

รายงานการวิจัยฉบับสมบูรณ์

อัลกอริธึมสำหรับบาร์โค้ดสองมิติชนิดความจุสูงโดยใช้บาร์โค้ดสี  
Algorithm for High Capacitance 2D-Barcode using Color Barcode

นายลักกะพันธ์ คล้ายดอกจันทร์  
นายเกษมสุข เสพศิริสุข

ได้รับทุนสนับสนุนงานวิจัยจากเงินรายได้ ประจำปีงบประมาณ พ.ศ. 2555  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
วิทยาเขตชุมพรเขตรอุดมศักดิ์ จังหวัดชุมพร

b. 12612741  
i. ....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

เลขหมู่.....  
เลขทะเบียน 137325  
รับเดือนปี 22 ส.ย. 2558

ชื่อโครงการ (ภาษาไทย)	อัลกอริธึมสำหรับบาร์โค้ดสองมิติชนิดความสูงโดยใช้บาร์โค้ดสี
แหล่งเงิน	เงินรายได้
ประจำปีงบประมาณ	2555 จำนวนเงินที่ได้รับการสนับสนุน 165,000 บาท
ระยะเวลาทำการวิจัย	1 ปี 6 เดือน ตั้งแต่ 1 ตุลาคม 2554 ถึง 31 มีนาคม 2556
นายสักกะพันธ์ คล้ายดอกจันทร์	หัวหน้าโครงการ หลักสูตรวิศวกรรมอิเล็กทรอนิกส์ วิทยาเขตชุมพรฯ
นายเกษมสุข เสพศิริสุข	ผู้ร่วมโครงการ หลักสูตรวิศวกรรมอิเล็กทรอนิกส์ วิทยาเขตชุมพรฯ

### บทคัดย่อ

งานวิจัยนี้นำเสนอการพัฒนาอัลกอริธึมบาร์โค้ดสี โดยในการตรวจจับจะใช้อุปกรณ์รับภาพสีเช่นกล้องเว็บแคมบันทึกภาพแล้วนำมาหาค่าสีเพื่อแปลงกลับเป็นข้อมูล การสร้างบาร์โค้ดสีนั้นเป็นข้อมูลภาพสองมิติ โดยทำการเลือกสีจากผลการทดลองที่มีความแยกชัดมากที่สุด 16 สี แทนข้อมูลจำนวน 4 บิต จากผลการทดลองพบว่าจำเป็นต้องส่งสีอ้างอิงด้วย โดยอัลกอริธึมที่นำเสนอสามารถทนต่อสภาพความเข้มแสงในหลายระดับได้ และทนต่อการบีบอัด JPEG ที่ค่าคุณภาพไม่ต่ำกว่า 60

คำสำคัญ : บาร์โค้ด, อัลกอริธึม, บาร์โค้ดสี, การประมวลผลภาพ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Research Title: Algorithm for High Capacitance 2D-Barcode using Color Barcode  
Researcher: Sakkapan Klaidokchan and Kasemsuk Sepsirisuk  
Faculty: Chumphon Campus Department: Electronics Engineering

## ABSTRACT

This research describes a development of algorithm of 2D color barcode. In decoding process, an imaging device, webcam, is used for capturing barcode image which the encoded data is extracted from. In encoding process, the 16 distinguishable colors selected from experiments represent 4 bits of data. From the experiments, the reference colors are required to be sent with data for decoding. The proposed algorithm can withstand the JPEG compression with quality at 60.

**Keywords:** Barcode, Algorithm, Color Barcode, Image Processing



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

งานวิจัยชิ้นนี้สามารถสำเร็จลุล่วงได้เป็นอย่างดี ด้วยการสนับสนุนและความช่วยเหลือจากหลายฝ่าย ซึ่งทางทีมงานวิจัยขอขอบพระคุณทุก ๆ ท่านดังต่อไปนี้

ขอขอบพระคุณบิดา มารดา และอาจารย์ ซึ่งคอยให้การอบรมสั่งสอน ให้คำแนะนำและให้การสนับสนุนอย่างเต็มที่ตลอดมา โดยขอกราบขอบพระคุณไว้ ณ โอกาสนี้

การวิจัยครั้งนี้ได้รับทุนสนับสนุนการวิจัยจากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง วิทยาเขตชุมพรเขตรอุดมศักดิ์ จากแหล่งทุนเงินรายได้ ประจำปีงบประมาณ พ.ศ.2555

คุณค่าและประโยชน์อันพึงบังเกิดจากงานวิจัยนี้ ทีมวิจัยขอมอบแต่ผู้มีพระคุณทุกท่าน



นายสั๊กกะพันธ์ คล้ายดอกจันทร์

นายเกษมสุข เสพศิริสุข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญภาพ.....	VII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์.....	1
1.3 ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย.....	1
1.4 ขอบเขตของการวิจัย.....	2
1.5 ประโยชน์หรือผลที่คาดว่าจะได้รับ.....	2
1.6 แผนการถ่ายทอดเทคโนโลยีหรือผลการวิจัยสู่กลุ่มเป้าหมาย.....	2
บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง.....	3
2.1 การประมวลผลภาพ (Image Processing).....	3
2.2 รูปแบบสีแบบอาร์จีบี.....	4
2.3 รูปแบบสีแบบระดับสีเทา.....	4
2.4 การตรวจสอบความเปลี่ยนแปลงของภาพ (Image differencing).....	5
2.5 จุดภาพหรือพิกเซล(pixel).....	6
2.6 ความละเอียดของรูปภาพ และความละเอียดในการแสดงผล.....	7
2.7 ทบทวนวรรณกรรมที่เกี่ยวข้อง.....	8
บทที่ 3 การออกแบบการทำงาน.....	10
3.1 โครงสร้างของระบบ.....	10
3.2 การออกแบบบาร์โค้ดสี.....	10
3.3 การเลือกใช้สีในบาร์โค้ดสี.....	12
3.4 อัลกอริธึมในการอ่านบาร์โค้ดสี.....	15
บทที่ 4 การทดลองและผลการทดลอง.....	18
4.1 สภาพแวดล้อมในการทดลองและผลการทดลองพื้นฐาน.....	18
4.2 การทดลองเรื่องปริมาณข้อมูลในขั้นตอนการสร้างบาร์โค้ด.....	20
4.3 การทดสอบอัลกอริธึมภายใต้สภาวะแสงที่ต่างกัน.....	20
4.4 การทดสอบอัลกอริธึมในภาพบาร์โค้ดที่ถูกบีบอัดด้วย JPEG.....	22

## สารบัญ(ต่อ)

	หน้า
บทที่ 5 สรุปผลการวิจัย.....	24
5.1 ข้อสรุป.....	24
5.2 ปัญหาที่พบในระหว่างการทำงานวิจัย.....	24
5.3 แนวทางในการพัฒนาต่อในอนาคต.....	24
บรรณานุกรม.....	25
ภาคผนวก.....	27
ภาคผนวก ก โค้ดของโปรแกรม.....	27
ภาคผนวก ข สรุปค่าใช้จ่ายการดำเนินโครงการวิจัย.....	42
ประวัตินักวิจัย.....	44



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

ตารางที่	หน้า
3.1 ตัวแทนสีที่เลือก.....	14
4.1 ผลการทดลองสร้างและอ่านบาร์โค้ดสี.....	19
4.2 คุณลักษณะของบาร์โค้ดสีที่มีปริมาณข้อมูลต่างกัน.....	20
4.3 ผลการอ่านบาร์โค้ดในสภาวะแสงต่าง ๆ.....	21
4.4 ผลการอ่านบาร์โค้ดที่ไฟล์ภาพถูกบีบอัดด้วย JPEG.....	22



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญภาพ

ภาพที่	หน้า
2.1 ความสัมพันธ์ของสัญญาณภาพเทียบกับเวลา.....	3
2.2 รูปแบบของสีแบบ RGB.....	4
2.3 รูปแบบสีแบบระดับสีเทา.....	4
2.4 ตัวอย่างการทำการตรวจสอบความเปลี่ยนแปลงของภาพ.....	5
2.5 เปรียบเทียบภาพที่มี Salt-and-pepper noise กับภาพที่ใช้ใช้ Median filter.....	6
2.6 ขนาดของพิกเซล (pixel).....	6
2.7 ความละเอียดของภาพที่แตกต่างกัน.....	7
2.8 บาร์โค้ด 2 มิติชนิดต่าง ๆ.....	8
2.9 บาร์โค้ดสี เพื่อความสวยงาม.....	9
2.10 บาร์โค้ดสี และการเทียบขนาด.....	9
2.11 บาร์โค้ดสีแบบเมตริกซ์.....	9
3.1 โครงสร้างของระบบ.....	10
3.2 ส่วนประกอบบาร์โค้ดสี.....	11
3.3 การเรียงลำดับข้อมูลในบาร์โค้ดสี.....	11
3.4 ผลของแสงต่อค่าสี.....	12
3.5 กลุ่มสีที่ใช้ทดสอบความทนทานต่อการพิมพ์และถ่ายภาพ.....	13
3.6 การกระจายของสีต้นฉบับ.....	13
3.7 การกระจายของสีที่ถ่ายด้วยกล้องเว็บแคม.....	13
3.8 ภาพฉายการกระจายตัวของสี.....	14
3.9 โพลชาร์ตของอัลกอริธึมในการอ่านบาร์โค้ด.....	15
3.10 ภาพไบนารีและภาพฉายลงบนแกนแนวนอน.....	16
3.11 ภาพไบนารีเฉพาะส่วนแถบอ้างอิงแนวตั้งและภาพฉายลงบนแกนแนวตั้ง.....	16
3.12 ตำแหน่งของข้อมูลที่ระบุด้วยจุดสีขาว.....	17
4.1 หน้าต่างโปรแกรมในการสร้างและอ่านบาร์โค้ดสี.....	18
4.2 บาร์โค้ดสีในการทดสอบ.....	21
4.3 บาร์โค้ดที่ถ่ายได้จากกล้องเว็บแคมหลังการบีบอัดด้วย JPEG.....	23

# บทที่ 1

## บทนำ

ในบทนี้จะกล่าวถึงความเป็นมาและความสำคัญของปัญหา สาเหตุความสำคัญที่เป็นปัจจัยหลักรวมทั้งวัตถุประสงค์ของการวิจัย แนวความคิดที่ใช้ในการทำวิจัย ขอบเขตของการวิจัย ประโยชน์ที่คาดว่าจะได้รับ รวมทั้งโครงสร้างในการทำรายงานวิจัยฉบับนี้

### 1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันเลขดัชนีสำหรับการอ้างอิงมีบทบาทในชีวิตประจำวันอย่างมาก ไม่ว่าจะเป็นเลขบัตรประจำตัวประชาชน เลขหนังสือ ISBN. ตัวเลขไอพีของคอมพิวเตอร์ หรือชื่อเว็บไซต์ต่างๆ และสิ่งที่ไม่ได้กล่าวถึงก็คือรหัสสินค้าต่างๆ ซึ่งพบว่าเลขดัชนีหลาย ๆ ตัวนั้นต้องมีการเพิ่มจำนวนหลักขึ้นเพื่อให้รองรับกับจำนวนสิ่งของที่มากขึ้น เช่น การเพิ่มหลัก ISBN, การเพิ่มหลักไอพีคอมพิวเตอร์จาก 4 ชุด เป็น 6 ชุด หรือการออกแบบเลขทะเบียนรถเพื่อให้รองรับกับจำนวนรถที่เพิ่มขึ้น และปัจจุบันคอมพิวเตอร์หรือข้อมูลดิจิทัลถูกนำมาใช้แทนการป้อนข้อมูลด้วยมนุษย์ เช่น บาร์โค้ดสินค้า แท็ก (Tag) ต่างๆ ซึ่งจำเป็นต้องปรับปรุงให้รองรับจำนวนหลักข้อมูลที่เพิ่มขึ้นเช่นเดียวกัน จึงเป็นความสำคัญในการพัฒนารูปแบบการเก็บข้อมูลที่คอมพิวเตอร์อ่านได้ (machine readable) อย่างยิ่ง

ในปัจจุบันข้อมูลข่าวสารมีมากขึ้นจึงมีความต้องการข้อมูลจำนวนมาก บาร์โค้ดจากเดิมเป็นบาร์โค้ด 1 มิติถูกแทนที่ด้วยบาร์โค้ด 2 มิติที่พบเห็นกันอย่างแพร่หลายเช่น QR-Code ซึ่งสามารถเก็บข้อมูลสั้นๆ เช่น หมายเลขโทรศัพท์ หรือ ที่อยู่เว็บไซต์ เป็นต้น ต่อมาความต้องการเก็บข้อมูลปริมาณมากๆ ผลักดันให้มีการพัฒนาบาร์โค้ด 2 มิติให้สามารถเก็บข้อมูลได้มากขึ้นด้วยการเพิ่มจำนวนสีจาก ขาวและดำ เป็นหลากหลายสีเพื่อแทนข้อมูลได้หลายบิต แต่ทั้งนี้งานวิจัยด้านนี้ยังไม่เป็นที่แพร่หลายจึงเห็นควรต้องทำการพัฒนาให้มีประสิทธิภาพดีเหมาะสำหรับการใช้งานในสภาพแวดล้อมจริง โดยใช้กับอุปกรณ์ที่มีราคาถูก เช่น กล้องเว็บแคม เป็นต้น

### 1.2 วัตถุประสงค์

เพื่อศึกษาและออกแบบพัฒนาอัลกอริธึมของบาร์โค้ดสี ที่สามารถเข้ารหัสข้อมูลในปริมาณที่มากกว่าบาร์โค้ดขาวดำที่ใช้งานอยู่ในปัจจุบัน ทำการทดสอบประสิทธิภาพของอัลกอริธึมที่นำเสนอในสภาพแวดล้อมการใช้งานต่าง ๆ

### 1.3 ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย

ในการศึกษาวิจัยนี้คณะวิจัยมีกรอบแนวความคิดในการพัฒนาอัลกอริธึมสำหรับบาร์โค้ดสี 2 มิติ คือการคัดเลือกสีให้ได้จำนวนมากที่สุดที่มีความแยกชัดสามารถจำแนกออกเป็นสีต่างๆ ได้อย่างถูกต้องถึงแม้จะผ่านกระบวนการพิมพ์และการถ่ายภาพ (print and snapshot) มีการใช้เส้นลายอ้างอิงสำหรับการปรับปรุงพิกัดตำแหน่งและแก้ไขการหมุนเอียงของภาพถ่าย และเพื่อให้สามารถใช้งานได้ในสภาพที่แตกต่างกันจำเป็นต้องมีรหัสสีอ้างอิงพิมพ์ลงในบาร์โค้ด เพิ่มเติมจากข้อมูลอีกด้วย

#### 1.4 ขอบเขตของการวิจัย

ทำการศึกษาและพัฒนาอัลกอริธึมสำหรับสร้างและอ่านบาร์โค้ดสี 2 มิติ โดยพัฒนาบนระบบปฏิบัติการ windows โดยใช้กล้องเว็บแคมเป็นตัวจับภาพ โดยกำหนดให้บาร์โค้ดสีนั้นมีขนาดประมาณ 3 x 3 เซนติเมตรเพื่อให้มีขนาดเหมาะสมใช้งานได้ทั่วไป

ทำการทดลองประสิทธิภาพของอัลกอริธึมที่พัฒนาขึ้นภายในสภาวะแวดล้อมต่าง ๆ และทดสอบกับการบีบอัดภาพก่อนการพิมพ์เพื่อตรวจสอบความทนทานของอัลกอริธึม

#### 1.5 ประโยชน์หรือผลที่คาดว่าจะได้รับ

##### 1.5.1 ประโยชน์ที่คาดว่าจะได้รับ

- ได้อัลกอริธึมสำหรับบาร์โค้ดสี 2 มิติ ที่สามารถใช้งานร่วมกับกล้องเว็บแคมทั่วไปตามท้องตลาด
- ส่งเสริมการพัฒนางานวิจัยทางด้านคอมพิวเตอร์และเทคโนโลยีสารสนเทศ
- เป็นการพัฒนาศักยภาพนักศึกษาที่เป็นผู้ช่วยในการวิจัย

##### 1.5.2 หน่วยงานที่นำผลการวิจัยไปใช้ประโยชน์

- ใช้ในการเรียนการสอน/โครงการฝึกอบรม/งานฟาร์ม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง วิทยาเขตชุมพร
- หน่วยงานภาครัฐ เอกชนและผู้สนใจทั่วไป

#### 1.6 แผนการถ่ายทอดเทคโนโลยีหรือผลการวิจัยสู่กลุ่มเป้าหมาย

เมื่อเสร็จสิ้นการวิจัยแล้ว จะทำการถ่ายทอดผลงานให้กับกลุ่มเป้าหมายต่างๆ เช่น นักศึกษา ในวิทยาเขตชุมพร และหน่วยงานหรือบุคคลภายนอกใน การเผยแพร่ในรูปแบบรายงานการวิจัย และการตีพิมพ์ในรูปแบบการประชุมวิชาการ การบูรณาการผลงานวิจัยต่อการเรียนการสอนและการถ่ายทอดวิชาการแก่สังคม

## บทที่ 2

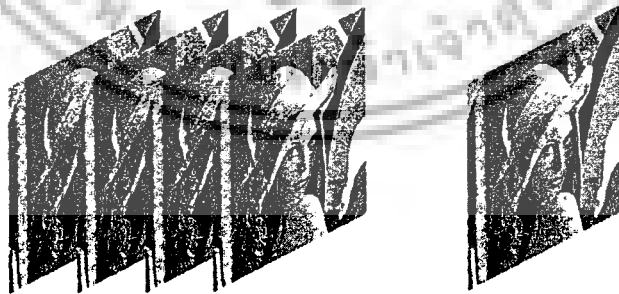
# ทฤษฎีและหลักการที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึง หลักการและทฤษฎีที่เกี่ยวข้องในการจัดทำโครงการ ซึ่งประกอบไปด้วย องค์ประกอบของภาพถ่าย หลักการของการแยกสี การตรวจสอบความเปลี่ยนแปลงของภาพ รูปแบบสีแบบ ระดับสีเทา จุดภาพหรือพิกเซล ความละเอียดในการแสดงผล

### 2.1 การประมวลผลภาพ (Image Processing)

การประมวลผลภาพ (Image processing) คือเป็นการประยุกต์ใช้งานการประมวลผลสัญญาณบน สัญญาณ 2 มิติเช่นภาพนิ่ง (ภาพถ่าย) หรือภาพวิดีโอ (วีดีโอ) และยังรวมถึงสัญญาณ 2 มิติอื่นๆที่ไม่ใช่ภาพ แนวความคิดและเทคนิคในการประมวลผลสัญญาณสำหรับสัญญาณ 1 มิตินั้นสามารถปรับมาใช้กับภาพได้ไม่ยาก แต่นอกเหนือจากเทคนิคจากการประมวลผลสัญญาณแล้วการประมวลผลภาพก็มีเทคนิคและแนวความคิดที่ เฉพาะซึ่งจะมีความหมายกับสัญญาณ 2 มิติเท่านั้นแต่อย่างไรก็ตามเทคนิคบางอย่างจากการประมวลผลสัญญาณ ใน 1 มิติจะค่อนข้างซับซ้อนเมื่อนำมาใช้กับ 2 มิติ

เมื่อหลายสิบปีมาแล้วการประมวลผลภาพนั้นจะอยู่ในรูปของการประมวลผลสัญญาณอนาล็อก (Analog) โดยใช้อุปกรณ์ปรับแต่งแสง (Optics) ซึ่งวิธีเหล่านั้นก็ไม่ได้หายสาบสูญหรือเลิกใช้ไปยังมีใช้เป็นส่วน สำคัญสำหรับการประยุกต์ใช้งานบางอย่างเช่น ฮอโลกราฟี (Holography) แต่เนื่องจากอุปกรณ์คอมพิวเตอร์ใน ปัจจุบันราคาถูกลงและเร็วขึ้นมากการประมวลผลภาพดิจิทัล (Digital image processing) จึงได้รับความนิยม มากกว่าเพราะการประมวลผลที่ได้ซับซ้อนขึ้นแม่นยำและง่ายในการลงมือปฏิบัติการประมวลผลภาพดิจิทัล (Digital image processing) เป็นสาขาที่กล่าวถึงเทคนิคและอัลกอริทึมต่างๆที่ใช้การประมวลผลภาพที่อยู่ใน รูปแบบดิจิทัล (ภาพดิจิทัล) ภาพในที่นี้รวมความหมายถึงสัญญาณดิจิทัลใน 2 มิติอื่นๆ โดยทั่วไปคำนี้เมื่อใช้ อย่างกว้างๆจะครอบคลุมถึงสัญญาณวิดีโอ (Video) หรือภาพเคลื่อนไหวซึ่งจะเป็นชุดของภาพนิ่งเรียกว่าเฟรม (Frame) หลายๆภาพต่อกันไปตามเวลา



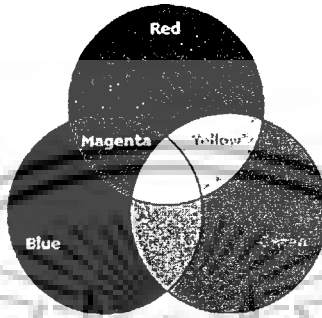
ภาพที่ 2.1 ความสัมพันธ์ของสัญญาณภาพเทียบกับเวลา

ซึ่งก็คือสัญญาณ 3 มิติเมื่อนับเวลาเป็นมิติที่ 3 หรืออาจจะครอบคลุมถึงสัญญาณ 3 มิติอื่นๆเช่นภาพ 3 มิติทางการแพทย์หรืออาจจะมากกว่านั้นเช่นภาพ 3 มิติและหลายชนิด (Multimodal image)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 รูปแบบสีแบบอาร์จีบี

เป็นระบบสีพื้นฐานของคอมพิวเตอร์ที่ใช้ในการแสดงผลโดยจุดย่อยของภาพ (Pixel) จะประกอบด้วยค่าสี 3 ค่าคือแดง (R) เขียว (G) และน้ำเงิน (B) การผสมสีทั้งสามนี้ด้วยค่าต่างๆกันจะก่อให้เกิดสีที่แตกต่างกันโดยคอมพิวเตอร์จะเก็บค่าสีนี้แยกกันโดยใช้ขนาดข้อมูล 1 ไบต์ต่อ 1 สีทำให้ค่าของสีนั้นมีได้ 256 ระดับและผสมได้สีทั้งหมด 16 ล้านสีสามารถแสดงโมเดลของสีได้ดังภาพที่ 2.2



ภาพที่ 2.2 รูปแบบของสีแบบ RGB

## 2.3 รูปแบบสีแบบระดับสีเทา

ภาพระดับสีเทาหรือ Gray Scale เป็นภาพที่ประกอบไปด้วยค่าของพิกเซลที่เป็นค่าเฉดสีของสีเทาซึ่งได้จากระดับสีจากการเปลี่ยนสีจากสีดำไปเป็นสีขาวโดยจะถือสีขาวเป็นสีที่มีแก่ที่สุดและสีดำเป็นสีที่อ่อนที่สุด เนื่องจากค่าของพิกเซลที่เป็นสีขาวจะมีค่ามากกว่าค่าของสีดำสิ่งที่ภาพระดับสีเทาแตกต่างกับภาพขาว-ดำคือในภาพขาว-ดำจะเก็บข้อมูลโดยมีอยู่สองสีคือสีขาวและสีดำแต่ในภาพระดับสีเทาจะมีความละเอียดมากกว่ากล่าวคือจะเก็บค่าของเฉดสีเทาที่อยู่ระหว่างการเปลี่ยนจากสีดำเป็นสีขาวการนำตัวเลขมาแทนค่าของพิกเซลในภาพระดับสีเทานั้นใช้เป็นเปอร์เซ็นต์คือ 0%(สีดำ) ถึง 100%(สีขาว) แต่เมื่อนำมาใช้งานร่วมกับเครื่องพิมพ์จะมีการทำงานที่ตรงกันข้ามกล่าวคือจะต้องทำการกลับค่าของเปอร์เซ็นต์จึงจะพิมพ์ได้ตรงตามที่ต้องการเนื่องจากการอ่านค่าของเครื่องพิมพ์คือ 0% คือไม่ปล่อยหมึกออกมาซึ่งหมายความว่าส่วนนั้นจะเป็นสีขาวส่วน 100%เครื่องพิมพ์จะปล่อยหมึกออกมาซึ่งนั่นคือสีดำในทางคอมพิวเตอร์เดิมจะมีการเก็บในลักษณะของเลขฐานสองจำนวนสี่บิตสามารถเก็บความแตกต่างได้ 16 ระดับแต่ในปัจจุบันนี้ได้มีการเพิ่มข้อมูลที่เก็บโดยเพิ่มเป็นเลขฐานสองจำนวน 8 บิต (0-255) ซึ่งจะสามารถเก็บค่าความแตกต่างได้



ภาพที่ 2.3 รูปแบบสีแบบระดับสีเทา

1. การแปลงภาพแบบอาร์จีบีไปสู่ภาพแบบระดับสีเทา ในการทำงานของไลบรารีโอเพนซีวีเมื่อต้องการจะทำการเปลี่ยนรูปแบบสีของภาพจาก RGB ไปสู่ภาพระดับสีเทาจะมีการเรียกการทำงานของฟังก์ชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CvtColor โดยการทำงานในฟังก์ชันนี้จะอ่านค่าที่ละพิกเซลแล้วทำการคำนวณเพื่อให้ได้ค่าใหม่ที่อยู่ในช่วง 0-255 ซึ่งการคำนวณจะเป็นดังสมการ  $Y=0.299*R + 0.587*G + 0.114*B$  โดยที่ Y คือค่าผลลัพธ์ในช่วง 0-255 ตัวแปร R, G และ B คือค่าของสีแดงสีเขียวและสีน้ำเงินตามลำดับ

2. เทรดโฮวด์ (Threshold) เป็นการแปลงภาพระบบขาวดำ (Grayscale) ซึ่งมีค่าอยู่ระหว่าง 0-255 ให้เป็นภาพที่มีค่าเพียงสองระดับ (Binary Image) โดยมีเงื่อนไขว่าถ้าความเข้มแสงของจุดภาพใดมีค่าต่ำกว่าหรือเท่ากับค่าเทรดโฮวด์ให้จุดภาพนั้นมีค่าเป็น 0 หรือเป็นสีดำและจุดภาพใดที่มีค่าสูงกว่าค่าเทรดโฮวด์ให้จุดภาพนั้นมีค่าเป็น 1 หรือสีขาว

## 2.4 การตรวจสอบความเปลี่ยนแปลงของภาพ (Image differencing)

การตรวจสอบความเปลี่ยนแปลงของภาพเป็นเทคนิคการประมวลผลภาพใช้ในการพิจารณาการเปลี่ยนแปลงระหว่างภาพความแตกต่างระหว่างสองภาพนั้นคำนวณจากการหาความแตกต่างระหว่างพิกเซล (Pixel) และสร้างภาพความแตกต่างจากผลลัพธ์นั้นสำหรับการทำงานของเทคนิคนี้ภาพทั้งสองภาพจะต้องวางตัวในแนวเดียวกันเพื่อให้จุดตรงกันและค่าโฟโตเมตริก(Photometric) ต้องทำให้สอดคล้องกันทั้งการคาลิเบรท (Calibration) หรือการทำโพสโพรเซสซิง(Post-processing) ซึ่งความซับซ้อนของการทำการทำโพสโพรเซสซิง (Post-processing) จำเป็นต้องทำก่อนการทำอิมเมจดิฟเฟอเรนซ์ซึ่งขึ้นกับประเภทของรูปภาพ



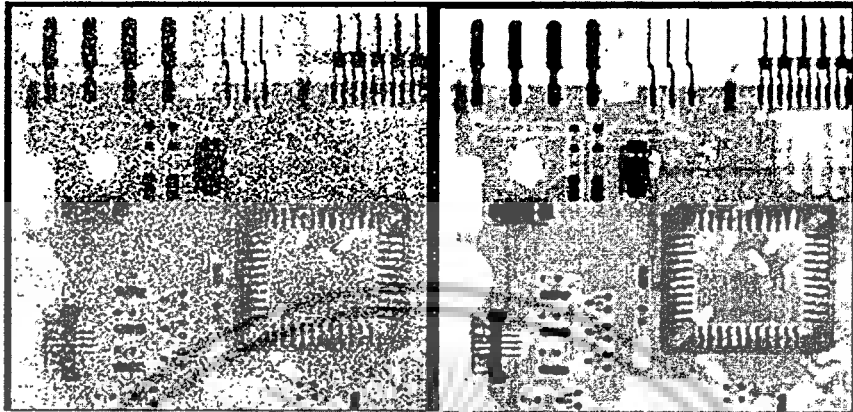
ภาพที่ 2.4 ตัวอย่างการทำการตรวจสอบความเปลี่ยนแปลงของภาพ

การกรองข้อมูลโดยการหาค่ากลาง (Median filter)

เป็นเทคนิคของ Order-statistics filter ที่เป็นที่รู้จักกันดีที่สุดเทคนิคนี้จะทำการแทนที่ค่าของพิกเซลนั้นด้วยค่ากลาง (Median) ของค่าความเข้มแสงของพิกเซลที่อยู่ในพื้นที่ภายในเทมเพลตตัวกรองนั้นซึ่งช่วยลดผลกระทบของตัวรบกวน (Noise) โดยที่ทำให้ภาพที่ได้มีความคมชัดมากกว่าเทคนิคการทำการกรองข้อมูลโดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหาค่ากลางที่มีขนาดของเทมเพลตตัวกรองขนาดเดียวกันโดยเฉพาะในกรณีของผลกระทบจาก Impulse noise หรือ Salt-and-pepper noise



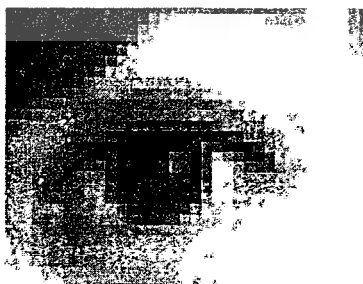
ภาพที่ 2.5 เปรียบเทียบภาพที่มี Salt-and-pepper noise กับภาพที่ใช้ใช้ Median filter

## 2.5 จุดภาพหรือพิกเซล(pixel)

จุดภาพหรือพิกเซล (pixel) เป็นหน่วยพื้นฐานของภาพ คือจุดภาพบนจอแสดงผล หรือ จุดภาพในรูปภาพที่รวมกันเป็นภาพขึ้น โดยภาพหนึ่งๆ จะประกอบไปด้วยจุดภาพหรือพิกเซลมากมาย และแต่ละภาพที่สร้างขึ้นจะมีความหนาแน่นของจุดภาพ หรือบางครั้งแทนที่ความละเอียด (ความคมชัด) ที่แตกต่างกันไป จึงใช้ในการบอกคุณสมบัติของภาพ จอภาพ หรือ อุปกรณ์แสดงผลภาพได้

จอภาพที่มีจำนวนพิกเซลมาก จะมีความละเอียดของภาพมาก โดยมากจะระบุจำนวนพิกเซลแนวนอน x แนวตั้ง เช่น 1366x768 พิกเซล คำว่า "พิกเซล" (pixel) มาจากคำว่า "พิกเจอร์" (picture) ที่แปลว่า รูปภาพ และ "อีเลเมนต์" (element) ที่แปลว่า องค์ประกอบ

นอกเหนือจากหน่วยของการแสดงที่เป็นจุดภาพหรือพิกเซล (Pixel) ยังมีคำว่า bpp ย่อมาจากคำว่า bits per pixel คือค่าของจำนวนสีที่สามารถแสดงได้ในหนึ่งพิกเซล ดังที่บอกไปในข้างต้นว่า 1 Pixel จะสามารถแสดงผลแสดงผลได้เพียงสีเดียวเท่านั้น ดังนั้นค่า bpp จะเป็นตัวบอกว่าใน 1 Pixel จะสามารถเปลี่ยนสีได้กี่สี (รูปคงไม่สวยแน่ถ้ามี Pixel หลายอันมาเรียงต่อกันแล้วทุกพิกเซลเป็นสีเดียวกัน) อาจเรียกว่า บิต (Bit ) เฉยๆก็ได้



ภาพที่ 2.6 ขนาดของพิกเซล (pixel)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.6 ความละเอียดของรูปภาพ และความละเอียดในการแสดงผล

1. ความละเอียดในการแสดงผล ความละเอียดในการแสดงผล ( Resolution ) คำนี้สามารถใช้ได้กับสถานการณ์ที่แตกต่างกัน เช่น ความละเอียดของการแสดงผลของเครื่องพิมพ์ หรือความละเอียดในการแสดงผลของจอภาพ ดังนั้นความละเอียดในการแสดงผลจึงหมายถึง จำนวนหน่วยต่อพื้นที่

2. ความละเอียดของรูปภาพ จำนวนจุดภาพที่ใช้ประกอบกันเป็นภาพหรือความละเอียดจากการสแกนภาพ การแสดงภาพได้ละเอียดมากเท่าใดนั้นขึ้นอยู่กับประเภทของจอภาพ VGA จะแสดงภาพได้ละเอียดน้อยกว่า SVGA ความละเอียดของภาพสามารถบอกเป็นตัวเลขสองจำนวน เช่น ความละเอียดของภาพขนาด 1024x768 ซึ่งเมื่อคำนวณออกมาแล้วก็คือจำนวนจุดที่จอภาพสามารถสร้างออกมาได้ ในกรณีนี้เลขจำนวนแรกคือจำนวนจุดในแนวนอนซึ่งเท่ากับ 1024 จุด ตัวเลขจำนวนที่สองคือจำนวนจุดในแนวตั้ง ซึ่งเท่ากับ 768 จุด



ภาพที่ 2.7 ความละเอียดของภาพที่แตกต่างกัน

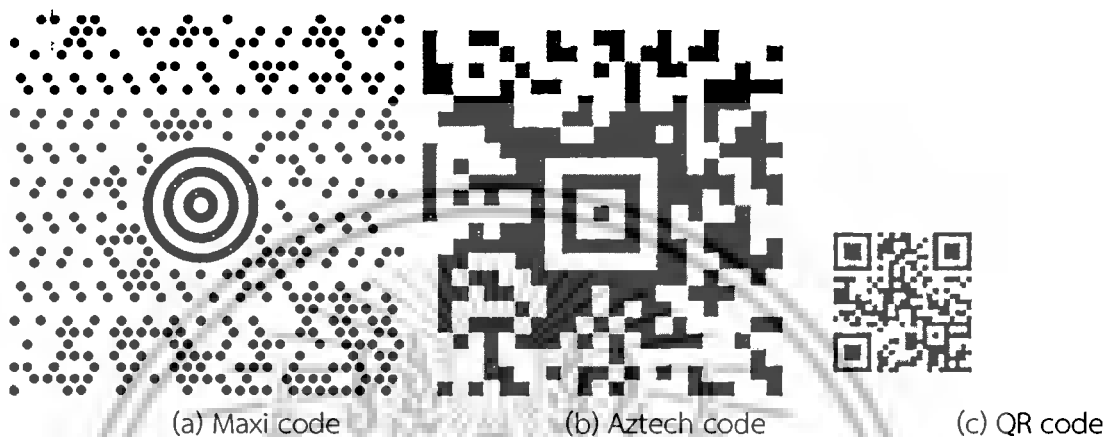
ขนาดแสดงภาพมาตรฐานมาดังนี้

VGA 0.3	ล้านพิกเซล= 640x480
SVGA 0.5	ล้านพิกเซล= 800x600
XGA 0.8	ล้านพิกเซล= 1024x768 (หรืออาจเรียก XVGA)
SXGA 1.3	ล้านพิกเซล= 1280x1024
EXGA 1.4	ล้านพิกเซล= 1400x1050
UXGA 1.9	ล้านพิกเซล= 1600x1200
QXGA 3.1	ล้านพิกเซล= 2048x1536
QSXGA 5.2	ล้านพิกเซล= 2560x2048
WQSXGA 6.6	ล้านพิกเซล= 3200x2048
QUXGA 7.7	ล้านพิกเซล= 3200x2400
WQUXGA 9.2	ล้านพิกเซล= 3840x2400

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.7 ทบทวนวรรณกรรมที่เกี่ยวข้อง

ข้อมูลที่คอมพิวเตอร์อ่านได้ (Machine readable data) นั้นมีการพัฒนามานานแล้วเช่น บาร์โค้ด แถบขาว-ดำที่พบทั่วไปตามบรรจภัณฑ์สินค้า และในยุคโลกาภิวัตน์นี้บาร์โค้ดสองมิติชนิดขาวดำได้นำมาใช้เพิ่มมากขึ้นดังตัวอย่างเช่น ดังภาพที่1



ภาพที่ 2.8 บาร์โค้ด 2 มิติชนิดต่าง ๆ

จากรูป Maxi code มีข้อดีในการตรวจสอบในสภาพที่หมุนต่างๆ ได้ดี และสามารถกึ่งข้อมูลที่ผิดพลาดได้ดี แต่มีข้อเสียคือ ความจุข้อมูลน้อย ส่วน Aztech และ QR นั้นมีลักษณะคล้ายกัน โดยเฉพาะ QR Code ที่ค่อนข้างแพร่หลายในเมืองไทย โดยมีกำเนิดมาจากประเทศญี่ปุ่น สามารถบรรจุข้อมูลเช่น URL หรือ เบอร์โทรศัพท์ ตลอดจนข้อมูลที่อยู่ โดยขึ้นกับขนาดของบาร์โค้ด ซึ่งอาจจุได้ถึง 2,500 ตัวอักษร แต่ต้องใช้ขนาดใหญ่มากซึ่งกินเนื้อที่ในสื่อ หรือหน้ากระดาษ

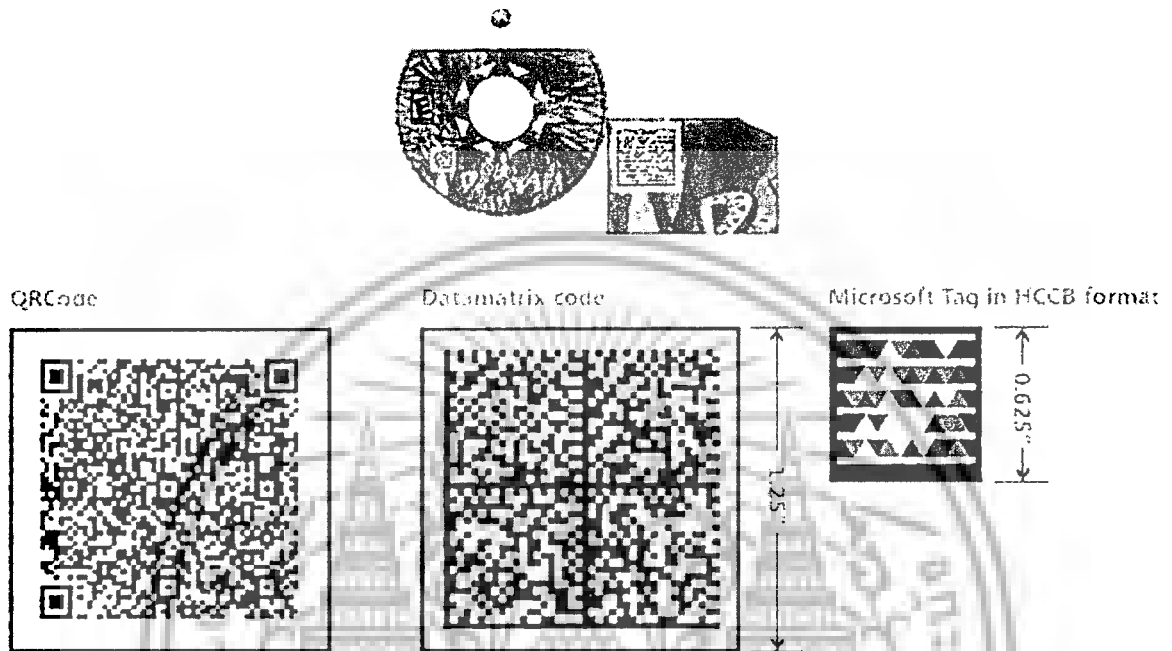
สำหรับ QR Code นั้นได้มีผู้ใช้ทำการปรับแต่งเองให้มีความสวยงามโดยใช้สีโทนเข้มแทนสีดำ ซึ่งทำให้สวยงามขึ้นแต่ไม่มีผลต่อความจุแต่อย่างใด ดังภาพที่ 2.9



ภาพที่ 2.9 บาร์โค้ดสี เพื่อความสวยงาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

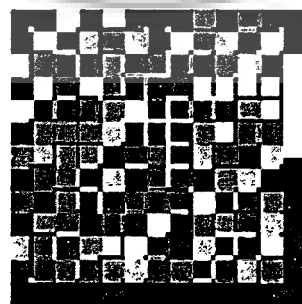
ในยุคของบาร์โค้ดสีนั้นได้มีการพัฒนาบ้างแล้ว เช่นแถบสีสามเหลี่ยมบนปกซีดี หรือแท็กของไมโครซอฟท์ ในภาพที่ 2.10 ขวามือล่าง ที่ใช้รูปสามเหลี่ยมเพื่อเพิ่มความจุข้อมูล ซึ่งเมื่อเทียบกับบาร์โค้ดขาวดำทางซ้ายมือ จะเห็นได้ว่าเก็บข้อมูลได้เท่ากับหรือมากกว่าในขนาดที่เล็กมาก



ภาพที่ 2.10 บาร์โค้ดสี และการเทียบขนาด

แต่แท็กของไมโครซอฟท์นั้นจะเก็บ link ไปยังข้อมูลที่แท้จริง แทนการเก็บข้อมูลจริง ซึ่งต่างจากชนิดอื่น ๆ ทำให้มีข้อจำกัดในการใช้งาน

นอกจากนี้มีการพัฒนาบาร์โค้ดสีในรูปของเมตริกซ์สีดังภาพที่ 2.11 แต่จากการสืบค้นพบข้อจำกัดของสีที่ใช้ เพียงจำนวน 8 สีเท่านั้น ทางคณะวิจัยจึงมีแนวคิดที่จะหาวิธีในการเพิ่มจำนวนสีขึ้น โดยยังคงรักษาประสิทธิภาพในการแยกชุด หรืออ่านข้อมูลในระดับที่ยอมรับได้ โดยมุ่งเน้นไปที่การประยุกต์ใช้งานในอุปกรณ์ไร้สาย เช่นมือถือ และกล้องเว็บแคม



ภาพที่ 2.11 บาร์โค้ดสีแบบเมตริกซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

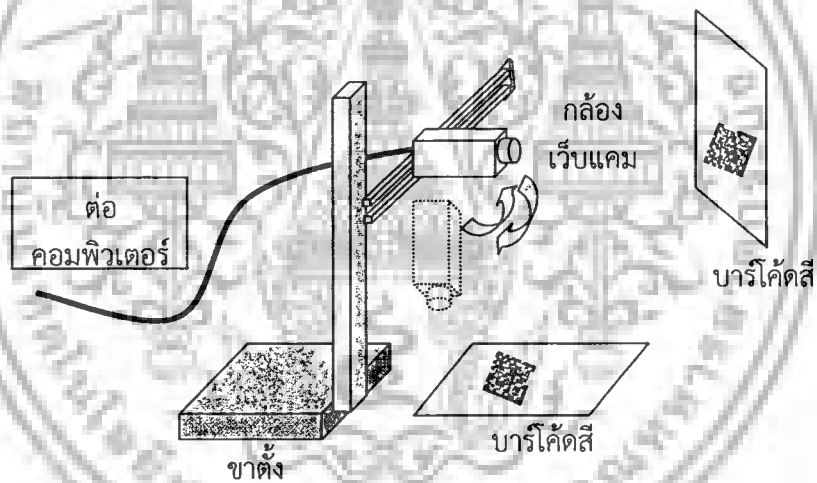
### บทที่ 3

## การออกแบบและพัฒนาอัลกอริธึม

ในบทนี้จะกล่าวถึงการออกแบบและพัฒนาอัลกอริธึมสำหรับบาร์โค้ดสี 2 มิติตามขอบเขตที่ได้กล่าวมาในบทแรก โดยในระบบที่นำเสนอนี้สามารถแบ่งการทำงานออกเป็น 2 ส่วนหลักคือ การออกแบบสร้างรูปภาพบาร์โค้ดสี และการพัฒนาอัลกอริธึมในการอ่านบาร์โค้ด ซึ่งทั้ง 2 ส่วนจะต้องมีการออกแบบให้สอดคล้องและจำเป็นต้องคำนึงให้อีกส่วนหนึ่งนั้นสามารถทำงานได้อย่างมีประสิทธิภาพ

#### 3.1 โครงสร้างของระบบ

ในงานวิจัยนี้ได้ทำการออกแบบระบบในการอ่านบาร์โค้ดดังแสดงในภาพที่ 3.1 ประกอบไปด้วยโครงสร้างขาตั้งสำหรับติดตั้งกล้องเว็บแคมโดยสามารถปรับระยะและแนวในการถ่ายภาพทั้งแบบภาพฉายด้านหน้าและภาพฉายด้านบน โดยกล้องเว็บแคมที่ใช้เป็นกล้องที่เชื่อมต่อทางช่องเสียบยูเอสบี (USB)



ภาพที่ 3.1 โครงสร้างของระบบ

#### 3.2 การออกแบบบาร์โค้ดสี

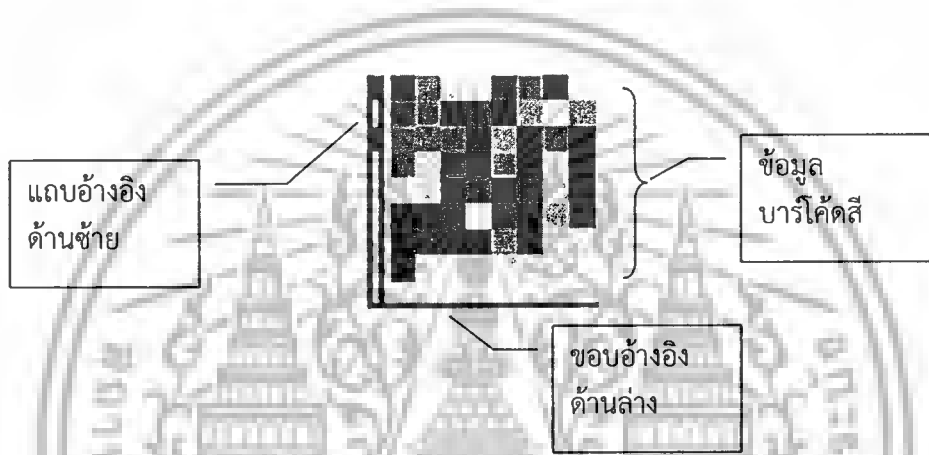
ลักษณะของบาร์โค้ดที่ได้ออกแบบนั้นจะมีลักษณะดังแสดงในภาพที่ 3.2 โดยพื้นที่ส่วนใหญ่จะเป็นพื้นที่ของข้อมูล และมีแถวสีดำด้านซ้ายและด้านล่างเป็นขอบอ้างอิง สำหรับแถวอ้างอิงนั้นมีไว้สำหรับอ้างอิงตำแหน่งของบาร์โค้ดสีและใช้สำหรับการปรับปรุงภาพถ่ายในการหมุนภาพ การแก้การเอียง หรือแก้การถ่ายภาพที่ไม่ได้ระนาบ

สำหรับแถบด้านอิงด้านซ้ายจะมีลักษณะเป็นแถบสีดำ 2 แถบที่มีแถบสีขาวอยู่ระหว่างกลาง 1 แถบ โดยความกว้างของทั้ง 3 แถบมีขนาดเท่ากัน ยกเว้นบริเวณด้านบนของแถบอ้างอิง โดยด้านบนของแถบอ้างอิงจะ

มีลักษณะพิเศษ โดยมีลักษณะแถบจากซ้ายไปขวาและจากบนลงล่าง โดยให้มีความสูงเท่ากับความสูงของ บาร์โค้ด ข้อมูล 1 บล็อก เป็นดังนี้

- แถวที่ 1 สีดำ สีดำ สีดำ
- แถวที่ 2 สีดำ สีขาว สีดำ
- แถวที่ 3 สีดำ สีดำ สีดำ
- แถวที่ 4 เป็นต้นไป สีดำ สีขาว สีดำ

ทั้งนี้ก็เพื่อให้สามารถหาความสูงของบล็อกข้อมูลในขั้นตอนการอ่านได้ โดยอาจดูจากความสูงของ แถวที่ 1 หรือ แถวที่ 2 หรือ แถวที่ 3 หรือใช้หาค่าเฉลี่ยของทั้ง 3 แถวเป็นต้น



ภาพที่ 3.2 ส่วนประกอบบาร์โค้ดสี

สำหรับบล็อกข้อมูลบาร์โค้ดสีนั้นจะมีการจัดเรียงลำดับจากซ้ายไปขวาก่อนแล้วตามด้วยจากบนลงล่าง ยกตัวอย่างเช่น จากภาพที่ 3.2 บล็อกสีด้ามุมซ้ายมือบนจะมีลำดับอยู่ที่ 1 ส่วนบล็อกสีขาวแถวบนจะมีลำดับอยู่ที่ 4 ส่วนบล็อกในแถวที่ 2 จะเริ่มต้นจากซ้ายมือเป็นลำดับที่ 9 เป็นต้น

หากเรานำบล็อกข้อมูลแต่ละบล็อกมาเรียงกันเป็นแถวเดียวกัน จะได้แถวที่ยาว 1 แถว ซึ่งส่วนประกอบของข้อมูลใน 1 แถวนั้นจะประกอบไปด้วยทั้ง Header และข้อมูลจริง Data ดังแสดงในภาพที่ 3.3 มีรายละเอียด คือแต่ละ 1 บล็อกแทนข้อมูล 4 บิต ดังนี้

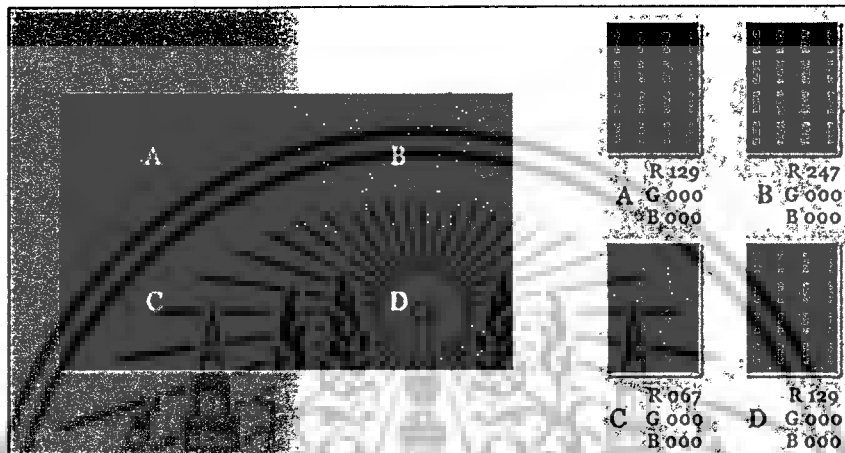
- กลุ่มที่ 1 เป็นบล็อกของสีอ้างอิงที่ใช้ในอัลกอริธึม จำนวน 16 บล็อก
- กลุ่มที่ 2 เป็นชนิดของบาร์โค้ด หรืออาจใช้สำหรับระบุรุ่น จำนวน 1 บล็อก
- กลุ่มที่ 3 เป็นจำนวนตัวอักษรของข้อมูล แบบ Unsigned Char 8 bit จำนวน 2 บล็อก
- กลุ่มที่ 4 เป็นกลุ่มข้อมูล มีจำนวนเท่ากับ 2 เท่าของจำนวนตัวอักษร

16 Color Ref.	T	Size	Data
---------------	---	------	------

ภาพที่ 3.3 การเรียงลำดับข้อมูลในบาร์โค้ดสี

### 3.3 การเลือกใช้สีในบาร์โค้ดสี

ในงานวิจัยนี้ได้เลือกใช้โมเดลสี RGB เนื่องจากเป็นโมเดลพื้นฐานไม่จำเป็นต้องมีการคำนวณเพิ่มเติม แต่กระนั้นก็มีผลต่อประสิทธิภาพของอัลกอริธึมที่นำเสนอ โดยจากผลการศึกษา Briggs 2007 พบว่าความเข้มแสงมีผลกระทบต่อค่าของสีอย่างมาก ดังแสดงในภาพที่ 3.4 จะเห็นได้ว่าเมื่อมีเงามาบดบังจะทำให้ค่าสีที่อ่านได้จากกล้องนั้นเปลี่ยนแปลงไปจนไม่เห็นความแตกต่างของสี เช่น พื้นที่ A และ D ในภาพ



ภาพที่ 3.4 ผลของแสงต่อค่าสี

ปัญหาอีกประการที่สำคัญคือบาร์โค้ดสีที่ถูกสังเคราะห์ขึ้นในคอมพิวเตอร์จะต้องถูกพิมพ์ลงบนกระดาษ จากนั้นในการอ่านค่าก็จะต้องอ่านด้วยกล้องหรือเครื่องสแกนเพื่อแปลงเป็นข้อมูลดิจิทัล เพื่อนำไปประมวลผลต่อไป ซึ่งขั้นตอนที่กล้องอ่านค่าสีนั้นจะเกิดการสูญเสีย หรือมีสัญญาณรบกวน ทำให้ค่าที่สีอ่านได้ กับค่าสีที่สังเคราะห์ขึ้นนั้นมีความแตกต่างกัน

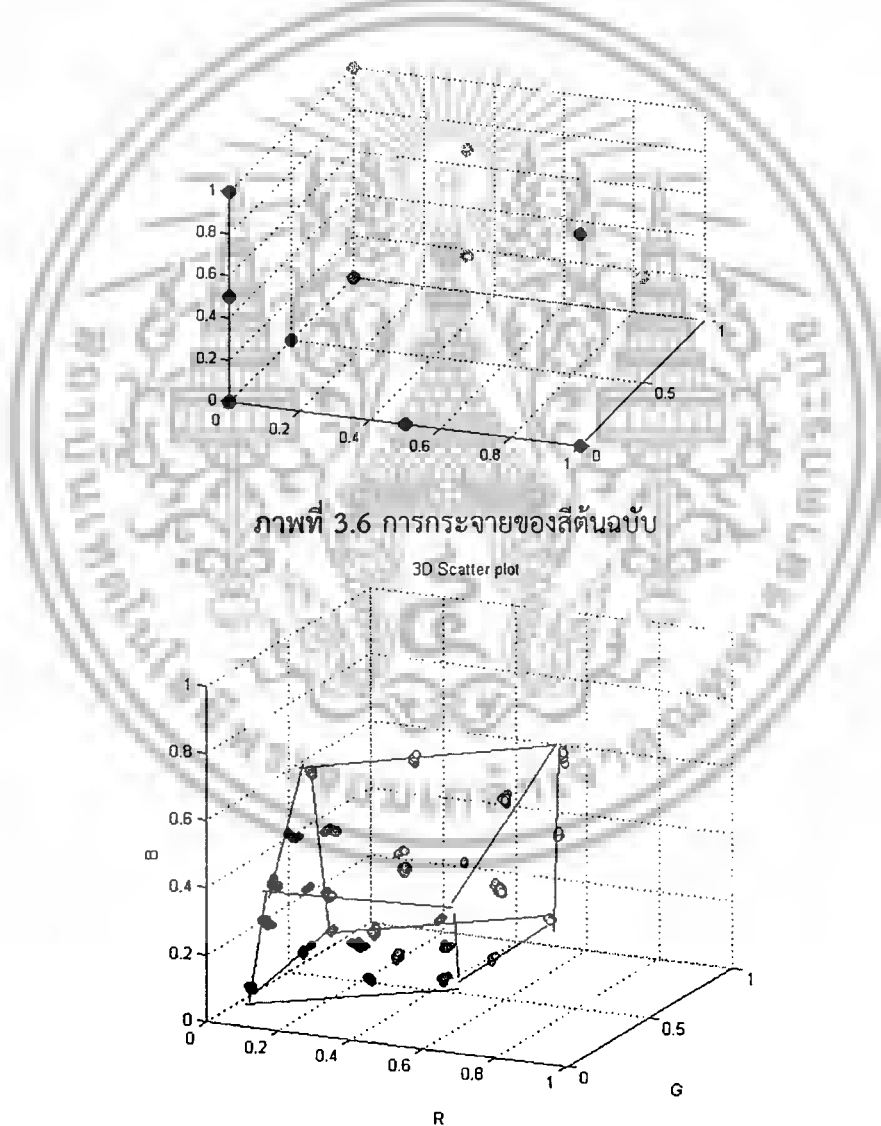
ในงานวิจัยนี้ได้ทำการทดลองหาผลของกระบวนการพิมพ์และถ่ายภาพ (Print and Capture) เพื่อที่จะคัดเลือกสีที่เหมาะสมสำหรับการใช้งานในอัลกอริธึมที่นำเสนอนี้ โดยจากโมเดลสี RGB มีลักษณะเป็นลูกบาศก์ โดยที่มุมทั้ง 8 มุมของลูกบาศก์ จะประกอบไปด้วย สีดังต่อไปนี้ ดำ ขาว เขียว น้ำเงิน แดง ฟ้า ม่วง และส้ม

ในการทดลองเราแบ่งระดับความเข้มแสงในแต่ละแม่แสง ออกเป็น 3 ระดับ คือ 0, 0.5 และ 1.0 ทำให้เราได้กลุ่มสีที่จะทำงานทดสอบในขั้นต้น 27 สี แต่เนื่องจากในแนวทแยงมุมจากสีดำไปหาสีขาวเดิมมี 3 จุด คือ (0,0,0) (0.5,0.5,0.5) และ (1, 1, 1) นำจะมีความททานต่อการเปลี่ยนแปลงเราจึงเพิ่มอีก 2 สี ระหว่างจุด 3 ที่กล่าวมา รวมเป็นทั้งหมด 29 จุด ดังภาพที่ 3.5 ทำการพิมพ์ลงบนกระดาษและถ่ายด้วยกล้องเว็บแคม จากนั้นอ่านค่าสีจากภายในบล็อกสีขนาด 5x5 พิกเซลแล้วนำมาหาค่าเฉลี่ยในแต่ละสีจากนั้นนำค่ามาพล็อตลงกราฟจะมีลักษณะของกลุ่มสีต้นฉบับดังภาพที่ 3.6 และกลุ่มสีที่อ่านค่าจากกล้องดังภาพที่ 3.7

เมื่อเราทำการฉายภาพ 3.7 ลงไปยัง 3 ระนาบ คือ มุมมองด้านบน RG, RB มุมมองด้านหน้า และ มุมมองด้านข้าง GB ดังภาพที่ 3.8

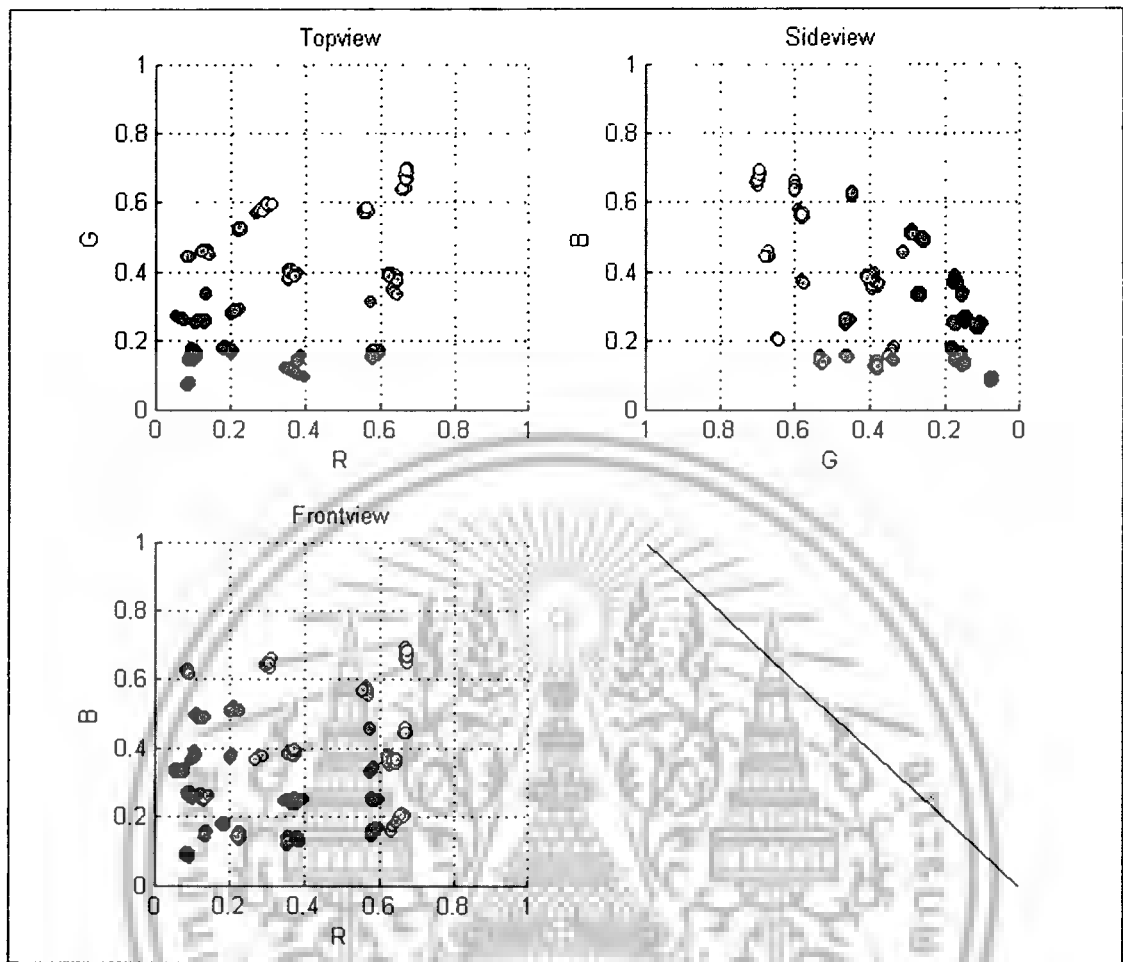


ภาพที่ 3.5 กลุ่มสีที่ใช้ทดสอบความทนทานต่อการพิมพ์และถ่ายภาพ



ภาพที่ 3.7 การกระจายของสีที่ถ่ายด้วยกล้องเว็บแคม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.8 ภาพฉายการกระจายตัวของสี

จากภาพฉายในระนาบ RG เราพบว่าสีแดงมีการกระจายตัวน้อยกว่ากลุ่มสีเขียว และกลุ่มสีในแนวทแยงมุมจากสีดำไปสีขาวก็มีการกระจายตัวน้อยเช่นกัน และจากภาพฉายในระนาบ GB นั้นกลุ่มสีเขียวและสีน้ำเงินนั้นมีการกระจายมาอยู่ใกล้ ๆ กัน และค่อนข้างกลมกลืนกัน ทำให้เป็นปัญหาในการแยกชัด

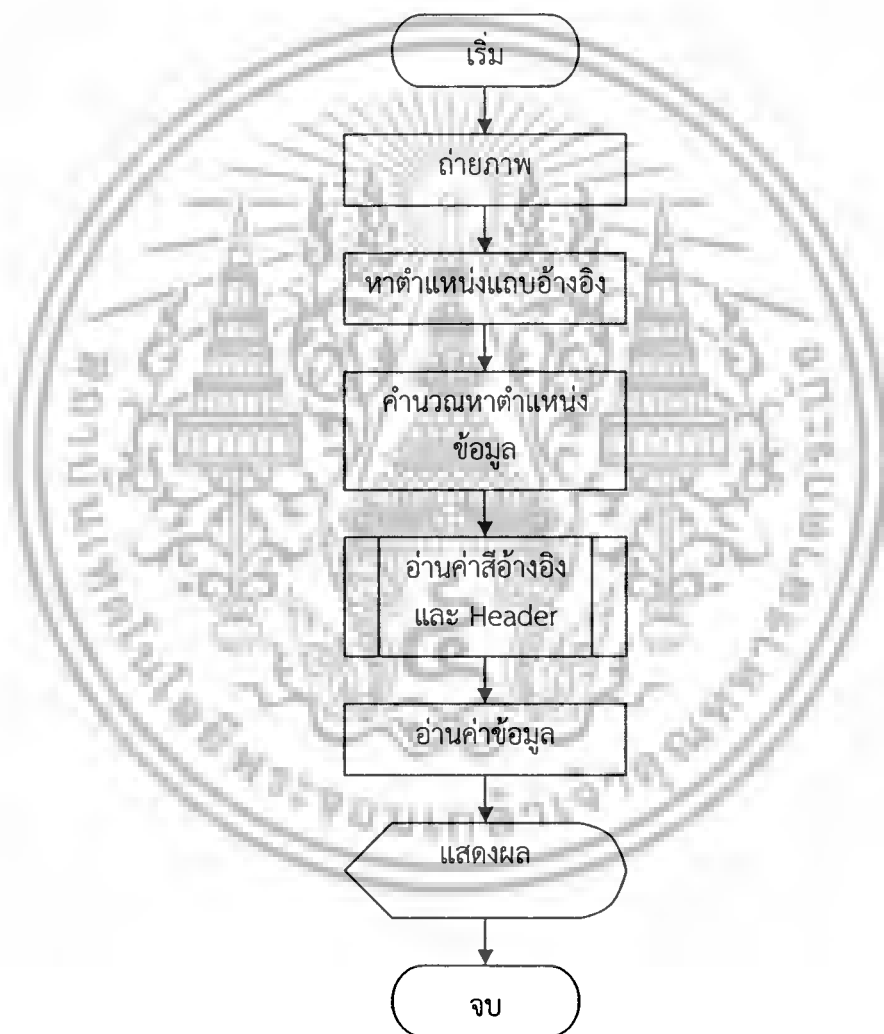
จากการทดลองดังกล่าวเราจึงคัดเลือกตัวแทนสีออกมาจำนวน 16 สีดังตารางที่ 3.1

ตารางที่ 3.1 ตัวแทนสีที่เลือก

กลุ่มของตัว	กลุ่มแม่แสง	กลุ่มแสงผสม	กลุ่มแสงมืด	กลุ่มแสงสว่าง
ขาว	แดง	เหลือง	แดงเข้ม	แดงอ่อน
เทาอ่อน	เขียว	ม่วง	เขียวเข้ม	เขียวอ่อน
เทาเข้ม	น้ำเงิน	ฟ้า	น้ำเงินเข้ม	น้ำเงินอ่อน
ดำ				

### 3.4 อัลกอริธึมในการอ่านบาร์โค้ดสี

ในการอ่านบาร์โค้ดนั้นมีขั้นตอนคร่าว ๆ ดังแสดงในภาพที่ 3.9 โดยเริ่มจากการถ่ายภาพโดยจะมีการกำหนดในการถ่ายให้อยู่ในแนวที่เหมาะสมเพื่อสะดวกในการประมวลผลซึ่งมีเส้นกรอบช่วยในการถ่าย จากนั้นภาพที่ถ่ายได้จะถูกนำไปหาตำแหน่งของแถบอ้างอิง เมื่อได้ตำแหน่งอ้างอิงแล้วทำการคำนวณหาจำนวนบล็อกสีทั้งหมดทั้งจำนวนแถวและจำนวนหลัก จากนั้นจะเป็นการทำซ้ำอ่านค่าสีอ้างอิงและ Header ตามที่ได้กำหนดไว้แล้วทำการอ่านบล็อกข้อมูลที่เหลือออกมา ทำการคำนวณหาสีอ้างอิงที่ใกล้เคียงที่สุดเพื่อถอดรหัสข้อมูล และแสดงผลข้อมูลที่ได้ออกทางหน้าต่างโปรแกรมเป็นการจบกระบวนการ สำหรับในรายละเอียดนั้นจะอธิบายเป็นหัวข้อย่อยที่สำคัญดังต่อไปนี้



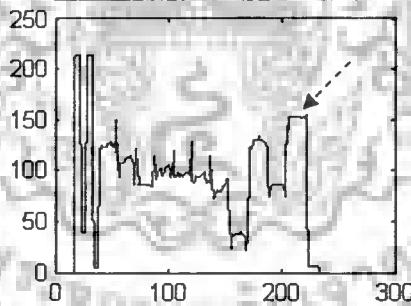
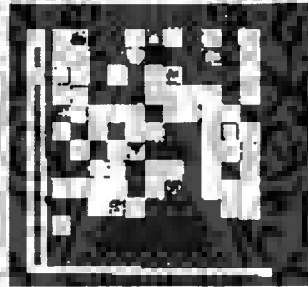
ภาพที่ 3.9 โฟลวชาร์ตของอัลกอริธึมในการอ่านบาร์โค้ด

#### 3.4.1 การหาตำแหน่งแถบอ้างอิง

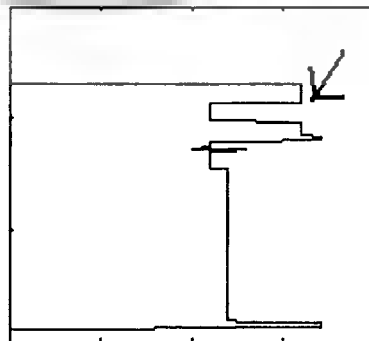
ในการหาตำแหน่งอ้างอิงนั้นจะกระทำภายใต้เงื่อนไขที่ว่าแถบอ้างอิงด้านซ้ายในแนวตั้งนั้นอยู่ในแนวตั้งไม่เอียง รายละเอียดในการหาตำแหน่งคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ทำการแปลงภาพเป็นภาพระดับเทา
2. คำนวณหาค่าเทรโซร์จากค่าที่น้อยที่สุดของขอบของภาพถ่าย 4 ด้าน
3. แปลงภาพระดับเทาเป็นภาพไบนารีที่มีค่ามากกว่าครึ่งหนึ่งของเทรโซร์ในข้อที่ 2 ดังภาพที่ 3.10 (บน)
4. ทำการฉายภาพลงบนแกนแนวนอน หรือทำการนับจำนวนพิกเซลที่อยู่ในหลักเดียวกันนั่นเอง
5. จากภาพถ่ายในข้อที่ 4 มีลักษณะดังภาพที่ 3.10(ล่าง) สามารถหาความกว้างของแถบอ้างอิงจากระยะห่างระหว่าง ยอด 2 ยอดที่มีค่าการฉายสูงที่สุด
6. ตำแหน่งหลักสุดท้ายของภาพที่เป็นบล็อกสี่ หาได้จาก ภาพที่ 3.10 (ล่าง) ดูจากขวาไปซ้ายหาตำแหน่งหลักแรกที่มีค่าการฉายภาพมากกว่า ความกว้างของแถบอ้างอิง (ลูกศรปะ)
7. ทำนองเดียวกันสามารถหาความกว้างของบล็อกสี่จากความกว้างของแถบแรกบนภาพถ่ายบนแกนแนวตั้งดังภาพ 3.11 (ลูกศรชี้)



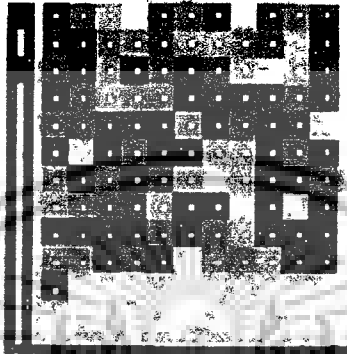
ภาพที่ 3.10 ภาพไบนารีและภาพฉายลงบนแกนแนวนอน



ภาพที่ 3.11 ภาพไบนารีเฉพาะส่วนแถบอ้างอิงแนวตั้งและภาพฉายลงบนแกนแนวตั้ง

### 3.4.2 การคำนวณหาตำแหน่งของข้อมูล

จากที่กล่าวมาแล้ว ทำให้เราทราบตำแหน่งของพื้นที่ข้อมูล ทำให้เราสามารถคำนวณ จำนวน แถวและหลัก ได้จาก การหารขนาดของพื้นที่ข้อมูลด้วยความกว้างและความสูงของบล็อกข้อมูล ซึ่งกำหนดไว้ เท่ากัน เมื่อทำการหักลบระยะ offset ต่างๆ แล้วทำให้เราได้ตำแหน่งของข้อมูล โดยทำการมาร์คและแสดงใน ภาพที่ 3.12



ภาพที่ 3.12 ตำแหน่งของข้อมูลที่ระบุด้วยจุดสีขาว

### 3.4.3 การอ่านบล็อกสีอ้างอิงและการอ่านข้อมูล

ตามที่ได้กล่าวไว้ในเรื่องการสร้างบาร์โค้ด การเรียงลำดับของบล็อกจะเริ่มจากบล็อกสีอ้างอิง 16 บล็อก ตามด้วยชนิดของข้อมูล 1 บล็อก และจำนวนตัวอักษรของข้อมูล 2 บล็อก

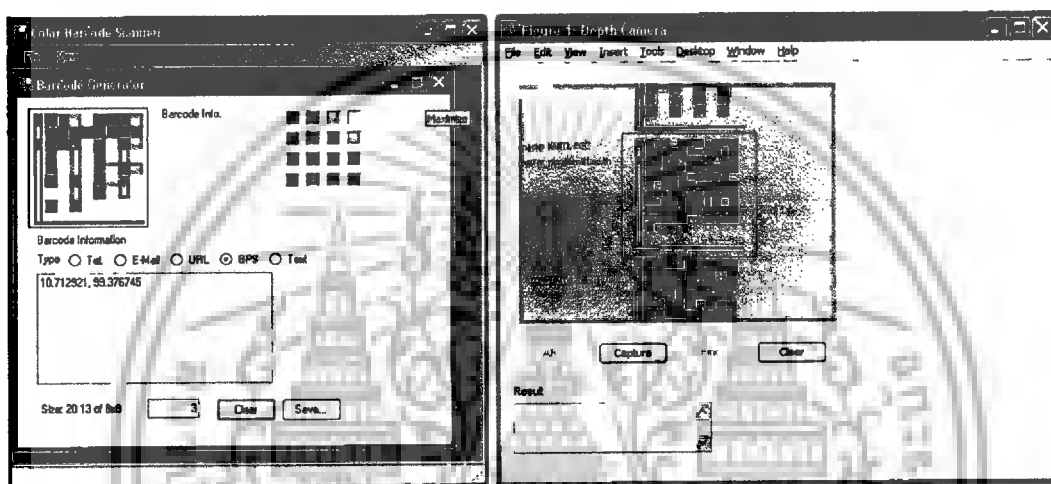
การอ่านค่าสีของบล็อกสีอ้างอิง ทั้ง 3 มิติ (RGB) ทำได้โดยหาค่าเฉลี่ยบริเวณที่ทำการมาร์คไว้ แล้วเก็บเป็นค่าอ้างอิง มีทั้งหมด 16 ชุดซึ่งจะแทนข้อมูลได้ 4 บิต

ในการถอดรหัสข้อมูลทำได้โดยหาค่าความแตกต่างที่น้อยที่สุดระหว่างบล็อกสีที่ต้องการหาค่ากับ ค่าอ้างอิงทั้ง 16 ชุด โดยทำนองเดียวกันสามารถถอดรหัสข้อมูลได้ 4 บิต ต่อ 1 บล็อก โดยจะทำการอ่านจนครบ เท่ากับจำนวนข้อมูลคูณด้วย 2

## บทที่ 4

### การทดลองและผลการทดลอง

ในบทนี้เป็นการทดลองและผลการทดลองที่ได้ทำการทดสอบระบบที่ออกแบบและพัฒนา โดยมีรูปภาพแสดงหน้าต่างการสร้างบาร์โค้ดสีดังภาพที่ 4.1 โดยทางด้านซ้ายจะเป็นโปรแกรมสร้างบาร์โค้ดสี โดยกำหนดรหัสบาร์โค้ดสีแต่ละรหัสแทนข้อมูลจำนวน 4 บิต หรือเทียบได้กับใช้รหัสบาร์โค้ดสีจำนวน 2 บล็อกต่อจำนวนข้อมูล 1 ตัวอักษร ทางด้านขวาจะเป็นหน้าต่างของโปรแกรมอ่านบาร์โค้ดโดยรับภาพจากกล้องเว็บแคม



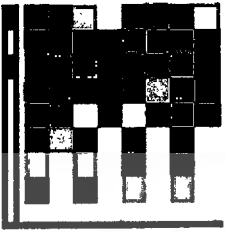
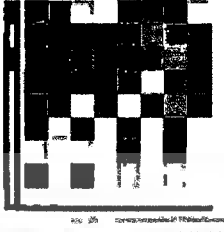


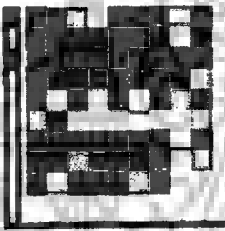

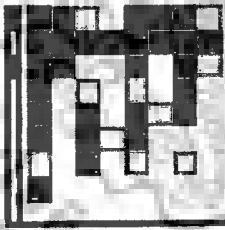

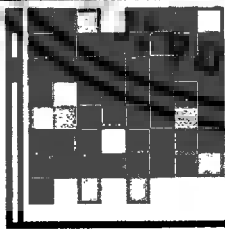
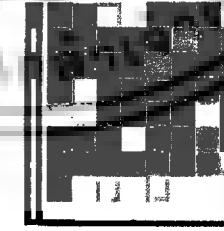
ภาพที่ 4.1 หน้าต่างโปรแกรมในการสร้างและอ่านบาร์โค้ดสี

#### 4.1 สภาพแวดล้อมในการทดลองและผลการทดลองพื้นฐาน

ในการทดลองจะใช้กล้องเว็บแคม ยี่ห้อ OKER 16.0 Mega pixels ชนิดปรับโฟกัสได้ ติดตั้งบนขาตั้งระยะห่างจากเลนส์ถึงบาร์โค้ดที่นำมาอ่านประมาณ 10 เซนติเมตร ทำการบันทึกภาพในความละเอียด 640 x 480 พิกเซล ในสภาพแสงภายในอาคารส่องสว่างด้วยหลอดฟลูออเรสเซนต์ภายในห้อง ความสว่าง ประมาณ 270 lx

ทำการสร้างบาร์โค้ดสีโดยเลือกชนิดข้อมูลทั้ง 5 แบบ แล้วนำไปพิมพ์ลงกระดาษขาว 80 แกรม ด้วยเครื่องพิมพ์อิงค์เจ็ท HP Deskjet F2180 ความละเอียดชนิดดีที่สุดในค่าด้วยโปรแกรมอ่านบาร์โค้ดสี จะได้ผลการทดลองดังตารางที่ 4.1 จากผลการทดลองในหัวข้อนี้จะเห็นได้ว่าบาร์โค้ดทั้งหมดสามารถอ่านค่าได้อย่างถูกต้อง โดยเฉพาะกรณีข้อมูลในแถวที่ 2 มีปริมาณข้อมูลมาก (45 ไบต์) ทำให้ขนาดของบาร์โค้ดสีแต่ละตำแหน่งมีขนาดเล็กกว่าเมื่อเทียบกับการทดลองในแถวอื่นๆ

**ตารางที่ 4.1 ผลการทดลองสร้างและอ่านบาร์โค้ดสี**

ชนิด	บาร์โค้ดที่สร้าง	บาร์โค้ดในการอ่าน	ข้อมูลที่อ่านได้
ข้อมูล			
Tel ----- KMITL PCC +6677506422			KMITL PCC +6677506422
Email ----- Webmaster KMITL PCC webmaster_pcc@kmitl.ac.th			Webmaster KMITL PCC webmaster_pcc@kmitl.ac.th
URL ----- KMITL http://www.kmitl.ac.th/			KMITL http://www.kmitl.ac.th/
GPS ----- 10.724411,99.374728			10.724411,99.374728
Text ----- Meeting @Bld4 Jan 22			Meeting @Bld4 Jan 22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.2 การทดลองเรื่องปริมาณข้อมูลในขั้นตอนการสร้างบาร์โค้ด

ในการทดลองนี้จะเก็บข้อมูลเรื่องปริมาณข้อมูลที่มีผลต่อจำนวน และขนาดของบาร์โค้ด โดยได้ทำการป้อนข้อมูลด้วยปริมาณที่ต่างกัน แล้วบันทึกผลได้ดังตารางที่ 4.2 ซึ่งจะเห็นได้ว่าอัลกอริธึมจะพยายามให้บาร์โค้ดมีลักษณะเป็นสี่เหลี่ยมจัตุรัส จากการทดสอบอ่านข้อมูลบาร์โค้ดที่มีปริมาณ 255 ตัวอักษร ปรากฏว่าไม่สามารถอ่านค่าได้ อันเนื่องมาจากจำนวนพิกเซลน้อย เมื่อทำการพิมพ์ผ่านเครื่องพิมพ์อิงเจ็ทจะเกิดการปนกันและการซีมของสีที่อยู่ติดกัน

ตารางที่ 4.2 คุณลักษณะของบาร์โค้ดสีที่มีปริมาณข้อมูลต่างกัน

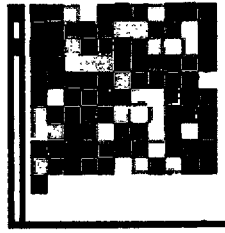
ปริมาณข้อมูล N (ตัว/ไบต์)	ขนาดของรหัสสีบาร์โค้ด (พิกเซล)	จำนวนรหัสสี (แถวxหลัก)
1-2	21	5x5
3-8	17	6x6
9-14	15	7x7
15-22	13	8x8
23-30	11	9x9
31-40	10	10x10
41-50	9	11x11
51-62	8	12x12
63-74	8	13x13
⋮	⋮	⋮
255	4	24x24

#### 4.3 การทดสอบอัลกอริธึมภายใต้สภาวะแสงที่ต่างกัน

ในการทดลองนี้จะนำบาร์โค้ดสีที่ได้พิมพ์แล้วจากตัวอย่างเดิมในการทดสอบตามตารางที่ 4.1 แถวที่ 2 ดังภาพที่ 4.2 ซึ่งแทนข้อมูลที่มีรายละเอียดดังต่อไปนี้

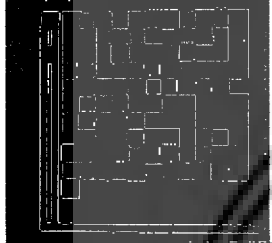
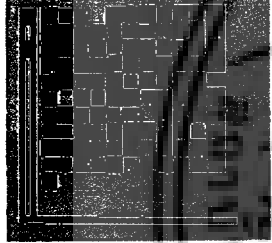
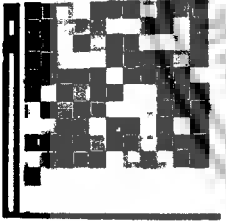
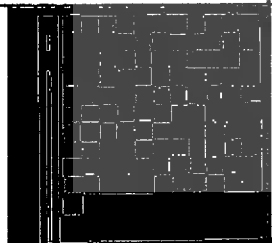
- เป็นข้อมูลชนิด E-mail
- ขนาดของข้อมูล คือ 45 ตัวอักษร
- แทนข้อมูล “Webmaster KMITL PCC  
webmaster\_pcc@kmitl.ac.th”
- ได้บาร์โค้ดขนาด 9 พิกเซล
- จำนวนแถวและหลักของบาร์โค้ด คือ 11x11

ได้นำบาร์โค้ดดังกล่าวมาทำการอ่านค่าข้อมูลภายใต้สภาวะแสงต่างกันออกไป ได้ผลลัพธ์ดังแสดงในตารางที่ 4.3



ภาพที่ 4.2 บาร์โค้ดสีในการทดสอบ

ตารางที่ 4.3 ผลการอ่านบาร์โค้ดในสภาวะแสงต่าง ๆ

บาร์โค้ดในการอ่าน	สภาวะแสง	ข้อมูลที่อ่านได้	หมายเหตุ
	-ไม่มีแสงภายนอกบริเวณ -ใช้กระดาษบังแสงโดยตรง จากหลอดไฟเพดาน -ความเข้มแสง 80 lx	Webmaster KMITL PCC webmaster_pcc@kmitl.ac.th	ถูกต้อง 100%
	-รับแสงโดยตรงจากหลอดไฟ เพดาน -มีเงาของตัวกล้องปรากฏทาง ด้านล่าง -มุมขวาล่างเกิดเงา ค่อนข้างมาก -ความเข้มแสง 170 lx	??ebnasxFW 7ปี ITM P74077z7bnast7W%	
	-มีแสงจากโคมไฟชนิดหลอด ประหยัดไฟด้านข้าง -ความเข้มแสง 550 lx	??ebnas~FW 7ปี ITM J74077z7bnast7W%	
	-วางกระดาษในแนวตั้ง -แสงจากหลอดไฟเพดาน ขนานกับกระดาษ -ความเข้มแสง 150lx	Webmaster KMITL PCC webmaster_pcc@kmitl.ac.th	ถูกต้อง 100%

จากผลที่ได้จะเห็นได้ว่าสภาพแวดล้อมของแสงมีผลต่อบาร์โค้ดอย่างมากเนื่องจากหากการกระจายของแสงไม่สม่ำเสมอจะทำให้การอ่านข้อมูลและถอดรหัสผิดพลาด อีกทั้งพบว่าสีในโทนสีเขียวล้องเว็บแคมมีความสามารถในการแยกขีดน้อยกว่าโทนสีอื่น ๆ

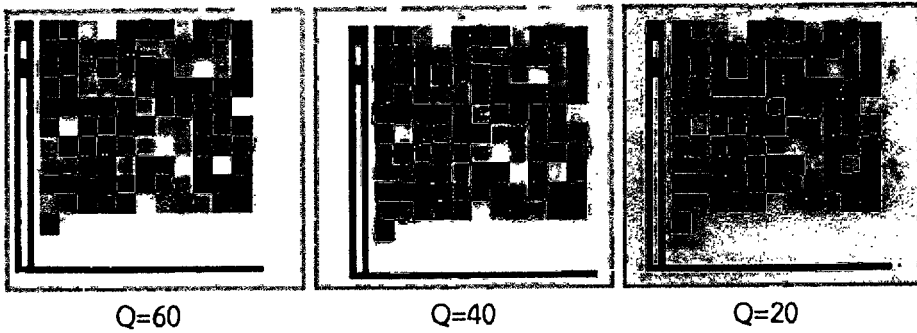
#### 4.4 การทดสอบอัลกอริธึมในภาพบาร์โค้ดที่ถูกบีบอัดด้วย JPEG

ในการทดลองนี้จะนำภาพบาร์โค้ดสีในภาพที่ 4.2 ซึ่งเป็นภาพที่บันทึกไว้แบบไม่มีการบีบอัด (นามสกุลไฟล์ .BMP) มาทำการบีบอัดเพื่อลดขนาดของไฟล์ ตามมาตรฐาน Joint Photographic Experts Group (JPEG) โดยแยกกลุ่มการทดลองต่างตัวแปรค่า Quality (Q) ตั้งแต่คุณภาพภาพสูงไปจนคุณภาพภาพต่ำ (จาก 100 ถึง 10) โดยภาพบาร์โค้ดมีขนาด 128 x 128 พิกเซล จัดวางแผ่นกระดาษที่พิมพ์บาร์โค้ดไว้ในแนวตั้ง และถ่ายภาพจากมุมมองด้านหน้า ไม่ได้ถ่ายภาพจากมุมมองด้านบน เพื่อลดปัญหาผลจากแสงสะท้อนรบกวน

ตารางที่ 4.4 ผลการอ่านบาร์โค้ดที่ไฟล์ภาพถูกบีบอัดด้วย JPEG

ค่า Q	PSNR (dB)	ขนาดไฟล์ (ไบต์)	ความถูกต้องของการอ่าน
100	30.703	17,173	100%
90	29.737	8,303	100%
80	28.214	6,369	100%
70	26.787	5,471	100%
60	25.861	4,874	100%
50	24.946	4,491	91.11% W?bmaster KMITL PCC webmast?r_pcc@??itl.ac.th
40	24.158	4,129	84.44% Webmaster KMITL PCC webmaster_pcc@??it?..?%th
30	22.754	3,667	100%
20	21.454	3,123	93.33% Webmaster KMITL PBC webmaster_pcb@kmitl.ac%th
10	19.609	2,348	ข้อมูลขนาดบาร์โค้ดผิดพลาด ?R?b?fr(MX" ?HJTM← ?? ?@?u?z? ? x? ?hg

จากตารางที่ 4.4 พบว่าการบีบอัดที่ค่า Q =60 เป็นค่าสูงสุดที่ยอมรับได้โดยทั่วไป จากค่า PSNR ที่มีค่ามากกว่า 25 dB แต่หากให้มีการบีบอัดมากขึ้นจะทำให้เกิดสัญญาณรบกวนขึ้นในภาพทำให้การอ่านค่าสีนั้น เกิดข้อผิดพลาดขึ้น โดยแสดงคุณภาพของภาพให้เห็นในภาพที่ 4.3



Q=60                      Q=40                      Q=20  
 ภาพที่ 4.3 บาร์โค้ดที่ถ่ายได้จากกล้องเว็บแคมหลังการบีบอัดด้วย JPEG

จากภาพที่ 4.3 จะเห็นได้ว่าเมื่อมีการบีบอัดแบบ JPEG ค่า Q ต่ำๆ จะเกิดเป็นสัญญาณรบกวนชนิด Block Artifact ตามผลของการบีบอัดแบบ JPEG ที่พบได้ทั่วไป



## บทที่ 5

# สรุปผลการวิจัย

### 5.1 ข้อสรุป

จากการทดลองและผลการทดลองพบว่าอัลกอริธึมสำหรับบาร์โค้ดสีที่นำเสนอ นั้นสามารถใช้งานได้ดีในระดับหนึ่งโดยสามารถสร้างและอ่านบาร์โค้ดสีได้โดยมีขนาดประมาณ  $3 \times 3$  เซนติเมตร จุข้อมูลได้ประมาณ 64 ไบต์ ซึ่งเทียบได้กับตัวอักษร 64 ตัวอักษรซึ่งเพียงพอสำหรับเก็บข้อมูลต่างๆ ไม่ว่าจะเป็น ข้อมูลหมายเลขโทรศัพท์ นามบัตร ข้อมูลจดหมายอิเล็กทรอนิกส์ เว็บไซต์ หรือแม้กระทั่งค่าพิกัดตำแหน่งจีพีเอส

อัลกอริธึมที่นำเสนอ นั้นสามารถใช้กับภาพบาร์โค้ดที่มีการบีบอัดข้อมูลเพื่อลดขนาดไฟล์ภาพลงด้วย JPEG โดยมีค่าคุณภาพที่เหมาะสมคือ Q เท่ากับ 60 ขึ้นไป

### 5.2 ปัญหาที่พบในระหว่างการทำงานวิจัย

1. เนื่องจากกล้องเว็บแคมทั่วไปนั้นจะมีความไวต่อแสงอย่างมาก และข้อจำกัดในการตั้งค่าปรับสมดุลความสว่างแบบอัตโนมัติ (Auto white balance) ทำให้เกิดข้อผิดพลาดในการอ่านบาร์โค้ด ในสภาวะที่มีแสงรบกวนไม่สม่ำเสมอ หรือเกิดเงา

2. ในการพิมพ์ด้วยเครื่องพิมพ์อิงค์เจ็ท ซึ่งมีความละเอียดในการพิมพ์ 600x600 dpi มีความละเอียดทั่วไป ทำให้มีข้อจำกัด แต่ถ้าหากเป็นการพิมพ์ในโรงพิมพ์น่าจะทำให้ผลลัพธ์ดียิ่งขึ้น

### 5.3 แนวทางในการพัฒนาต่อไปในอนาคต

จากปัญหาที่พบในเรื่องของแสงจากสภาพแวดล้อมมีผลต่อประสิทธิภาพของอัลกอริธึมอย่างมาก ซึ่งสามารถหาแนวทางแก้ไขได้เช่นการใช้โมเดลสีอื่นๆ เช่น HSV YCrCb ที่มีผลต่อการเปลี่ยนแปลงของแสงน้อยกว่า เป็นต้น หรือการพัฒนาต่อยอดไปสู่กล้องในโทรศัพท์ เป็นต้น

## บรรณานุกรม

- [1] Parikh, D.; Jancke, G., "Localization and Segmentation of A 2D High Capacity Color Barcode," Applications of Computer Vision, 2008. WACV 2008. IEEE Workshop on , vol., no., pp.1,6, 7-9 Jan. 2008.
- [2] Bo Chen; Xiao-hui Qiang; Ling Yu, "A lightweight color barcode algorithm and application in mobile e-commerce," Computer Science & Education (ICCSE), 2013 8th International Conference on , vol., no., pp.805,809, 26-28 April 2013
- [3] Xiaofei Feng; Herong Zheng, "Design and realization of 2D color barcode with high compression ratio," Computer Design and Applications (ICCD), 2010 International Conference on , vol.1, no., pp.V1-314,V1-317, 25-27 June 2010
- [4] Songwen Pei; Guobo Li; Baifeng Wu, "Codec System Design for Continuous Color Barcode Symbols," Computer and Information Technology Workshops, 2008. CIT Workshops 2008. IEEE 8th International Conference on , vol., no., pp.539,544, 8-11 July 2008
- [5] Tan, K.T.; Chai, D.; Kato, H.; Siong Khai Ong, "Designing a Color Barcode for Mobile Applications," Pervasive Computing, IEEE , vol.11, no.2, pp.50,55, Feb. 2012
- [6] Tan, A.K.T.; Chai, D., "Improving mobile color 2D-barcode JPEG image readability using DCT coefficient distributions," Audio, Language and Image Processing (ICALIP), 2012 International Conference on , vol., no., pp.123,128, 16-18 July 2012
- [7] Kato, H., "Performance of a color 2D barcode as a pervasive computing tool," Intelligent Signal Processing and Communication Systems (ISPACS), 2010 International Symposium on , vol., no., pp.1,4, 6-8 Dec. 2010
- [8] H. Kato and K. T. Tan, "Pervasive 2D-barcode for camera phone applications", IEEE Pervasive Computing, Volume 6, Issue 4, Oct.-Dec. 2007, pp.76 – 85.
- [9] ColorZip™ Japan Inc, "What is ColorCode™," 2006;<http://www.colorzip.co.jp/en/about/technology.html>.
- [10] Microsoft, "High Capacity Color Barcode Technology," Feb. 2009;<http://research.microsoft.com/en-us/projects/hccb/about.aspx>.
- [11] Pointil Systems, "Barcode Design Tips," Feb. 2009;<http://www.pointil.com/resources/barcodetips.htm>.
- [12] Rohs, M., "Real-World Interaction with Camera-Phones", 2nd International Symposium on

Ubiquitous Computing Systems (UCS 2004), Tokyo, Japan, 2004, pp. 39 - 48.

[13] D. Briggs, "The Dimensions of Colour," Nov. 2007; <http://www.huevaluechroma.com/>.

[14] Wellner, Pierre D., Adaptive Thresholding for the DigitalDesk, Technical Report EPC-93-110, Technical Report Rank Xerox Research Centre Cambridge, UK, 1993.

[15] Heckbert, P.S., Fundamentals of Texture Mapping and Image Warping, Technical Report, 1989, Berkeley, CA, USA, 1989

[16] H. Kato, "2D-barcodes for Mobile Devices," doctoral thesis, School of Computer and Information Science, Edith Cowan University, 2009.

[17] S. K. Ong, D. Chai and K. T. Tan, "The use of border in colour 2D barcode," 2008 International Symposium on Parallel and Distributed Processing with Applications (ISPA'2008), Sydney, Australia, 10-12 Dec. 2008, pp. 999-1005.





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ส่วนโปรแกรมหลักในการสังเคราะห์บาร์โค้ดสี

Option Explicit On

Public Class BarcodeForm  
#Region " Fields "

```
''' <summary>  
''' The point at which a draw or drag operation started.  
''' </summary>  
Private startPoint As Point  
  
Private xOffset As Integer = 5  
Private yOffset As Integer = 5  
  
''' <summary>  
''' The point at which the current drawing operation ends.  
''' </summary>  
Private endPoint As Point  
''' <summary>  
''' The boxes the user has drawn on the form.  
''' </summary>  
Private boxes As New List(Of Rectangle)  
  
Private colorsOfBox As New List(Of Color)  
  
Private colorsOfTemplate As New List(Of Color)  
  
Private posOfBox As New List(Of Point)  
''' <summary>  
''' The index of the box the user has clicked on to  
''' drag or right-clicked on to show a context menu.  
''' </summary>  
Private selectedIndex As Integer  
''' <summary>  
''' Indicates whether the application is currently in draw mode.  
''' </summary>  
''' <remarks>  
''' When not in draw mode the app is in drag mode.  
''' </remarks>  
Private drawMode As Boolean = True  
Private numOfBox As Integer  
Private sizeOfBox As Integer
```

#End Region 'Fields

#Region " Event Handlers "

#End Region

#Region " Methods "

```
''' <summary>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

''' Moves a box in the list to change its position in the z-order.
''' </summary>
''' <param name="oldIndex">
''' The index of the box to move.
''' </param>
''' <param name="newIndex">
''' The index to move the box to.
''' </param>
Private Sub ChangeRectangleIndex(ByVal oldIndex As Integer, ByVal newIndex As
Integer)
'Get the box at the specified index.
Dim box As Rectangle = Me.bboxes(oldIndex)

'Removing a box will decrement all subsequent indexes so, if the
'new index is after the old, the new index must be decremented.
If oldIndex < newIndex Then
    newIndex -= 1
End If

'Move the box.
Me.bboxes.RemoveAt(oldIndex)
Me.bboxes.Insert(newIndex, box)

'Redraw the area occupied by the box.
Me.InvalidateRectangle(box)
Me.picBarcode.Update()
End Sub

''' <summary>
''' Gets the box described by two diagonally opposite points.
''' </summary>
''' <param name="startPoint">
''' The first point that describes the box.
''' </param>
''' <param name="endPoint">
''' The second point that describes the box.
''' </param>
''' <returns>
''' A <see cref="Rectangle" /> described by the two specified points.
''' </returns>
Private Function GetRectangle(ByVal startPoint As Point, _
ByVal endPoint As Point) As Rectangle
'The top is the lesser of the two X coordinates.
'The left is the lesser of the two Y coordinates.
'The width is the absolute difference of the two X coordinates.
'The height is the absolute difference of the two Y coordinates.
Return New Rectangle(Math.Min(startPoint.X, endPoint.X), _
Math.Min(startPoint.Y, endPoint.Y), _
Math.Abs(startPoint.X - endPoint.X), _
Math.Abs(startPoint.Y - endPoint.Y))
End Function

''' <summary>
''' Gets the index of the front-most box containing the specified point.
''' </summary>
''' <param name="location">
''' A location within the box to find.
''' </param>
''' <returns>

```

```

''' An <b>Int32</b> containing the index of a box,
''' or -1 if the specified location is not within a box.
''' </returns>
Private Function GetRectangleIndexAtPoint(ByVal location As Point) As Integer
    Dim box As Rectangle

    'Assume no box by default.
    Dim result As Integer = -1

    'Loop backwards so we check front-most boxes first.
    For index As Integer = Me.Boxes.Count - 1 To 0 Step -1
        box = Me.Boxes(index)

        If box.Contains(location) Then
            'The current box contains the specified location.
            result = index

            'Look no further.
            Exit For
        End If
    Next

    Return result
End Function

''' <summary>
''' Invalidates an area of the form containing the specified box.
''' </summary>
''' <param name="box">
''' A <see cref="Rectangle" /> that needs to be repainted.
''' </param>
Private Sub InvalidateRectangle(ByVal box As Rectangle)
    'When invalidating a Rectangle the right-most
    'column and bottom-most row of pixels is excluded.
    'Inflate the box by 1 pixel in every direction to
    'ensure those excluded pixels are also repainted.
    box.Inflate(1, 1)

    Me.picBarcode.Invalidate(box)
End Sub

#End Region 'Methods

Private intTypeOfBC As UShort

Private Sub BarcodeForm_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Me.colorsOfTemplate.Clear()
    Me.colorsOfTemplate.Add(Panel1.BackColor)
    Me.colorsOfTemplate.Add(Panel2.BackColor)
    Me.colorsOfTemplate.Add(Panel3.BackColor)
    Me.colorsOfTemplate.Add(Panel4.BackColor)
    Me.colorsOfTemplate.Add(Panel5.BackColor)
    Me.colorsOfTemplate.Add(Panel6.BackColor)
    Me.colorsOfTemplate.Add(Panel7.BackColor)
    Me.colorsOfTemplate.Add(Panel8.BackColor)
    Me.colorsOfTemplate.Add(Panel9.BackColor)
    Me.colorsOfTemplate.Add(Panel10.BackColor)

```

```

Me.colorsOfTemplate.Add(Panel11.BackColor)
Me.colorsOfTemplate.Add(Panel12.BackColor)
Me.colorsOfTemplate.Add(Panel13.BackColor)
Me.colorsOfTemplate.Add(Panel14.BackColor)
Me.colorsOfTemplate.Add(Panel15.BackColor)
Me.colorsOfTemplate.Add(Panel16.BackColor)

```

```

'Start type from #1
intTypeOfBC = 1
End Sub

```

```

Private Sub picBarcode_Paint(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles picBarcode.Paint
    Dim nBrush As SolidBrush = New SolidBrush(Color.Black)
    Dim H, W, H1, W1, i As Integer
    Dim nPen As Pen
    W = Val(txtSize.Text)
    H = 118
    H1 = 123 - W
    W1 = xOffset + (W * 2)
    With e.Graphics
        .Clear(Color.White)
        .FillRectangle(nBrush, xOffset, yOffset, W, H)
        .FillRectangle(nBrush, W1, yOffset, W, H)
        .FillRectangle(nBrush, xOffset, H1, H, W)
        .FillRectangle(nBrush, W1 - W, yOffset, W, sizeOfBox)
        .FillRectangle(nBrush, W1 - W, yOffset + sizeOfBox + sizeOfBox, W,
sizeOfBox)
        i = 0
        'Draw all the boxes one by one, from back-most to front-most.
        For Each box As Rectangle In Me.bboxes
            If (i < Me.colorsOfBox.Count) Then
                nBrush = New SolidBrush(Me.colorsOfBox.Item(i))
                nPen = New Pen(nBrush)
            Else
                nBrush = New SolidBrush(Color.White)
                nPen = New Pen(nBrush)
            End If
            .FillRectangle(nBrush, box)
            .DrawRectangle(nPen, box)
            i = i + 1
        Next
    End With
End Sub

```

```

Private Sub txtInfo_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles txtInfo.TextChanged
    '(txtInfo.TextLength * 2) + x)) + 1
    ' x=5 -> 4bit type + 16bit size
    ' x=3 -> 4bit type + 8bit size
    ' x=19 -> 4bit type + 8bit size + 16 Ref.Color 'change to 19 for Ref in May
25,2014

    numOfBox = Fix(Math.Sqrt((txtInfo.TextLength * 2) + 19)) + 1
    sizeOfBox = Fix(106 / numOfBox)

    lblSize.Text = "Size: " & txtInfo.TextLength & " " & sizeOfBox & " of " &
numOfBox & "x" & numOfBox

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 31  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End Sub

```
Private Sub btnGen_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnGen.Click
    Dim nBox As Rectangle
    Dim nPoint, nPoint2 As Point
    Dim i, j As Short
    Dim nSize As Size

    Dim byteLenOfInfo As UShort
    Dim byteBuff, byteData As Byte
    Dim strBuff As String
    'Toggle between draw mode and drag mode.
    Me.drawMode = Not Me.drawMode

    nSize = New Size(sizeOfBox, sizeOfBox)
    'Indicate to the user what the next button click will do.
    If Me.drawMode Then
        Me.btnGen.Text = "Generate"
        If Me.bboxes.Count > 0 Then Me.bboxes.Clear()
        If Me.colorsOfBox.Count > 0 Then Me.colorsOfBox.Clear()
        '   groupBox1.Text = Me.bboxes.Count
        picBarcode.Invalidate()
    Else
        For i = 0 To (numOfBox - 1)
            For j = 0 To numOfBox - 1
                nPoint = New Point(17 + (j * sizeOfBox), 5 + (sizeOfBox * i))
                nPoint2.X = nPoint.X + sizeOfBox
                nPoint2.Y = nPoint.Y + sizeOfBox
                nBox = Me.GetRectangle(nPoint, nPoint2)
                Me.bboxes.Add(nBox)
                Me.InvalidateRectangle(nBox)
                Me.picBarcode.Update()
            Next
        Next
        Me.btnGen.Text = "Clear"
        '   groupBox1.Text = Me.bboxes.Count

        'Creating Ref. Block
        For i = 0 To 15
            Me.colorsOfBox.Add(colorsOfTemplate.Item(i))
        Next

        ' Creating Type block
        Me.colorsOfBox.Add(colorsOfTemplate.Item(intTypeOfBC))

        ' Creating Length of Info 8 bit
        byteLenOfInfo = txtInfo.TextLength
        'byteBuff = byteLenOfInfo >> 12
        'byteBuff = byteBuff And &HF
        'Me.colorsOfBox.Add(colorsOfTemplate.Item(byteBuff))
        'byteBuff = byteLenOfInfo >> 8
        'byteBuff = byteLenOfInfo And &HF
        'Me.colorsOfBox.Add(colorsOfTemplate.Item(byteBuff))
        byteBuff = byteLenOfInfo >> 4
        byteBuff = byteBuff And &HF
        Me.colorsOfBox.Add(colorsOfTemplate.Item(byteBuff))
        byteBuff = byteLenOfInfo And &HF
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Me.colorsOfBox.Add(colorsOfTemplate.Item(byteBuff))

'Creating Databox
For i = 1 To txtInfo.TextLength
    strBuff = txtInfo.Text
    strBuff = Mid(strBuff, i, 1)
    byteData = Convert.ToByte(strBuff(0))
    byteBuff = byteData >> 4
    byteBuff = byteBuff And &HF
    'byteBuff = byteBuff Xor &HA
    Me.colorsOfBox.Add(colorsOfTemplate.Item(byteBuff))
    byteBuff = byteData And &HF
    'byteBuff = byteBuff Xor &HA
    Me.colorsOfBox.Add(colorsOfTemplate.Item(byteBuff))
Next

picBarcode.Invalidate()

```

```

End If ' Generating
End Sub

```

```

Private Sub radTel_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles radTel.CheckedChanged
    txtInfo.Text = "Name" + vbCrLf + "+66(0) xxx xxxx"
    txtInfo.SelectAll()
    intTypeOfBC = 1
End Sub

```

```

Private Sub radEMail_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles radEMail.CheckedChanged
    txtInfo.Text = "Name Surname" + vbCrLf + "+name@host.com"
    txtInfo.SelectAll()
    intTypeOfBC = 2
End Sub

```

```

Private Sub radURL_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles radURL.CheckedChanged
    txtInfo.Text = "http://www.abcde.com"
    txtInfo.SelectAll()
    intTypeOfBC = 3
End Sub

```

```

Private Sub radGPS_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles radGPS.CheckedChanged
    txtInfo.Text = "-10.712921, 99.376745"
    txtInfo.SelectAll()
    intTypeOfBC = 4
End Sub

```

```

Private Sub radText_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles radText.CheckedChanged
    txtInfo.Text = "Enter text here"
    txtInfo.SelectAll()
    intTypeOfBC = 5
End Sub

```

```

Private Sub btnSaveBarcode_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSaveBarcode.Click

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 33  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Dim newBitmap As Bitmap = New Bitmap(128, 128)
Dim g As Graphics
g = Graphics.FromImage(newBitmap)

Dim nBrush As SolidBrush = New SolidBrush(Color.Black)
Dim H, W, H1, W1, i As Integer
Dim nPen As Pen
W = Val(txtSize.Text)
H = 118
H1 = 123 - W
W1 = xOffset + (W * 2)
With g
    .Clear(Color.White)
    .FillRectangle(nBrush, xOffset, yOffset, W, H)
    .FillRectangle(nBrush, W1, yOffset, W, H)
    .FillRectangle(nBrush, xOffset, H1, H, W)
    .FillRectangle(nBrush, W1 - W, yOffset, W, sizeOfBox)
    .FillRectangle(nBrush, W1 - W, yOffset + sizeOfBox + sizeOfBox, W,
sizeOfBox)
    i = 0
    'Draw all the boxes one by one, from back-most to front-most.
    For Each box As Rectangle In Me.Boxes
        If (i < Me.colorsOfBox.Count) Then
            nBrush = New SolidBrush(Me.colorsOfBox.Item(i))
            nPen = New Pen(nBrush)
        Else
            nBrush = New SolidBrush(Color.White)
            nPen = New Pen(nBrush)
        End If
        .FillRectangle(nBrush, box)
        .DrawRectangle(nPen, box)
        i = i + 1
    Next
End With

Dim sfd As New SaveFileDialog
sfd.Title = "Save your barcode as...."
sfd.InitialDirectory = My.Computer.FileSystem.SpecialDirectories.MyPictures
sfd.Filter = "Jpeg files (*.jpg)|*.jpg|Bitmap files (*.bmp)|*.bmp"

Dim result As DialogResult = sfd.ShowDialog
Dim jpgEncoder As System.Drawing.Imaging.ImageCodecInfo =
GetEncoder(System.Drawing.Imaging.ImageFormat.Jpeg)

' Create an Encoder object based on the GUID
' for the Quality parameter category.
Dim myEncoder As System.Drawing.Imaging.Encoder =
System.Drawing.Imaging.Encoder.Quality

' Create an EncoderParameters object.
' An EncoderParameters object has an array of EncoderParameter
' objects. In this case, there is only one
' EncoderParameter object in the array.
Dim myEncoderParameters As New System.Drawing.Imaging.EncoderParameters(1)

Dim myEncoderParameter As New
System.Drawing.Imaging.EncoderParameter(myEncoder, 100&)
myEncoderParameters.Param(0) = myEncoderParameter

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

'bmp1.Save("c:\TestPhotoQualityFifty.jpg", jpgEncoder, myEncoderParameters)

If result = Windows.Forms.DialogResult.OK Then
    If sfd.FileName <> String.Empty Then
        'Set to Jpeg by default.
        Dim MyImageFormat As System.Drawing.Imaging.ImageFormat =
System.Drawing.Imaging.ImageFormat.Jpeg
        If sfd.FileName.ToString.ToUpper.EndsWith("JPG") Then
            MyImageFormat = System.Drawing.Imaging.ImageFormat.Jpeg
        ElseIf sfd.FileName.ToString.ToUpper.EndsWith("BMP") Then
            MyImageFormat = System.Drawing.Imaging.ImageFormat.Bmp
        End If
        newBitmap.Save(sfd.FileName, jpgEncoder, myEncoderParameters)
    End If
End If

End Sub

Private Function GetEncoder(ByVal format As System.Drawing.Imaging.ImageFormat) As
System.Drawing.Imaging.ImageCodecInfo

    Dim codecs As System.Drawing.Imaging.ImageCodecInfo() =
System.Drawing.Imaging.ImageCodecInfo.GetImageDecoders()

    Dim codec As System.Drawing.Imaging.ImageCodecInfo
    For Each codec In codecs
        If codec.FormatID = format.Guid Then
            Return codec
        End If
    Next codec
    Return Nothing
End Function

End Class

```

### % ส่วนโปรแกรมหลัก ในการอ่านบาร์โค้ด

```

function DepthGui2()

    orgX = 0;
    orgY = 0;
    blockSize = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

oriOffset = 0;
DepthVid = videoinput('winvideo', 1);%make object for depth camera
set(DepthVid,'FramesPerTrigger',1);%capture 1 frame every time
                                     %is triggered
set(DepthVid,'TriggerRepeat',Inf);%infinite amount of triggers
triggerconfig(DepthVid, 'Manual');%trigger Depthvid manually
%within program      %create figure to contain buttons/axes
window=figure('Color',[0.9255 0.9137 0.8471],'Name','Depth Camera',...
    'DockControl','off','Units','Pixels',...
    'toolbar','none',...
    'Position',[200 200 560 420]);
set(gcf, 'WindowButtonMotionFcn', @mouseMove);
set(gcf, 'CloseRequestFcn', @myCloseRequestFcn);
myim = [];
%create axis to contain video feed
ax1=axes('Parent',window,'XTick',[],'YTick',[],...
    'XColor',[0.9255 0.9137 0.8471],...
    'YColor',[0.9255 0.9137 0.8471],...
    'Units','Pixels',...
    'Position',[20 160 320 240]);
%create timer that calls the dispim fctn every 0.05 secs
t = timer('TimerFcn',@dispim, 'Period', 0.1,...
    'executionMode','fixedRate');
%create start button which calls the startbCallback fctn
startb=icontrol('Parent',window,'Style','pushbutton','String',...
    'START',...
    'Units','Pixels',...
    'Position',[14 119 70 22],...
    'Callback',@startbCallback);
%create stop button which calls the stopbCallback fctn
stopb=icontrol('Parent',window,'Style','pushbutton',...
    'String','Capture',...
    'Enable','off',...
    'Units','Pixels',...
    'Position',[100 119 70 22],...
    'Callback',@stopbCallback);

findposb=icontrol('Parent',window,'Style','pushbutton',...
    'String','Find',...
    'Enable','off',...
    'Units','Pixels',...
    'Position',[180 119 70 22],...
    'Callback',@findposbCallback);

clearb=icontrol('Parent',window,'Style','pushbutton',...
    'String','Clear',...
    'Enable','on',...
    'Units','Pixels',...
    'Position',[260 119 70 22],...
    'Callback',@clearbCallback);
resulttxt = uicontrol('Parent',window,'Style','text',...
    'String','Result',...
    'HorizontalAlignment','left',...
    'Position',[14 78 50 20]);
dataedit =uicontrol('Parent',window,'Style','edit',...
    'BackgroundColor',[1.0 1.0 1.0],...
    'Max',10,...

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'HorizontalAlignment','left',...
'Position',[14 30 200 50]);
```

%dispim fuction does all image processing and displays the image

**% ส่วนโปรแกรมย่อยในการแสดงผลวิดีโอ และ Overlay ก่อนบันทึกภาพ**

```
function dispim(hobj,event)
    trigger(DepthVid)%trigger DepthVid to capture image
    im=getdata(DepthVid,1,'double');%convert to double
    %superimpose red box
    im(100:350,210,2:3)=0;
    im(100:350,210,1)=1;
    im(100:350,480,2:3)=0;
    im(100:350,480,1)=1;
    im(100,210:480,2:3)=0;
    im(100,210:480,1)=1;
    im(350,210:480,2:3)=0;
    im(350,210:480,1)=1;

    %change red box to blue if area is "touched"
    myim = im(101:349,211:479,:);
    imshow(myim);
    %display im
    drawnow;
end
```

**% ส่วนโปรแกรมย่อยเริ่มเปิดเว็บแคม**

```
function startbCallback(hobj,event)
    start(DepthVid);%open object
    start(t)%open object
    set(stopb,'Enable','on');
    set(startb,'Enable','off');
end
```

**% ส่วนโปรแกรมย่อยเริ่มหยุดเว็บแคมและใช้ภาพสุดท้ายก่อนหยุด**

```
function stopbCallback(hobj,event)
    stop(t)%close object
    stop(DepthVid);%close object
    set(stopb,'Enable','off');
    set(findposb,'Enable','on');
    imshow(myim);
end
```

**% ส่วนโปรแกรมย่อยในการเริ่มต้นใหม่**

```
function clearbCallback(hobj,event)
    stop(t)%close object
    stop(DepthVid);%close object
    set(startb,'Enable','on');
    set(stopb,'Enable','off');
    set(findposb,'Enable','off');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

subplot(1,1,1);
end

```

% ส่วนโปรแกรมย่อยในการอ่านค่าบาร์โค้ด

```

function findposbCallback(hobj,event)
[r,c] = size(myim);

%Find minimum border blackground gray level
myimGray = rgb2gray(myim);
grayTh = min(myimGray(1,:)); %Top
grayTh2 = min(myimGray(r,:)); %Buttom
if( grayTh > grayTh2)
    grayTh = grayTh2;
end
grayTh2 = min(myimGray(:,1)); %Left
if( grayTh > grayTh2)
    grayTh = grayTh2;
end
grayTh2 = min(myimGray(r,:)); %Right
if( grayTh > grayTh2)
    grayTh = grayTh2;
end
grayTh = grayTh/2.0;
myimGray = (myimGray < grayTh);
subplot(221);imshow(myimGray);
topProject = sum(myimGray);
subplot(222);plot(topProject);

%Find double Reference Bar (ver) boundary
N=1;
mybound = [];
topMax = max(topProject);
top_half = topMax /2;
while (topProject(N) < top_half)
    N = N+1;
end
mybound = [mybound N]; %first col of ref #1 1

while (topProject(N)>= top_half)
    N = N+1;
end
mybound = [mybound N]; %first hold between bars 2
while (topProject(N) < top_half)
    N = N+1;
end
mybound = [mybound N]; %first col of ref #2 3
while (topProject(N)>= top_half)
    N = N+1;
end
mybound = [mybound (N-1)]; %last col of Ref #2 4

N = size(topProject,2);
%Find last col of Barcode
while(topProject(N) < (mybound(4)-mybound(1)))
    N = N-1;
end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end
mybound = [mybound N]; %last col of Barcode          5

%Find Box size
leftProject = sum(myimGray(:,mybound(1):mybound(4)),2)'; %transpose
leftMax = max(leftProject) * 0.9;

N=1;
while (leftProject(N) < leftMax)
    N = N+1;
end
mybound = [mybound N]; %first ROW of ref #1          6

while (leftProject(N)>= leftMax)
    N = N+1;
end
mybound = [mybound (N-1)]; %first hold FROM TOP      7

%Store data
oriY = mybound(6);
oriX = round((mybound(1)+(mybound(4)-mybound(1)+1)*(4/3))-1);
blockSize = (mybound(7)-mybound(6)+1);
oriOffset = floor((blockSize-1)/2);
numOfBlock = round((mybound(5)-oriX)/blockSize);

%Mark position
x = oriX + oriOffset;
y = oriY + oriOffset;

myimGray = myim;

%create
readIm = zeros(numOfBlock*numOfBlock,3);
readOffset = floor(blockSize/5);
%Find center of each data block
y = y - blockSize;
N = 1;
for i = 1:numOfBlock
    y = y + blockSize;
    x = oriX - oriOffset;
    for j = 1:numOfBlock
        x = x + blockSize;
        %Mark
        myimGray(y,x,1) = 1;
        myimGray(y,x,2:3) = 0;
        %Find Mean
        readIm(N,:) = mean(mean(myim(y-readOffset:y+readOffset,x-
readOffset:x+readOffset,:)));
        N = N+1;
    end
end
disp(sprintf('Total readed block : %d\n',N-1));

%Find Data
bitdat = [];
Ref = readIm(1:16,:);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Data = readIm(17:end,:);
for i =1:size(Data,1)
    dist = bsxfun(@minus, Ref,Data(i,:) );
    dist2 = sqrt(sum(dist.*dist,2));
    [mindist, column] = min(dist2);
    bitdat = [bitdat column-1];
end

myType = bitdat(1);
disp(sprintf('Barcode Type : %d',myType));
myByteOfData = (bitdat(2)*16)+bitdat(3);
disp(sprintf('Data length (char) : %d',myByteOfData));

dataString = '';
N=4;
for i=1:myByteOfData
    temp = (bitdat(N)*16)+bitdat(N+1);
    dataString = sprintf('%s%c',dataString,char(temp));
    N = N+2 ;
end
disp('Data : -----');
disp(dataString)
disp('-----');
set(dataedit,'String',dataString);
imshow(myimGray);
save('para.mat',
'topProject','myimGray','myim','mybound','oriY','oriX','blockSize','numOfBlock','Re
f','Data','bitdat');
%save('readed.mat','readIm');

%Save readed image
temp = clock;
myoutName = sprintf('%s-%d.%d.jpg',date,temp(4),temp(5));
imwrite(myim,myoutName,'jpg');
end

% --- Executes when user attempts to close BarcodeReaderGUI.
function myCloseRequestFcn(hObject,event)
% hObject    handle to BarcodeReaderGUI (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: delete(hObject) closes the figure
delete(hObject);
delete(imaqfind);
end

function mouseMove (object, eventdata)
C = get (gca, 'CurrentPoint');
title(gca, ['(X,Y) = (', num2str(C(1,1)), ', ', num2str(C(1,2)), ')']);
end
end

```

%-----end of code-----



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ข

สรุปค่าใช้จ่ายการดำเนินโครงการวิจัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



แบบรายงานการใช้จ่ายเงินโครงการวิจัย  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

แหล่งงบประมาณแผ่นดิน (แบบปกติ)  แหล่งเงินรายได้

ชื่อโครงการ (ภาษาไทย) อัลกอริทึมสำหรับบาร์โค้ดสองมิติชนิดความสูงโดยใช้บาร์โค้ดสี.....

(ภาษาอังกฤษ) Algorithm for High Capacitance 2D-Barcode using Color Barcode.....

ชื่อ-สกุลหัวหน้าโครงการวิจัยผู้รับทุน/ผู้วิจัย (อ./ดร./ผศ./รศ./ศ.) อ.ศักดิ์พันธ์ คล้ายดอกจันทร์

รายงานในช่วงตั้งแต่วันที่ 1 ตุลาคม 2554 ถึงวันที่ 31 มีนาคม 2556

ระยะเวลาดำเนินการ 1 ปี 6 เดือน ตั้งแต่วันที่ 1 ตุลาคม 2554 ถึงวันที่ 31 มีนาคม 2556

ข้อมูลการรายงานค่าใช้จ่ายงบประมาณโครงการวิจัย

1. การเบิกจ่ายงบประมาณ (กรณีการจ่ายเงินถ้าจ่ายงวดเดียวให้ลบข้อที่ไม่เกี่ยวข้องออก)

งวดที่ 1 82,500 บาท 50 % วันที่ได้รับอนุมัติให้เบิกจ่ายเงิน (ว/ด/ป) 2 พฤศจิกายน 2554

งวดที่ 2 82,500 บาท 50 % วันที่ได้รับอนุมัติให้เบิกจ่ายเงิน (ว/ด/ป) 1 เมษายน 2555

2. สรุปงบประมาณค่าใช้จ่ายที่ใช้นับตั้งแต่เริ่มทำการวิจัยถึงปัจจุบัน (จำแนกตามหมวดค่าใช้จ่าย)

หมวดค่าใช้จ่าย	งบประมาณรวมทั้งโครงการ	ค่าใช้จ่าย (บาท)	คงเหลือ (เกิน)
งบบุคลากร : ค่าจ้างชั่วคราว	30,000	30,000	0.00
งบดำเนินงาน			
ค่าตอบแทน	-	-	-
ค่าใช้สอย	19,500	4,226.55	15,273.45
ค่าวัสดุ	115,500	130,773.45	(15273.45)
ค่าสาธารณูปโภค	-	-	-
งบลงทุน: ค่าครุภัณฑ์	-	-	-
<b>รวม</b>	<b>165,000.00</b>	<b>165,000.00</b>	<b>0.00</b>

(นายศักดิ์พันธ์ คล้ายดอกจันทร์)

ลงนามหัวหน้าโครงการวิจัยผู้รับทุน

10 / ๑๕.๕๖ / ๒๕๕๖

(นางสาวจิระนัย แก้วบังดู)

เจ้าหน้าที่การเงิน

10 ส.พ. 2557

## ข้อมูลประวัติคณะผู้วิจัย

### ประวัติส่วนตัว

ชื่อ-สกุล นายสั๊กกะพันธ์ คล้ายดอกจันทร์  
ตำแหน่งปัจจุบัน อาจารย์  
สถานที่ทำงาน สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง วิทยาเขตชุมพร  
หมู่ที่ 6 ตำบลชุมโค อำเภอปะทิว จังหวัดชุมพร 86160  
โทรศัพท์ 0-7750-6422  
E-mail kmnattay@kmitl.ac.th

### ประวัติการศึกษา

ชื่อย่อปริญญา	สาขา	สถาบันที่จบ	ปีที่จบ
วศ.ม.	วิศวกรรมอิเล็กทรอนิกส์	สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง	2545
วศ.บ.	วิศวกรรมอิเล็กทรอนิกส์	สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง	2542

สาขาวิจัยที่มีความชำนาญพิเศษ (แตกต่างจากวุฒิการศึกษา)

อิเล็กทรอนิกส์ทางการแพทย์และการประมวลผลสัญญาณ ระบบควบคุมและอิเล็กทรอนิกส์กำลัง

ผลงานวิจัย/งานสร้างสรรค์ที่ตีพิมพ์เผยแพร่ (ระดับชาติและนานาชาติ)

- เกษมสุข เสพศิริสุข และสั๊กกะพันธ์ คล้ายดอกจันทร์, “การศึกษาการวัดค่าสารโพลาไรในน้ำมันปาล์ม  
ทอดมันฝรั่งโดยการวัดค่าความจุไฟฟ้า”. การประชุมวิชาการวไลยลักษณวิจัย ครั้งที่ 4, หน้า 131-132,  
21 มิถุนายน 2555.

## ข้อมูลประวัติคณะผู้วิจัย

### ประวัติส่วนตัว

ชื่อ-สกุล นายเกษมสุข เสพศิริสุข  
ตำแหน่งปัจจุบัน อาจารย์  
สถานที่ทำงาน สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง วิทยาเขตชุมพร  
หมู่ที่ 6 ตำบลชุมโค อำเภอปะทิว จังหวัดชุมพร 86160  
โทรศัพท์ 0-7750-6422  
E-mail kskasems@kmitl.ac.th

### ประวัติการศึกษา

ชื่อย่อปริญญา	สาขา	สถาบันที่จบ	ปีที่จบ
D.Eng	Science and Technology	Tokai University, Japan	2552
วศ.ม.	วิศวกรรมอิเล็กทรอนิกส์	สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง	2544
วศ.บ.	วิศวกรรมอิเล็กทรอนิกส์	สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง	2542

สาขาวิจัยที่มีความชำนาญพิเศษ (แตกต่างจากวุฒิการศึกษา)

สาขาเทคโนโลยีสารสนเทศและนิเทศศาสตร์ สาขาวิศวกรรมศาสตร์และอุตสาหกรรมวิจัย

### ผลงานวิจัย/งานสร้างสรรค์ที่ตีพิมพ์เผยแพร่ (ระดับชาติและนานาชาติ)

เกษมสุข เสพศิริสุข และนพดล มณีรัตน์, "ตัวแบบคณิตศาสตร์สำหรับการวัดค่าสารโพลารีนในน้ำมันปาล์มทอดซ้ำสำหรับทอดไก่ด้วยความจุไฟฟ้า". วารสารมหาวิทยาลัยทักษิณ ปีที่ 16 ฉบับที่ 3 ฉบับพิเศษ, หน้า 59-64, พ.ศ. 2556.

เกษมสุข เสพศิริสุข และสั๊กกะพันธ์ คล้ายดอกจันทร์, "การศึกษาการวัดค่าสารโพลารีนในน้ำมันปาล์มทอดมันฝรั่งโดยการวัดค่าความจุไฟฟ้า". การประชุมวิชาการวลัยลักษณ์วิจัย ครั้งที่ 4, หน้า 131-132, 21 มิถุนายน 2555.

K. Sepsirisuk, K. Atsuta, and S. Kondo, "An Improved Wavelet-based Watermarking Method Using A Mathematical Morphology", Proc. WIAMIS, Santorini, Greece, 2007.

K. Sepsirisuk, K. Atsuta, and S. Kondo, "An Adaptive Digital Watermark Based on a Wavelet Tree Structure Using the Human Visual System.", Proc. IEEE ISCIT, Beijing, China, 2005.

K. Sepsirisuk, K. Atsuta, and S. Kondo, "Template Recovery of DWT-DFT Composite Watermarking Scheme Using Collinear Cross-Ratio.", Proc. ICCAS, Seoul, Korea, 2005.

N. Pantsaena, A. Ngamlamiad, C. Pintavirooj, M. Sangworasil and K. Sepsirisuk, "Fast Volume Rendering on Clustering System", World Congress on Medical Physics and Biomedical Engineering (WC2003), August 24-29, 2003. Sydney Australia.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้