit เมื่อเมือนำมารวมกับ foreground และ background (ใช้ mask) จุดภาพพิกเซลที่ส่งผลจะแก้ไขข้อความโดยการสอดแทรกเชิงเส้นของฉากหน้า และจุดภาพพิกเซลพื้นหลัง กับค่าอัลฟ่าจะทำการผสมตัวแปร ในพริบตา, ค่าอัลฟ่ากระทำเป็นค่าลักษณะโปร่งแสงสำหรับแต่ละจุดภาพพิกเซล

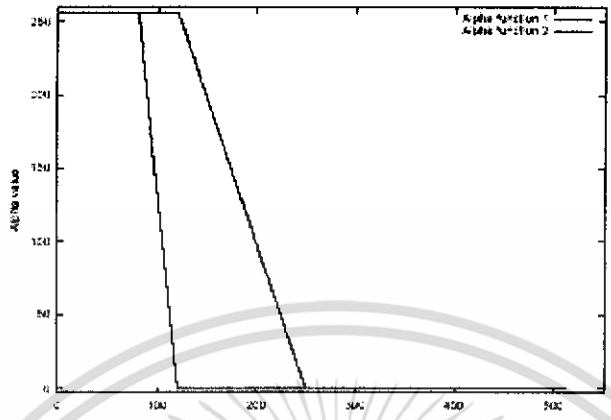
วิธีง่ายๆของการวัดระยะ d โดยสร้างขอบขึ้นรอบๆตัวนักแสดง เพื่อพยายามในการแยกออกจาก background ตั้งแต่ที่เรามีค่า transparency 8 bit มันอาจจะทำให้ขอบราบเรียบขึ้นโดยใช้ semi-transparent pixel บนขอบ ได้มีการทดลองอัลกอริทึมในการแก้ปัญหาในทางที่ดีขึ้น ระยะของ d คำนวณโดยใช้สมการที่ (2) ถูกใช้เป็น input เรียก alpha function ตัว alpha function จะ return ค่าระหว่าง 0 (transparent) และ 255 (opaque) โดยอาศัยวิธีจุด pixel จาก OP ใน RGB space สมการที่ (2)จะ return ค่าระหว่าง 0 (สำหรับสีเทาบริสุทธิ์) ถึง 510 (สำหรับสีน้ำเงินบริสุทธิ์) ซึ่ง RG และ B มีค่าอยู่ระหว่าง 0 และ 255 ฟังก์ชั่นอัลฟ่าทั้งสองแสดงในรูปที่ 2.12

เหตุผลที่แสดงฟังก์ชั่นอัลฟ่าทั้งสองคือ อัลกอริทึมที่มีการปรับตัวแปรควบคุมรูปร่าง ของฟังก์ชั่นอัลฟ่า ซึ่งใช้อัลกอริทึมสำหรับรายละเอียดของห้องสีน้ำเงินในวิดีโอ ใน code นี้อัลฟ่าฟังก์ชั่นจะปฏิบัติกับ 512-entry ในทางปฏิบัติทางลาดเชิงเส้นที่แสดงในรูปที่ 8 มีการทำงานที่ดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

มาก การผลิตขอบรอบๆนักแสดงราบรื่น ซึ่งที่ตำแหน่งเริ่มต้นและตำแหน่งสิ้นสุดของทางลาดจะเลือกสีน้ำเงินออก



รูปที่ 2.12 Two sample alpha functions

ในรูปที่ 2.13 .ในแต่ละรอบ จะทำการเปรียบเทียบเพียงสองครั้ง ก็จะทราบว่าเป็นจุดที่ต้องการหรือไม่ ในการทำ lookup นั้นต้องการ สามโอเปอร์เรชั่นในการคำนวณหา index ที่ถูกต้องในตาราง ถ้าเป็นจุดที่ไม่ใช่สีน้ำเงินจะใช้เพียงสองโอเปอร์เรชั่น ในขณะที่จุดสีน้ำเงินจะต้องใช้ถึงห้า แล้วแต่ภาพ ผลลัพท์จะอยู่ระหว่างสามและสี่โอเปอร์เรชั่นต่อพิกเซลล์โดยเฉลี่ย

```
// s points to the interleaved RGBA image
unsigned char *rp=s,*gp=s+1,
              *bp=s+2,*ap=s+3;
for (int i = 0; i < imgsize; i++)
{
  if (*bp > *rp && *bp > *gp )
    *ap = alpha_map[(*bp<<1) - *rp - *gp];
  else *ap = 255;
  rp += 4; bp += 4;
  gp += 4; ap += 4;
}
```

รูปที่ 2.13 Two sample alpha functions

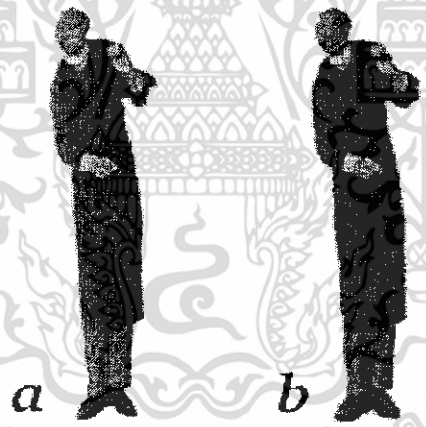
2.2.7 Image quality

มีหลายปัจจัยที่สามารถพิจารณาในการเปรียบเทียบคุณภาพของภาพที่ได้จากอัลกอริทึมของ software กับอุปกรณ์ hardware ในส่วนของ hardware นั้นสามารถเหลือเงาเอาไว้ได้ในขณะที่ถ้าเป็น software มันจะเอาเงาออก แต่ในการตั้งค่าในส่วนที่ 5 นั้นต้องการการย้ายออกของเงา เพื่อการเปรียบเทียบที่ดีคือ diff- cult ในการ test ภาพจะต้องเอาผลลัพธ์มาเปรียบเทียบกันและให้เวลา 2 นาทีในการ fine-tuning

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
62445

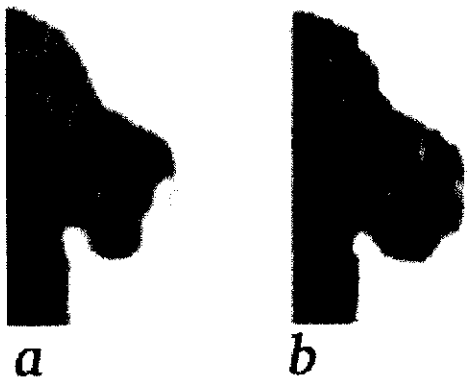
รูปที่ 2.14 Original Image

การสังเกตดูน่าสนใจขึ้น ดูรูปที่ 2.15 โดยดูที่เหลืออยู่บนแขนของศัลยแพทย์(ที่ด้านมือขวาของรูปภาพ) ส่วนหนึ่งที่ hardware ได้key อย่างไม่ถูกต้องโดยเอาบางส่วนของแขนออกไป (รูปที่ 2.15a)และ software keyer (รูปที่ 2.15b) สามารถจัดการส่วนนี้ได้ดีกว่าเล็กน้อย ได้ขยายให้ดูในภาพที่ 2.16 ซึ่ง software เองก็ผลิตรูปภาพได้ดีพอที่จะใช้กับโปรแกรมได้



รูปที่ 2.15 a) Hardware keyed, b) Software keyed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 Blow-up of elbow region: a) Hardware keyed, b) Software keyed

2.2.8 Performance

สำหรับส่วนนี้, วิธีคิด HSL เหมาะสมคือถูก bench markedรวมทั้งปรับปรุงอัลกอริทึม การทดสอบประมวลผลบน Intel Pentium II, ที่ความเร็ว450MHz ในการทดสอบทั้งสองจะต้องใช้อัลกอริทึมในหน่วยความจำหลัก วิธีคิดใช้กับรูปภาพในหน่วยความจำหลัก, และเฟรมต่อที่สอง(FPS) การแบ่งแยกตามชั้นถูกคำนวณอาศัยเวลาที่มันหยิบให้กุญแจสิ่งหนึ่งพันรูปภาพนี้เวลาและ Frames Per Second (FPS)

Resolution	HSL Algorithm	Improve Algorithm
720 ×576	12.02 FPS	43.16 FPS
368 × 288	50.45 FPS	173.01 FPS
128 × 256	158.98 FPS	625.00 FPS

ตารางที่ 2.1 Software HSL vs Improved Algorithm, FPS

Resolution	HSL Algorithm	Improve Algorithm
720 ×576	4.98 Mpixels/s	17.90 Mpixels/s
368 × 288	5.15 Mpixels/s	18.34 Mpixels/s
128 × 256	5.21 Mpixels/s	20.48 Mpixels/s

ตารางที่ 2.2 Software HSL vs Improved Algorithm, Mp/s

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในตารางที่ 2.1 จะแสดงผลลัพธ์ของการรันอัลกอริทึมที่ภาพขนาดต่างกันขณะที่ตารางที่ 2.2 ก็แสดงเหมือนกัน แต่แสดงในอัตราหนึ่งล้านของ pixel ต่อวินาที (Mpixel/s) การปรับปรุงอัลกอริทึมที่แสดงนั้นมีความสามารถเพิ่มขึ้น 340% และสูงถึง 390% ในการปฏิบัติเมื่อเปรียบเทียบกับวิธีคิด HSL

2.3 VideoOCX

VideoOCX คือ ActiveX control ที่ทำให้นักเขียนโปรแกรมสามารถทำ video capture ได้ง่ายขึ้นและทำการประมวลผลรูปภาพเข้าไปในโปรแกรมได้ ตัว control ที่เข้ากันได้ส่วนมากเป็นอุปกรณ์ Video-for-Window (VFW) เช่นเดียวกับกล้อง USB (Webcams) และ framgrabers ที่ทำงานร่วมกับ CCD VideoOCX จะยอมให้จับภาพวิดีโอในหนึ่งขั้นตอนที่ง่ายดาย โดยต้องนำ VideoOCX เข้าไปวางในโปรแกรม เปลี่ยนคุณสมบัติและเพิ่มคำสั่งเล็กน้อย ก็สามารถใช้งานได้

2.3.1 ความต้องการพื้นฐานสำหรับใช้งาน VideoOCX

- Windows 95/98/ME/NT/2000/XP (TM)
- Development environments that supports ActiveX (TM) technology (e.g. Visual Basic (TM), Visual C++ (TM), Borland C++ (TM), Delphi (TM)...)
- Video for Windows compatible image source (USB camera, framegrabber)
- Frame rate and processing speed depend on your system performance

2.3.2 ความสามารถของ VideoOCX

- ฟังชันก์ใช้งานง่าย code เพียงแค่ 3 บรรทัดก็สามารถใช้สำหรับเครื่องดูวิดีโอได้
- Capture color video frames เข้าไปหน่วยความจำด้วยความเร็วสูง (เช่น 25fps ที่ 768x576)
- เก็บ frame capture เป็น AVI sequence ในรูปแบบอื่นๆได้
- บันทึกภาพที่ capture ลงใน disk
- มี option แปลงภาพสีเป็นภาพขาวดำ
- เข้าถึงข้อมูลของ pixel RGB ได้
- รองรับอุปกรณ์วิดีโอcapture PAL/NTSC VIW ได้มากกว่า
- รองรับกล้อง USB
- รองรับ RGB AVI sequence
- แปลงภาพต้นฉบับ (AVI, Video) โดยการปรับ control เพียงตัวเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3 Method ต่างๆสำหรับการใช้งาน VideoOCX

- **Init Method**

Syntax : BOOL object.Init()

เป็นขั้นตอนแรกที่ VideoOCX จะเชื่อมต่อไปยังแหล่งข้อมูลภาพที่ระบุไว้ จะถูกเรียกเพียงครั้งเดียวจนกระทั่งมีการเปลี่ยนโหมดของแหล่งข้อมูลภาพ ในโหมด AVI (Mode = 1) จะแสดง DialogBox สำหรับเลือกไฟล์ขึ้นมาให้เลือกไฟล์วิดีโอฟอร์แม็ต AVI ที่จะใช้เป็นแหล่งข้อมูลภาพ ส่วนในโหมด VFW (Mode = 0) จะเป็นการเชื่อมต่อไปยังแหล่งข้อมูลภาพจากกล้อง

เมธอดนี้จะส่งค่ากลับเพื่อบอกว่าทำการเชื่อมต่อไปยังแหล่งข้อมูลภาพสำเร็จหรือไม่

- **InitFromFile Method**

Syntax : BOOL object.InitFromFile(LPCTSTR filename)

เป็นขั้นตอนแรกที่ VideoOCX จะเชื่อมต่อไปยังแหล่งข้อมูลภาพใน AVI โหมด ได้แก่ ไฟล์วิดีอฟอร์แม็ต AVI โดนจะระบุ Path ของไฟล์ผ่าน ตัวแปร filename

เมธอดนี้จะส่งค่ากลับเพื่อบอกว่าทำการเชื่อมต่อไปยังแหล่งข้อมูลภาพสำเร็จหรือไม่

- **Close Method**

Syntax : BOOL object.Close()

เป็นขั้นตอนสุดท้ายเมื่อเลิกใช้แหล่งข้อมูลภาพ

เมธอดนี้จะส่งค่ากลับเพื่อบอกว่าทำการ Close สำเร็จมั้ย

- **Start Method**

Syntax : BOOL object.Start()

เป็นขั้นตอนแรกก่อนทำการดึงภาพจากแหล่งข้อมูลภาพ

เมธอดนี้จะส่งค่ากลับเพื่อบอกว่าทำการ Start สำเร็จมั้ย

- **Stop Method**

Syntax : BOOL object.Stop()

เป็นขั้นตอนหลังเลิกทำการดึงภาพจากแหล่งข้อมูลภาพ

เมธอดนี้จะส่งค่ากลับเพื่อบอกว่าทำการ Stop สำเร็จมั้ย

- **GetColorImageHandle Method**

Syntax : long object. GetColorImageHandle()

ใช้สำหรับจองหน่วยความจำสำหรับเก็บภาพเพื่อนำไปใช้ในการประมวลผลต่างๆ

เมธอดนี้จะส่งกลับค่าที่ใช้อ้างอิงถึงหน่วยความจำส่วนนี้ เพื่อใช้ส่งต่อให้ Method อื่นๆ ใช้ในการประมวนผลต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **ReleaseImageHandle Method**

Syntax : void object .ReleaseImageHandle(long imagehandle)

ใช้สำหรับคืนหน่วยความจำที่เก็บภาพซึ่งอ้างอิงโดยตัวแปร imagehandle

- **CtlCapture Method**

Syntax : long object .CtlCapture(long imagehandle)

ใช้สำหรับจับภาพที่เฟรมปัจจุบันมาไว้ในหน่วยความจำที่อ้างอิงโดยตัวแปร imagehandle และเตรียมพร้อมสำหรับเฟรมถัดไป

เมธอดนี้จะส่งกลับค่าเป็นตำแหน่ง address เริ่มต้นของข้อมูลภาพในหน่วยความจำ ถ้าทำการจับภาพไม่สำเร็จจะส่งกลับค่าเป็น 0

- **CtlShow Method**

Syntax : void object .CtlShow(long imagehandle)

ใช้เมื่อต้องการนำภาพในหน่วยความจำที่อ้างอิงโดยตัวแปร imagehandle ขึ้นมาแสดงบน VideoOCX Control

- **SetPreview Method**

Syntax : BOOL object .SetPreview(BOOL show)

กำหนดการแสดงภาพตัวอย่างใน ViderOCX Control

- **GetImageWidth Method**

Syntax : long object .GetImageWidth(long imagehandle)

ใช้ดึงค่าความกว้างของภาพในหน่วยความจำที่อ้างอิงโดยตัวแปร imagehandle

- **GetImageHeight Method**

Syntax : long object .GetImageHeight(long imagehandle)

ใช้ดึงค่าความกว้างของภาพในหน่วยความจำที่อ้างอิงโดยตัวแปร imagehandle

- **GetImageDataSize Method**

Syntax : long object .GetImageDataSize(long imagehandle)

ใช้ดึงขนาดของหน่วยความจำของภาพในหน่วยความจำที่อ้างอิงโดยตัวแปร imagehandle

- **GetDataPointer Method**

Syntax : long object.GetDataPointer(long imagehandle)

ใช้ดึงตำแหน่ง address เริ่มต้นของข้อมูลภาพในหน่วยความจำของภาพในหน่วยความจำที่อ้างอิงโดยตัวแปร imagehandle

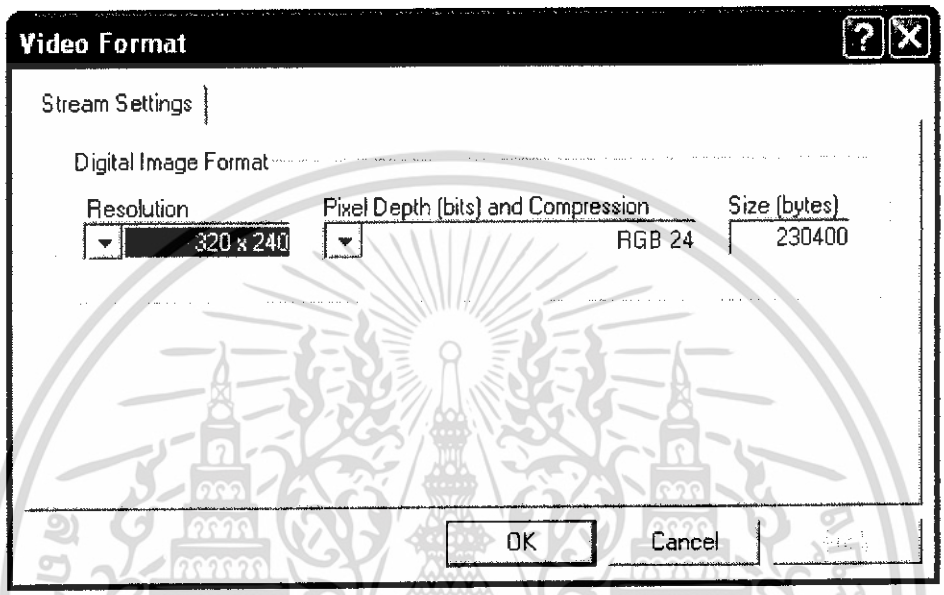
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **ShowFormatDlg Method**

Syntax : BOOL object.ShowFormatDlg()

ใช้เมื่อต้องการแสดง DialogBox เพื่อปรับค่า ขนาดของภาพใช้ได้กับโหมด VFW เท่านั้น และจะต้องทำการ เรียกใช้เมธอด Init ก่อนและไม่อยู่ในช่วงที่กำลังทำหารจับภาพ

เมธอดนี้จะส่งกลับค่า เป็น TRUE ถ้าสามารถทำการแสดง Format Dialog ได้สำเร็จ



รูปที่ 2.17 แสดงหน้าต่าง Video Format

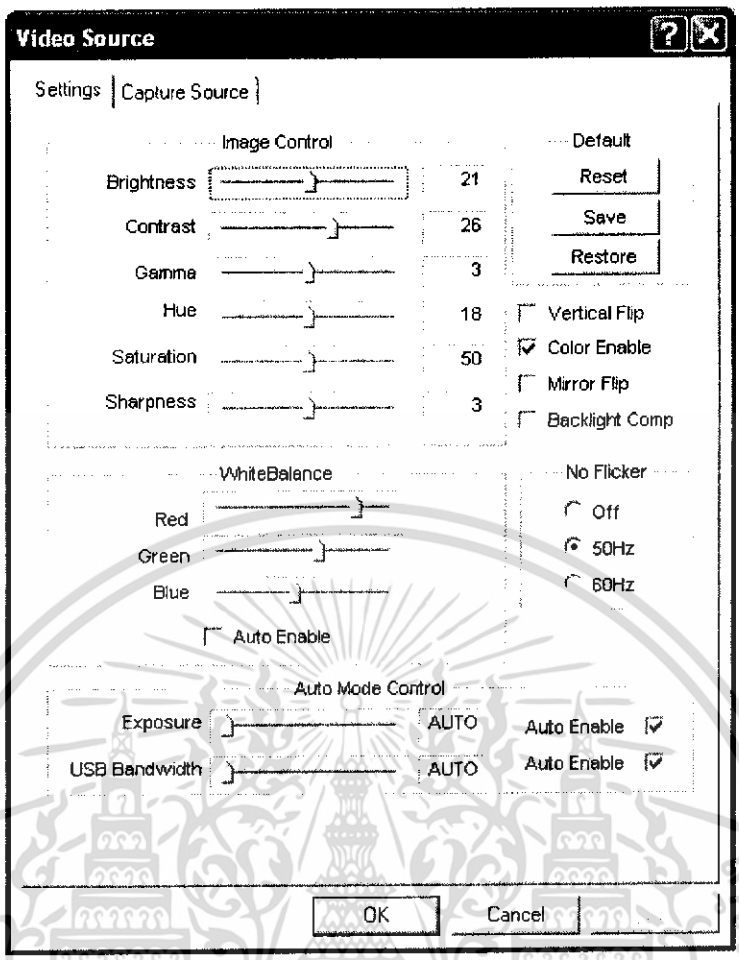
- **ShowSourceDlg Method**

Syntax : BOOL object.ShowSourceDlg()

ใช้เมื่อต้องการแสดง DialogBox เพื่อปรับค่าต่างๆเกี่ยวกับรูปภาพ ได้แก่ Brightness , Contast , Gamma , Hue , Saturation , Sharpness และค่าอื่นๆ ตามความสามารถของกล้อง

เมธอดนี้จะส่งกลับค่า เป็น TRUE ถ้าสามารถทำการแสดง Format Dialog ได้สำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



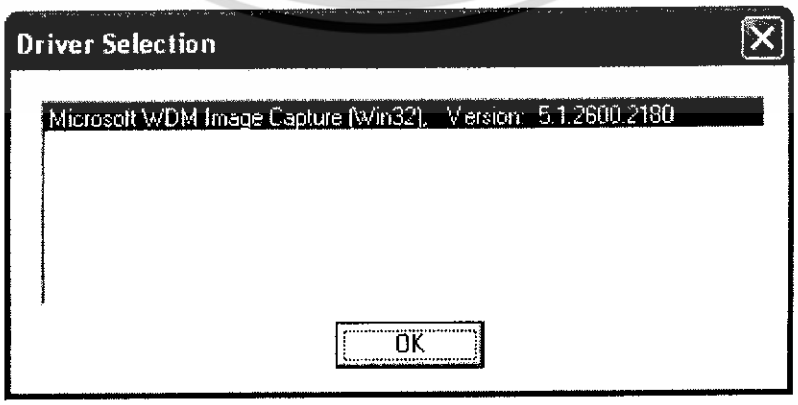
รูปที่ 2.18 แสดงหน้าต่าง Video Source

- **ShowDriverDlg Method**

Syntax : BOOL object.ShowDriverDlg()

ใช้เมื่อต้องการแสดง DialogBox สำหรับเลือก Driver

เมธอดนี้จะส่งกลับค่า เป็น TRUE ถ้าสามารถทำการแสดง Format Dialog ได้สำเร็จ



รูปที่ 2.19 แสดงหน้าต่าง Driver Selection

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **GetlastErrorString Method**

Syntax : LPCTSTR object.GetLastErrorString()

ใช้ดึงข้อความแสดงถึงความผิดพลาดล่าสุด ของ VideoOCX

- **AVISaveFrameInit Method**

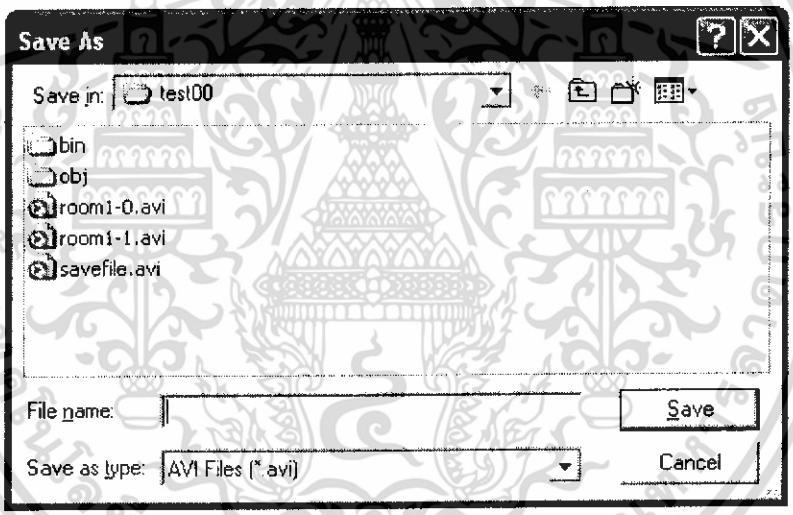
BOOL object.AVISaveFrameInit(String filename,

long imagehandle,

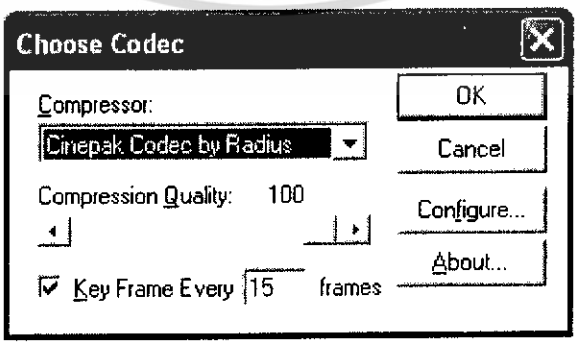
BOOL showcompressiondialog

)

ใช้เพื่อเปิดไฟล์ AVI หากกำหนด filename เป็น "" จะทำให้ VideoOCX แสดง DialogBox สำหรับ ตั้งชื่อไฟล์ที่ต้องการบันทึก ขนาดของภาพ จะถูกกำหนดตาม Imagehandle ที่ระบุไว้ หากกำหนด showcompressiondialog เป็น TRUE จะแสดง DialogBox สำหรับกำหนด ค่าในการทำ Compression



รูปที่ 2.20 แสดงหน้าต่างบันทึก



รูปที่ 2.21 แสดงหน้าต่าง Choose Code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **AVISaveFrameClose Method**

Syntax : BOOL object.AVISaveFrameClose()

ใช้เมื่อต้องการสิ้นสุดการบันทึกไฟล์ AVI และปิดไฟล์

- **AVISaveFrameAdd Method**

Syntax : BOOL object.AVISaveFrameAdd(long imagehandle)

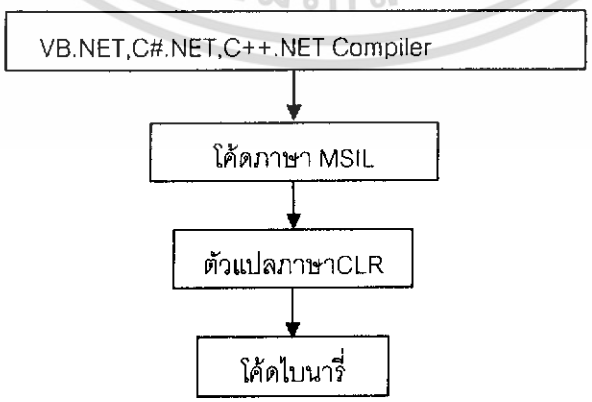
ใช้เพิ่มเฟรมภาพจากหน่วยความจำที่อ้างอิงโดยตัวแปร imagehandle ลงในไฟล์ AVI ที่ได้เปิดเตรียมไว้

2.4 Visual C#.NET

Visual stdio.NET ถือเป็นเครื่องมือพัฒนา Application ตัวแรกที่ออกมารองรับการทำงานตามแนวความคิดของสถาปัตยกรรม .NET Framework ประกอบไปด้วยเครื่องมือต่างๆดังนี้ Visual Basic.NET, Visual C#.NET, Visual C++.NET, Visual ASP.NET

ภาษา Visual C#.NET ถือเป็นภาษาที่เกิดขึ้นมาพร้อมกับแนวความคิดของการเขียนโปรแกรมในยุค .NET อาจกล่าวได้ว่าภาษา Visual C# คือภาษาต้นแบบในการเขียนโปรแกรม .NET นั่นเอง โดยที่ภาษาอื่นๆที่เกิดมาก่อนหน้านี้จะต้องปรับตัวเข้าหา .NET ทั้งหมดซึ่งสามารถสังเกตได้ว่า ไวยกรณ์การใช้งานของแต่ละภาษานั้นล้วนแล้วแต่ ถูกปรับเปลี่ยนไปจากเวอร์ชั่นก่อนหน้านี้อย่างสิ้นเชิง

หลักการทำงานของตัวแปรภาษา C# ความสำคัญอีกอย่างหนึ่งของภาษาต่างๆใน .NET ก็คือตัวแปรภาษาหรือที่เราเรียกว่า compiler จากอดีตที่ผ่านมาเราจะพบว่าแต่ละภาษาจะมีตัวแปลภาษาเป็นของตัวเอง แต่สำหรับภาษาต่างๆที่อยู่ใน Visual stdio.NET แล้ว Microsoft ได้ปรับปรุงตัวแปลภาษา เปลี่ยนไปอย่างสิ้นเชิง โดยไม่ว่าจะพัฒนา application ด้วยภาษาใดก็ตามจะอาศัยตัวแปรที่เรียกว่า CLR (Common Language Runtime) ทำหน้าที่แปลง code ไปสู่ภาษาเครื่อง



รูปที่ 2.22 แสดงหลักการทำงานของตัวแปลภาษา Visual C#

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.22 เมื่อเกิดการแปลโค้ดที่มาจากภาใดๆก็ตามใน .NET จะอาศัย CLR ทำหน้าที่แปลออกมาเป็นภาษากลางที่เรียกว่า IL (Intermediate Language) ก่อน เมื่อได้โค้ดของ IL มาแล้ว ถ้าต้องการแปลออกมาเป็นภาษาเครื่องก็จะอาศัยหลักการทำงานของ Virtual Machine แปลภาษา IL อีกครั้งหนึ่ง โดยอาศัย Compiler JIT (Just-In-Time) ซึ่งมีข้อดีคือ สามารถใช้ภาษาอะไรก็ได้ที่ถนัดเขียนโปรแกรมกับ .NET ก็ได้ผลลัพธ์เหมือนกัน หรืออาจจะสร้าง Application จากภาษาต่างๆ มากกว่าหนึ่งภาษาก็ได้ ทำให้ยืดหยุ่นต่อการพัฒนา Application แบบเป็นทีม

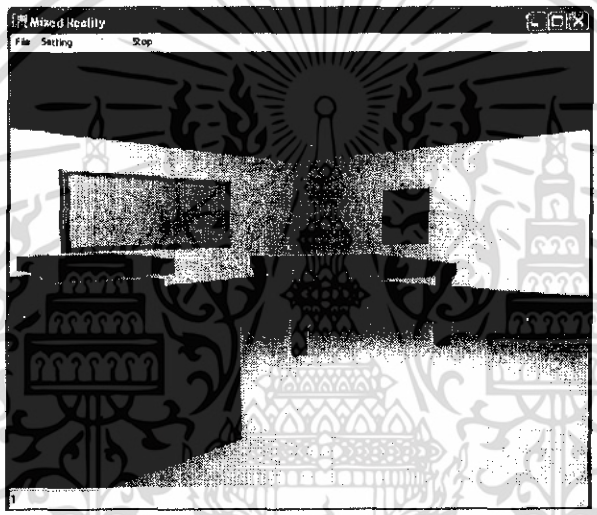
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและพัฒนา

3.1 การออกแบบระบบโดยภาครวม

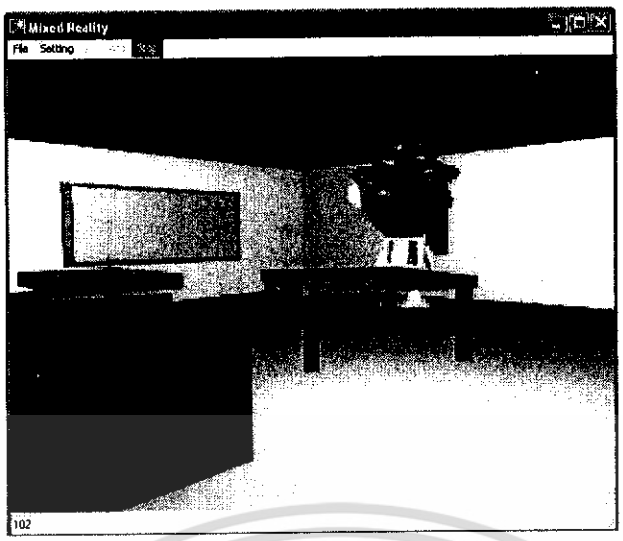
การพัฒนาสื่อแสดงผลในรูปแบบเสมือนจริงผสมเป็นการนำเอาภาพวิดีโอที่ได้จากกล้องมาผสมผสานกับภาพฉากที่เตรียมไว้ ซึ่งในการนำภาพมาผสมผสานกันนั้น สามารถแสดงภาพคนที่ได้จากกล้องให้เข้าไปอยู่ในสภาพแวดล้อมที่วัตถุสามารถแสดงอยู่หน้า หรือหลังภาพคนได้



รูปที่ 3.1 แสดงภาพฉากหลังที่เตรียมไว้

รูปที่ 3.2 แสดงภาพฉากหน้า

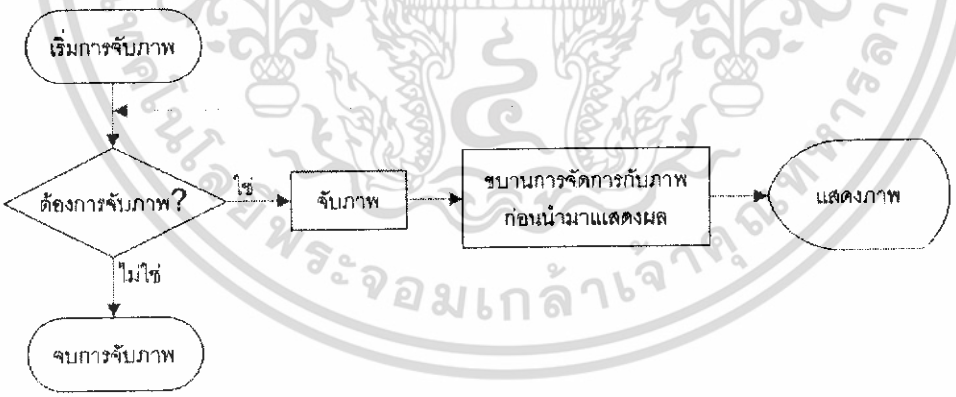
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 แสดงภาพรวมของการนำภาพมาผสมผสานกัน

3.2 การนำภาพจากกล้องเข้ามาใช้งาน

นำภาพจากกล้อง web camera เข้ามาใช้งานโดยใช้ ActiveX Control ที่ชื่อว่า VideoOCX โดยจะต้องทำการ Initial ตัว Control ก่อน และเมื่อเริ่มทำการจับภาพ จะสร้าง Thread ขึ้นมาเพื่อทำการจับภาพ และส่วนการจัดการต่างๆ เกี่ยวกับภาพ โดย Thread นี้จะทำงานจนกว่า จะยกเลิกการจับภาพ



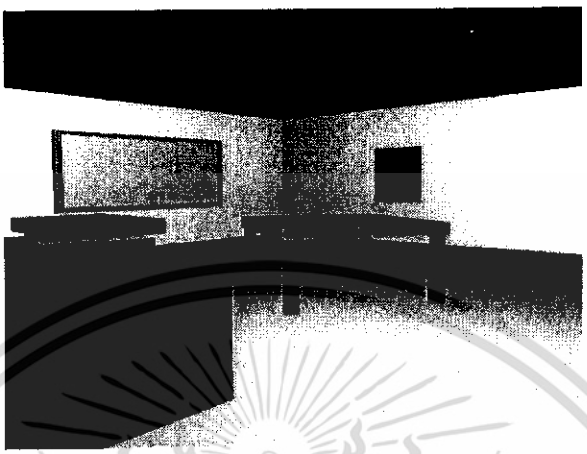
รูปที่ 3.4 แสดงการนำภาพจากกล้องเข้ามาใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การนำภาพฉากเข้ามาใช้งาน

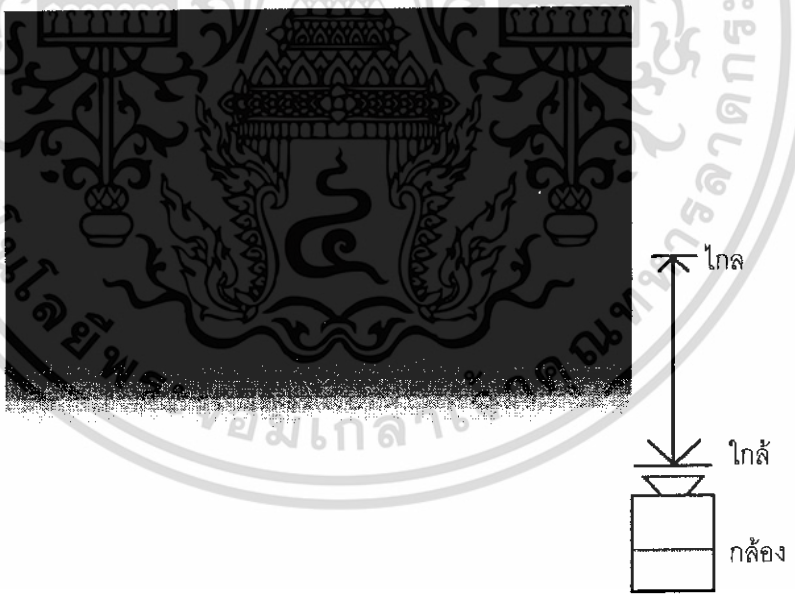
ส่วนของภาพฉากจะประกอบด้วยไฟล์ต่างๆ ได้แก่

- ไฟล์ที่บันทึกภาพฉาก



รูปที่ 3.5 แสดงภาพฉากหลัง

- ไฟล์ที่บันทึกภาพที่แสดงค่า Depth-key ของพื้นฉาก



รูปที่ 3.6 แสดงภาพ Depth-key ของพื้น และลักษณะของมุมกล้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

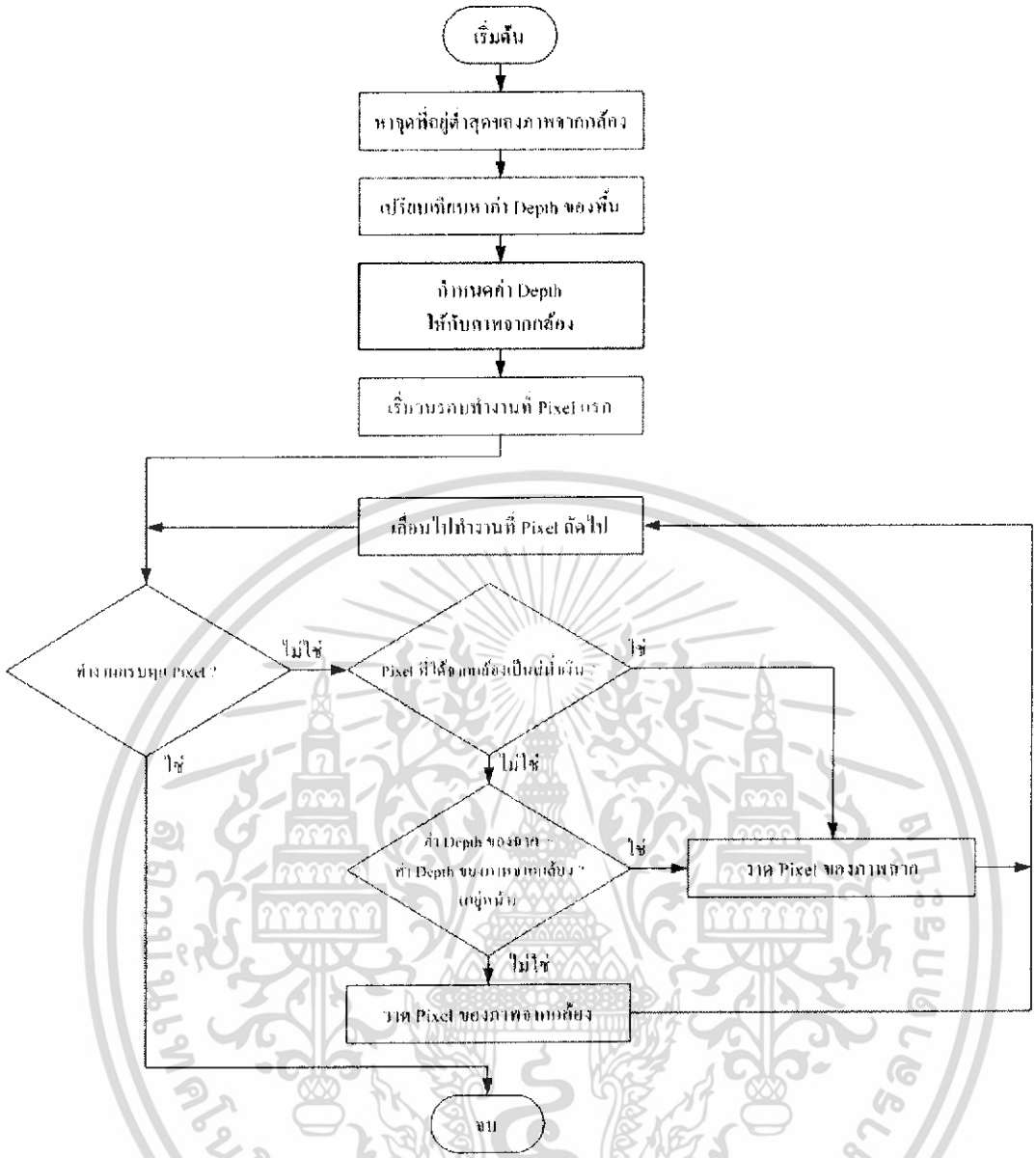
ไฟล์ที่บันทึกภาพที่แสดงค่า Depth-key ของวัตถุในฉาก

รูปที่ 3.7 แสดงภาพ Depth-key ของวัตถุในฉาก

3.4 การนำภาพมาผสมผสานกัน

นำภาพที่ได้จากกล้องมาตัดส่วนที่เป็นฉากสีน้ำเงินออกโดยใช้เทคนิค Chroma keying ทำให้ได้ภาพคนออกมา หลังจากนั้นค้นหาตำแหน่งเท้าของคน ได้แก่ ส่วนที่ต่ำสุดของภาพ ดึงค่า Depth-key ของพื้น โดยอ้างอิงจากจุดที่อยู่ต่ำสุดของภาพคน นำไปเป็นค่า Depth-key ของภาพคน วาดภาพฉากบางส่วนแทนส่วนที่เป็นฉากสีน้ำเงิน และวาดในส่วนที่มีค่า Depth-key ของวัตถุในฉากมากกว่า(อยู่หน้า) ค่า Depth-key ของภาพคน ทำให้เห็นภาพในลักษณะสามารถมีวัตถุอยู่ด้านหน้าของภาพคนได้ตามความเหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 แสดงกระบวนการผสมผสานภาพในหนึ่งเฟรม

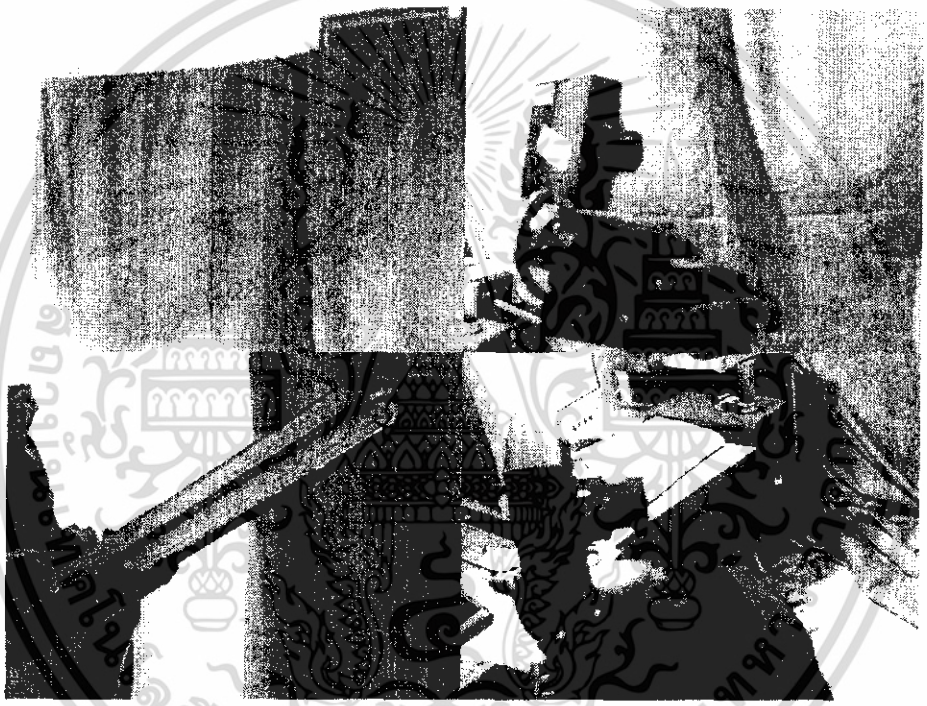
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การใช้งานระบบ

4.1 ติดตั้งอุปกรณ์ฮาร์ดแวร์ระบบ

การใช้งานระบบ Mixed Reality For Presentation จะต้องติดตั้งอุปกรณ์ สำคัญๆ ได้แก่ กล้อง webcamera , ฉาก blue Screen , และแหล่งกำเนิดแสง

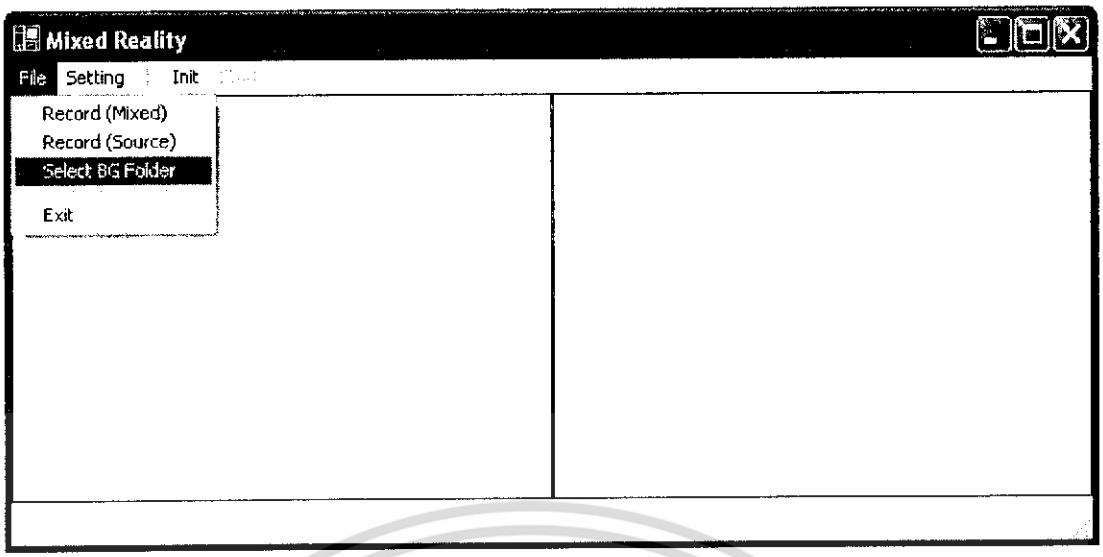


รูปที่ 4.1 แสดงภาพอุปกรณ์ที่ใช้งาน

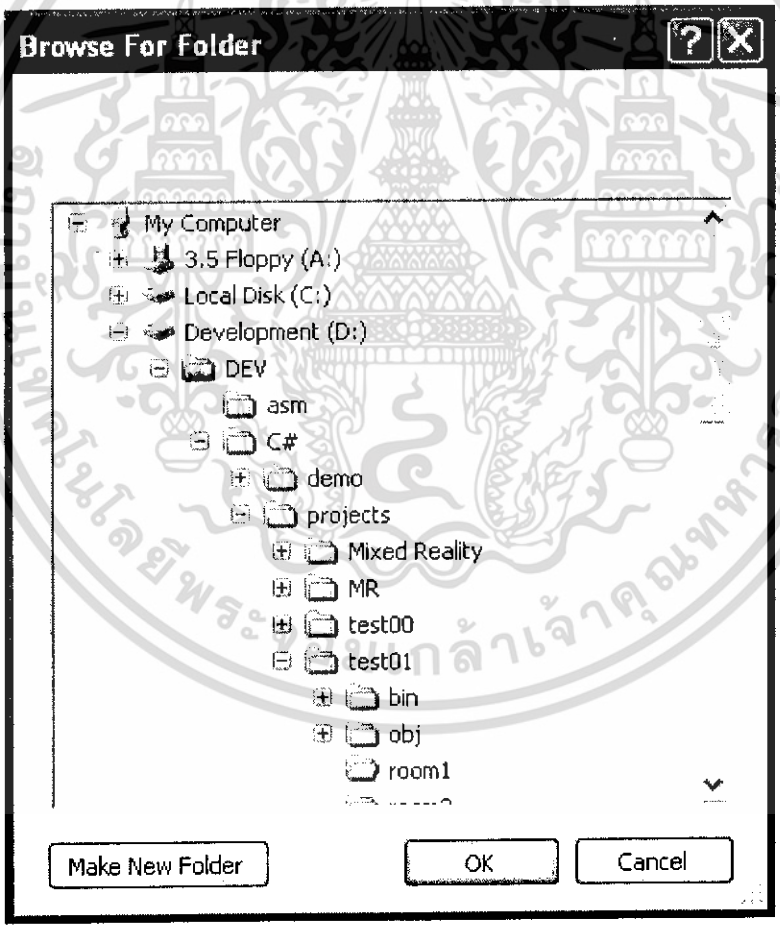
4.2 การนำภาพจากไฟล์เข้ามาใช้ในโปรแกรม

- การเลือกโฟลเดอร์ที่มีไฟล์ภาพเข้ามาใช้เป็น background ไปที่ File จากนั้นเลือก Select BG Folder (ตามรูปที่ 4.2) จะปรากฏ หน้าต่าง Brower for folder ให้เลือกโฟลเดอร์ที่ต้องการเพื่อนำไปเป็น Background

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



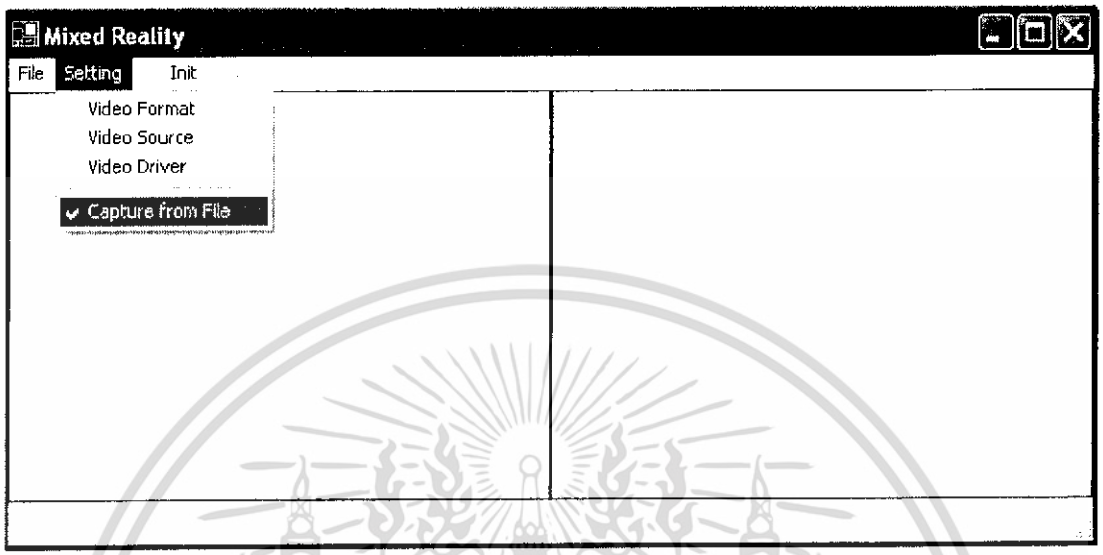
รูปที่ 4.2 เมนูสำหรับเลือก folder



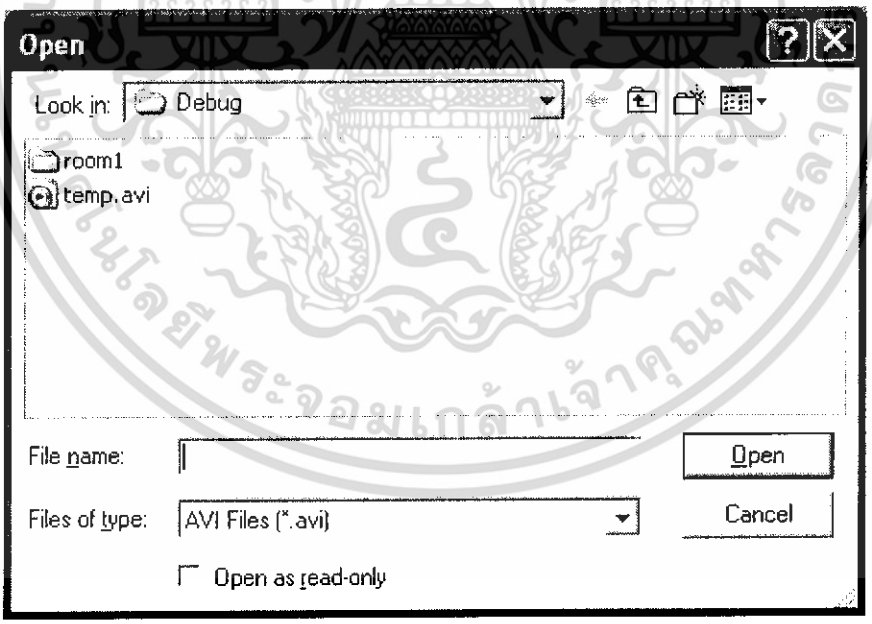
รูปที่ 4.3 เลือก folder สำหรับใช้เป็น background

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การเลือกไฟล์ภาพมาใช้เป็น foreground โดยไปที่ setting และเลือกเช็คที่ capture from file ดังแสดงในรูปที่ 4.4 เมื่อสั่งงานคำสั่ง Init จะปรากฏหน้าต่าง open เพื่อเลือกไฟล์ที่จะนำมารวมกับภาพ background ดังรูปที่ 4.5



รูปที่ 4.4 แสดงเมนูเช็คเพื่อ capture from file



รูปที่ 4.5 เลือกไฟล์ที่นำมาเป็นฉากหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

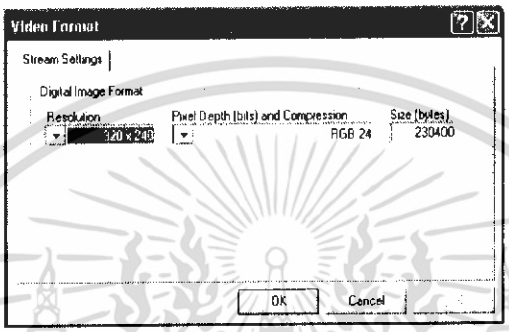
4.3 การกำหนดค่าพารามิเตอร์ต่างๆเกี่ยวกับกล้อง

การตั้งค่าต่างๆ เลือกได้ที่เมนู Setting มีรายการให้เลือกได้แก่

4.3.1 Video Format

ส่วนนี้ใช้เลือก ลักษณะของภาพ video ที่ใช้ได้แก่

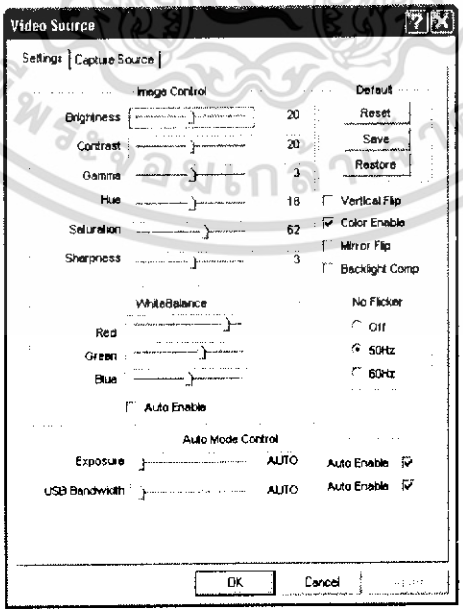
- Resolution ขนาดของภาพ
- Pixel Depth เป็น format ของสี



รูปที่ 4.6 แสดงหน้าต่างการปรับค่า Video Format

4.3.2 Video Source

ส่วนนี้ใช้เลือกปรับคุณสมบัติต่างๆ ของภาพได้แก่ Image Control , White Balance , Auto Mode Control และยังใช้เลือก Capture Source

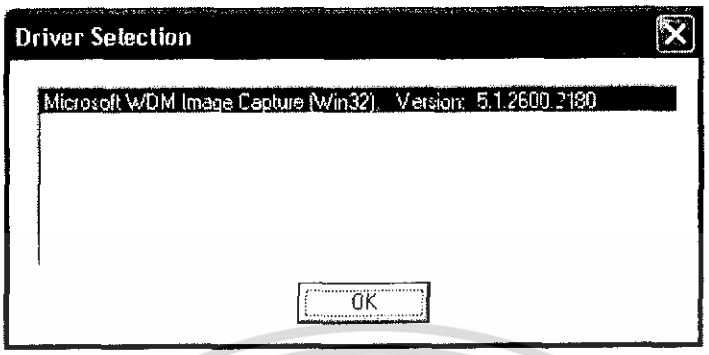


รูปที่ 4.7 แสดงหน้าต่างการปรับค่า Video Source

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 Video Driver

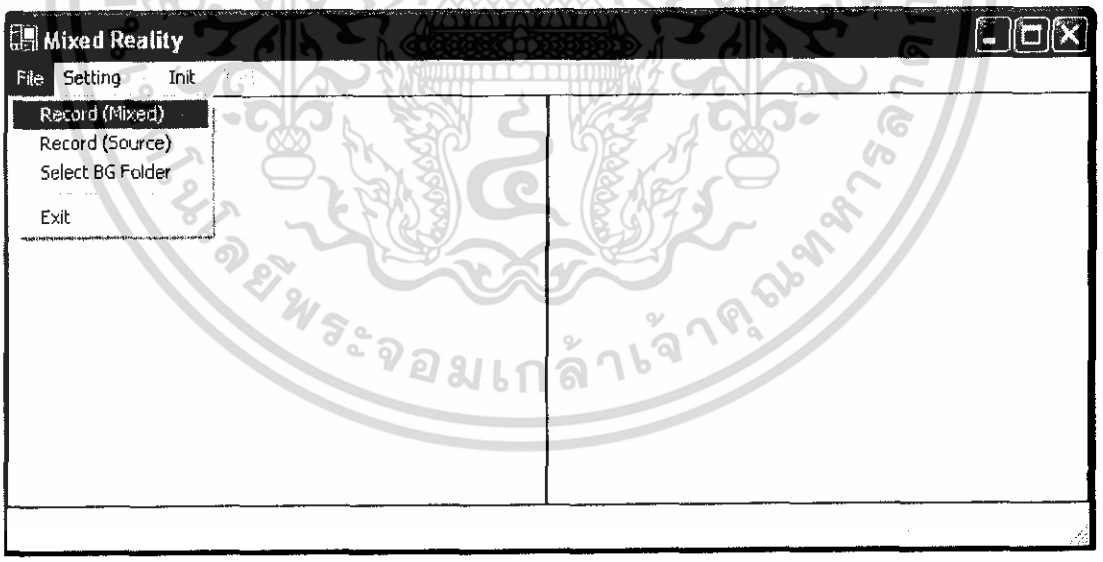
ส่วนนี้ใช้เลือก Driver ที่ใช้กับกล้อง



รูปที่ 4.8 แสดงหน้าต่างการเลือก Video Driver

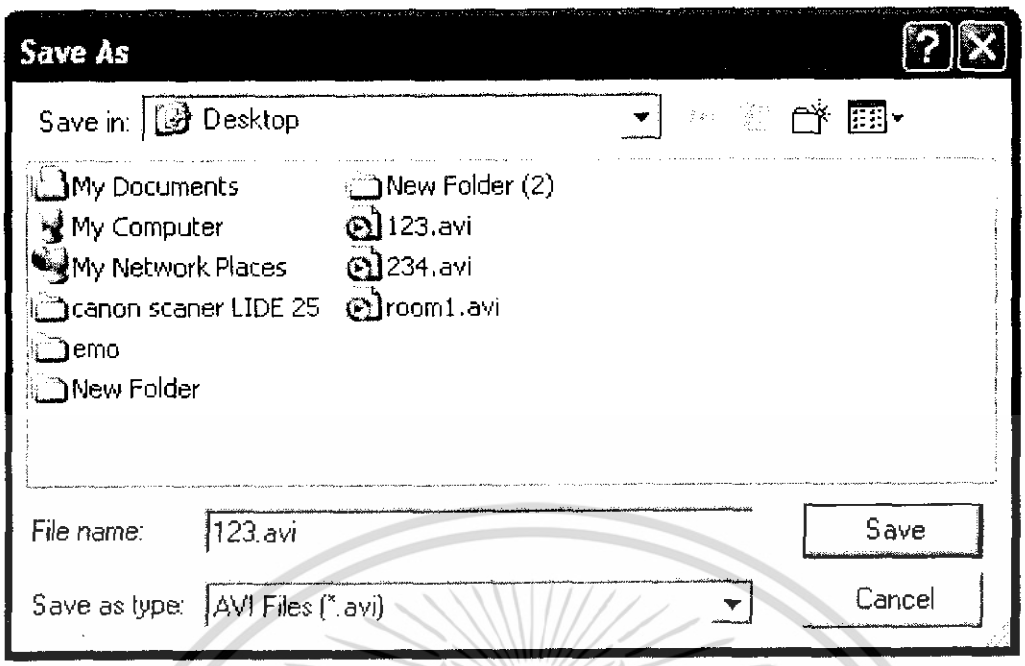
4.4 บันทึกเป็นไฟล์วิดีโอ

การบันทึกไฟล์ให้เช็คที่ Record ก่อน (รูปที่ 4.7) สั่งงาน start เมื่อสั่งงานคำสั่ง start จะขึ้นหน้าต่าง Save As ให้ตั้งชื่อไฟล์ที่จะทำการ save (รูปที่ 4.8)



***รูปที่ 4.9** แสดงเมนูเช็คเพื่อ record*

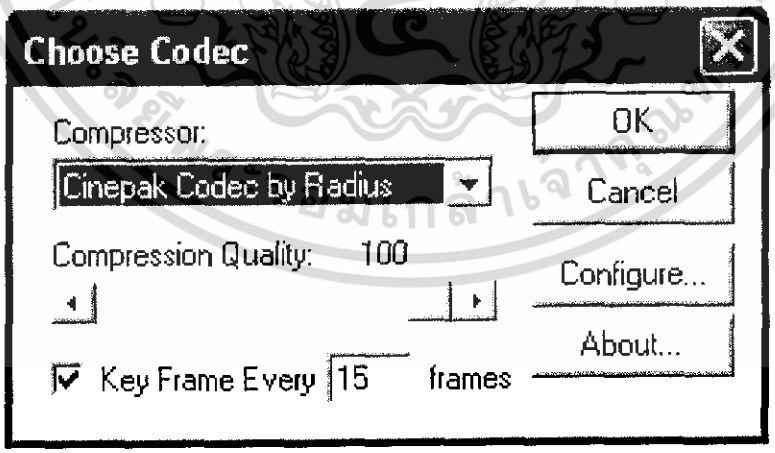
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 แสดงหน้าต่างสำหรับตั้งชื่อไฟล์ที่จะบันทึก

เมื่อสั่งงานคำสั่ง save จะขึ้นหน้าต่าง Choose Code สำหรับตั้งค่าความละเอียด

- Compressor วิธีการบีบอัด
- Compression Quality ค่าการบีบอัดไฟล์
- Key Frame Every n frame จำนวน frame ที่บันทึก



รูปที่ 4.11 แสดงหน้าต่างปรับค่าความละเอียดและวิธีการบีบอัดภาพ

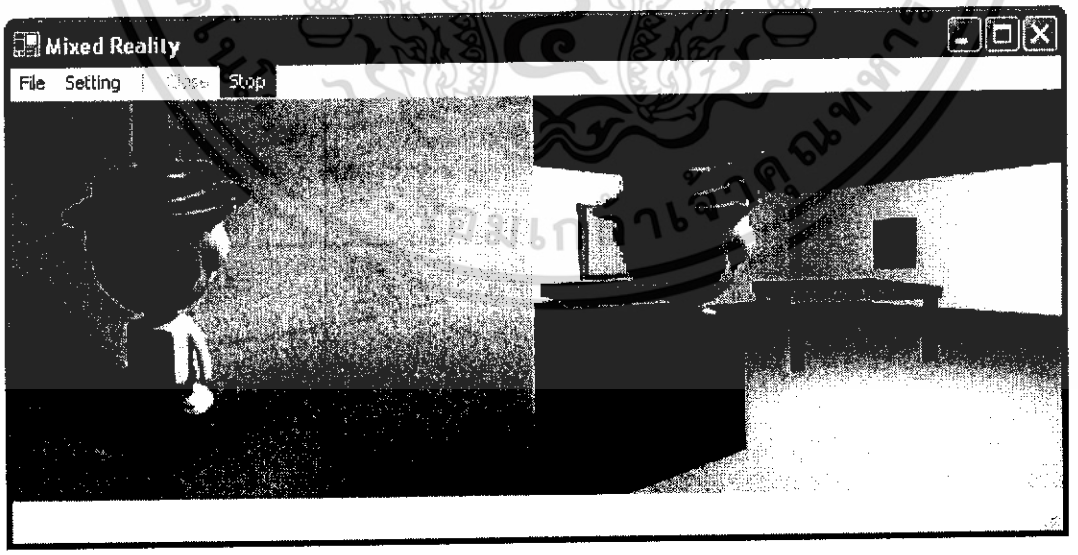
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 การนำภาพถ่ายจากกล้องวิดีโอเข้ามาใช้งาน

คลิ๊กที่ Init เพื่อเริ่มใช้งานกล้อง จะปรากฏภาพ Preview จากกล้อง ที่ช่องแสดงภาพจากกล้อง ทำการปรับตำแหน่งกล้องกับฉากให้อยู่ในตำแหน่งที่เหมาะสม

รูปที่ 4.12 แสดงภาพ Preview ที่ได้จากกล้อง

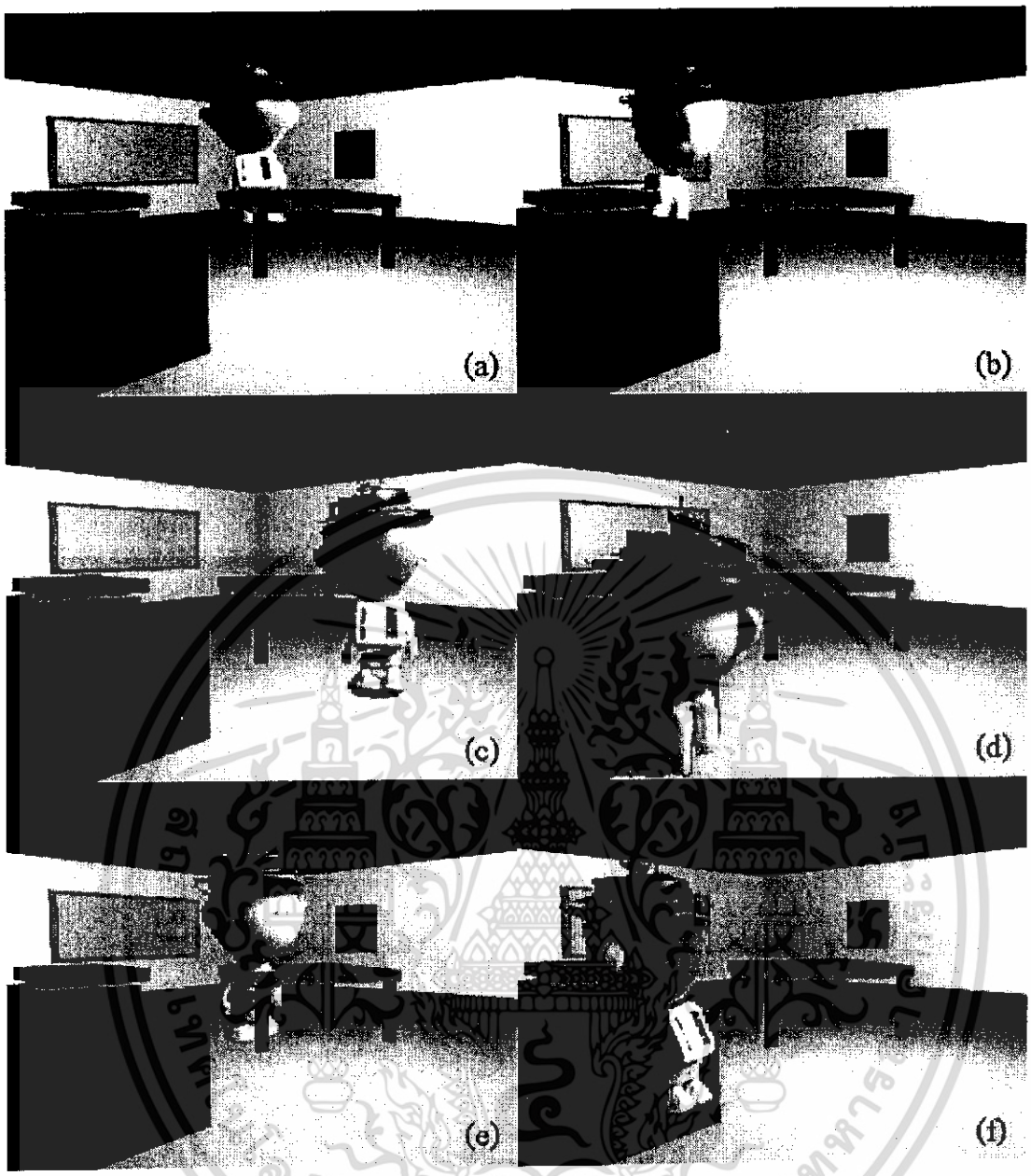
คลิ๊กที่ Start เพื่อเริ่มทำการจับภาพ เป็นการรวมภาพทั้งสองเข้าด้วยกัน (ภาพ background กับ ภาพจากกล้อง)



รูปที่ 4.13 แสดงการรวมภาพจากไฟล์ผสมกับภาพจากกล้อง

เมื่อต้องการหยุดการบันทึกให้กด stop ที่เมนู และเลือก close ที่เมนู เพื่อออกจากโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 แสดงการผสมผสานภาพในกรณีต่างๆ

จากรูปที่ 14 (a) และ รูปที่ 14 (b) แสดงการผสมผสานภาพ โดยที่ภาพตัวหุ่นจากกล้องวิดีโอ ยืนอยู่ด้านหลังโต๊ะซึ่งเป็นวัตถุที่อยู่ในภาพฉาก ปรากฏเป็นภาพโต๊ะบังตัวหุ่น

จากรูปที่ 14 (c) และ รูปที่ 14 (d) แสดงการผสมผสานภาพ โดยที่ภาพตัวหุ่นจากกล้องวิดีโอ ยืนอยู่ด้านหน้าโต๊ะซึ่งเป็นวัตถุที่อยู่ในภาพฉาก ปรากฏเป็นภาพตัวหุ่นบังโต๊ะ

จากรูปที่ 14 (e) และ รูปที่ 14 (f) แสดงการผสมผสานภาพ ซึ่งเกิดความผิดพลาดเนื่องจากตัวหุ่นยืนอยู่ในระยะที่อยู่บริเวณกลางโต๊ะ ทำให้ปรากฏเป็นภาพตัวหุ่นโผล่ออกมาจากภาพโต๊ะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และสรุป

5.1 บทสรุป

ความต้องการในการใช้เทคโนโลยี multimedia มีเพิ่มขึ้นมากมาย ดังเช่นในการนำเสนอผลงานทางสื่อโทรทัศน์ ภาพยนตร์ การ์ตูน หรืองานนำเสนอผลงานอื่นๆ อีกมากมาย แต่อุปกรณ์ที่มีประสิทธิภาพนั้นนำมาซึ่งการลงทุนที่สูง

โครงงานการนำเสนอผลงานด้วยสภาวะแวดล้อมผสม นั้นได้พัฒนาในส่วนของ software เพื่อลดต้นทุนทางด้าน hardware โดยสามารถนำภาพจากกล้องมารวมกับอีกภาพโดยใช้เทคนิค software chroma keying โดยภาพที่นำมาใช้สามารถใช้ได้ทั้งภาพนิ่ง และภาพเคลื่อนไหวจากไฟล์วิดีโอ สามารถแสดงภาพที่มีลักษณะการซ้อนทับกันของวัตถุที่มีระยะใกล้ไกลได้ โปรแกรมที่สร้างขึ้นมานั้นสามารถบันทึกเป็นไฟล์วิดีโอได้

5.2 วิจารณ์สิ่งที่ได้จากโครงงาน

จากการพัฒนาโครงงานการพัฒนาสื่อแสดงผลในรูปแบบเสมือนจริงผสม สามารถสร้างภาพที่มีลักษณะของสภาพแวดล้อมผสม ที่นำเอาภาพจากกล้อง เข้ามารวมกับภาพฉาก อีกภาพหนึ่ง ภาพที่ผสมผสานสามารถแสดง ตำแหน่งในทางลึก ได้โดยการแสดงภาพของวัตถุที่อยู่ใกล้ซ้อนทับภาพของคนซึ่งอยู่ไกลกว่า และ แสดงภาพวัตถุอยู่หลังภาพคน เมื่อวัตถุนั้นอยู่ไกลกว่า

ภาพที่ได้หลังจากการตัดฉากสีน้ำเงินออกแล้ว ในส่วนขอบที่ตัด มีความต่างของสีชัดเจน ดูไม่สวยงาม ไม่สมจริง

สามารถบันทึก ภาพที่มีการผสมผสานเป็นไฟล์ภาพยนตร์ ฟอร์แมต AVI ได้ ภาพเคลื่อนไหวที่ได้ ดูต่อเนื่องในระดับที่ยอมรับได้

การสร้างฉากที่ใช้ จะต้องสร้างภาพที่เก็บค่า Depth-key ขึ้นเอง ยังไม่สามารถเก็บเอาค่า Depth-key จากภาพฉากได้โดยตรง ทำให้เกิดความยุ่งยากในการสร้างฉาก

5.3 ปัญหาอุปสรรคและแนวทางแก้ไข

กล้องที่ใช้เป็นกล้อง webcam มีระยะโฟกัสแคบ ทำให้ภาพที่มีระยะต่างกัน มีความชัดเจนไม่สม่ำเสมอ ทำให้ภาพที่นำมาตัดไม่ชัดเจน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างฉากที่ใช้ จะต้องสร้างภาพที่เก็บค่า Depth-key ขึ้นเอง ยังไม่สามารถเก็บเอาค่า *Depth-key* จากภาพฉากได้โดยตรง ทำให้เกิดความยุ่งยากในการสร้างฉาก แนวทางแก้ไข ใช้วิธีการสร้างฉาก โดยการทำ 3D reconstruction จากภาพสองภาพ

5.4 แนวทางการพัฒนาต่อ

พัฒนาให้ขอบที่ตัดจากฉากสีน้ำเงินมีความนุ่มนวลสมจริงขึ้น จากที่เป็นขอบตัดชัดเจน

พัฒนาให้ใช้ฉากที่เป็น ภาพเคลื่อนไหว

พัฒนาให้ใช้กับฉากที่เป็น Realtime 3D Graphics

พัฒนาให้รวมเสียงเข้าไว้ในไฟล์วิดีโอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] ศุภชัย สมพานิช,2546,"คู่มือการเขียนโปรแกรมและการใช้งาน Visual C# .NET " สำนักพิมพ์อินโฟเพรส นนทบุรี

[2] พรพล สาครินทร์,กฤษฎา แก้วมณี ,2544 "พื้นฐานการก้าวสู่โลก 3มิติ 3D Graphics" สำนักพิมพ์ ซัคเซส มีเดีย กรุงเทพฯ

[3] VideoOCX , 2002 , "ActiveX Control for Video Capture Programming" [Online] URL : http://www.videoocx.de/

[4] F. van den Bergh and V. Lalioti, "Software Chroma Keying in an Immersive Virtual Environment", South African Computer Journal, 1999, University of Pretoria

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้