



ระบบควบคุม
การติดตามวัตถุอัตโนมัติโดยข้อมูลภาพ

Automatic
control system of object
following by using image data

โดย
นายธนา ศรีแสง
นายประจักษ์ มณีมงคล
นายปัญญา พงษ์วิรัตน์

ปริญญาานิพนธ์ฉบับนี้
เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาอุตสาหกรรมศาสตรบัณฑิต
สาขาเทคโนโลยีคอมพิวเตอร์อุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เจ้าคุณทหาร ลาดกระบัง
ปีการศึกษา 2535

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น ลิดทิ้งความผิดให้ด้วยตนเองและต้องแจ้งถึงเจ้าของเอกสารฉบับนี้เมื่อคืนให้

032676

ปริญญานิพนธ์ ปีการศึกษา 2/2535

ภาควิชา เทคโนโลยีการวัดคุมทางอุตสาหกรรม
สาขาวิชา เทคโนโลยีคอมพิวเตอร์อุตสาหกรรม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง Automatic control system of object following
by using image data

ผู้จัดทำ

1.นาย ธนา	ศรีแสง	34.162207
2.นาย ประจักษ์	มณีมงคล	34.162212
3.นาย ปัญญา	พงษ์วิรัตน์	34.162216

..... น.ร. ส.ร. อาจารย์ที่ปรึกษา
(อาจารย์ ทรงชัย วีระทวีมาศ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

032678

หัวข้อปริญญานิพนธ์	ระบบควบคุมการติดตามวัตถุอัตโนมัติโดยข้อมูลภาพ	
นักศึกษา	นาย ธนา ศรีแสง	34.162207
	นาย ประจักษ์ มณีมงคล	34.162212
	นาย ปัญญา พงษ์วิรัตน์	34.162216
อาจารย์ที่ปรึกษา	อ.ทรงชัย วีระทวิมาศ	
ระดับการศึกษา	อุตสาหกรรมศาสตรบัณฑิตทางเทคโนโลยี	
	การควบคุมอุตสาหกรรม	
ปีการศึกษา	พ.ศ. 2535	

บทคัดย่อ

ในโลกของการนำเอาระบบคอมพิวเตอร์มาประยุกต์ใช้ในปัจจุบันนี้มีมากมายหลายด้าน ซึ่งการนำคอมพิวเตอร์มาใช้ในระบบควบคุมก็เป็นอีกรูปแบบหนึ่งของการประยุกต์ใช้งาน " ระบบควบคุมการติดตามวัตถุอัตโนมัติโดยข้อมูลภาพ " (Automatic control system of object following by using image data) ก็เป็นวิธีการหนึ่งในการนำเอาคอมพิวเตอร์มาใช้ในการควบคุม ซึ่งเป็นการนำข้อมูลภาพมาแปลงเป็นข้อมูลดิจิทัล จากนั้นส่งข้อมูลไปประมวลผลที่ CPU และนำสัญญาณที่ประมวลผลแล้วกลับมาควบคุมมอเตอร์ของกล้องวิดีโอให้เคลื่อนที่ตามวัตถุนั้นอีกทีหนึ่ง โดยกล้องวิดีโอจะเป็นตัวจับภาพและส่งสัญญาณภาพไปแปลงเป็นข้อมูลดิจิทัล และส่งไป CPU เพื่อประมวลผลต่อไป

Thesis : Automatic control system of object following
by using image data

Student	: Mr.Thana Srisaeng	34.162207
	Mr.Prajak Maneemongkol	34.162212
	Mr.Panya Pongvirat	34.162216

Advisor : Mr.Songchai Weerathaweemart

Education Level : Bachelor of industrial instrument
technology

Education year : 1992

Abstract.

In the world of computer application had a many system which use computer to control system and applied to the other elses . " AUTOMATIC CONTROL SYSTEM OF OBJECT FOLLOWING BY USING IMAGE DATA " is the one of many application which also used computer to control the system by used image data invert to digital for processing by CPU of PC . The data which already processed will sent to reference and control motor of vedio camera for movement. The vedio camera will be image tracker and send image signal to ADC (Analog to digital converter) for the next processing.

สารบัญ

บทคัดย่อ		หน้า
Abstract		
บทที่ 1 บทนำ		1
บทที่ 2 การมองภาพ		2
- การแปลงภาพจริงเป็นข้อมูลภาพ		3
บทที่ 3 การประมวลผลภาพเบื้องต้น		8
- Pixel		8
- Gray Scale		12
- Quantizing Error		15
- Histrogram		19
- Construction		22
บทที่ 4 การประมวลผลผลการมองเห็นของเครื่องจักรเบื้องต้น		24
- Introduction		24
- Modanic one point transformation		26
- Dynamic two point transformation		31
- Convolution		39
บทที่ 5 ขอบภาพ		40
- Digital filter		40
- Low pass filter		40
- High pass filter		43
- Edge enhancement		47
บทที่ 6 Stepping Motors		59
- Exitation modes		62
- Single phase exitation		62
- Two phase exitation		63
- Half step mode exitation		64

- Bridge excitation	66
บทที่ 7 การจัด Address สำหรับ Memory และ I/O ของ PC	67
- การใช้งาน Address สำหรับ I/O Port	69
- การ Decode Address สำหรับ I/O Port	72
- วงจร การ Decode Address สำหรับ I/O Port	78
- Characteristic of Digi Image Model DZ-II ...	79
- Driver Circuit And Operating	93
- Power Supply Circuit and Operating	96
บทที่ 8 สรุปผลงานวิจัย	99

กิตติกรรมประกาศ

หนังสืออ้างอิง

ภาคผนวก โปรแกรมคอมพิวเตอร์



บทที่ 1

บทนำ

ปัจจุบันวิธีการประมวลผลภาพ (Image Processing) ได้มีการนำมาใช้ในการประมวลผลของภาพต่างๆ กันอย่างแพร่หลาย เช่น การประมวลผลภาพถ่ายดาวเทียม ระบบการจดจำลายมือเขียน เป็นต้น ซึ่งโครงงานนี้ก็ได้นำหลักการมาประยุกต์ใช้ในการติดตามวัตถุอัตโนมัติโดยใช้ข้อมูลภาพ

ในการติดตามวัตถุจะใช้วิธี กำหนดระดับสีของวัตถุ และระดับสีของพื้นให้มีความแตกต่างกัน

1.1 วัตถุประสงค์ของปริญญานิพนธ์

ในการติดตามวัตถุอัตโนมัติโดยใช้ข้อมูลภาพ ซึ่งใช้วิธีการเปรียบเทียบระดับสีของวัตถุ และสีพื้น นี้ จะมีความรวดเร็วในการเปรียบเทียบข้อมูลมากกว่าแบบที่ใช้ขอบภาพ ดังนั้น จึงสามารถทำงานได้เร็วขึ้น และสามารถที่จะพัฒนา ให้มีประสิทธิภาพมากขึ้นได้

1.2 ขอบเขตของปริญญานิพนธ์

ในส่วนของปริญญานิพนธ์นี้ ต้องการที่จะให้วัตถุสีดำ เคลื่อนที่ในพื้นที่สีขาว ซึ่งมีรายละเอียดกล่าวถึงเทคนิค และวิธีการ ในการประมวลผลข้อมูลภาพ โดยแบ่งย่อยออกเป็นบทต่างๆ ดังต่อไปนี้

บทที่ 2 การมองภาพ กล่าวถึง ระบบการมองภาพที่แท้จริง จะเกี่ยวข้องกับการแปลงภาพ, การวิเคราะห์ภาพ และ การเข้าใจภาพ

บทที่ 3 การประมวลผลการภาพเบื้องต้น เป็นพื้นฐานของระบบการประมวลผลภาพ การแทนภาพ ในระบบดิจิทัล แบบต่างๆ

บทที่ 4 กล่าวถึงการประมวลผลการมองเห็นของเครื่องจักรเบื้องต้น

บทที่ 5 กล่าวถึงการหาขอบภาพของวัตถุ โดยวิธีการต่างๆ

บทที่ 6 กล่าวถึง Stepping Motor และ การควบคุมการทำงาน

บทที่ 7 กล่าวถึงการจัดแอดเดรสของเครื่อง PC และ การต่อ Hard Ware ต่างๆ

บทที่ 8 เป็นบทสรุปรวบรวมผลงานวิจัยสำหรับทุกๆ บท

ภาคผนวก เป็นรายละเอียดของโปรแกรม

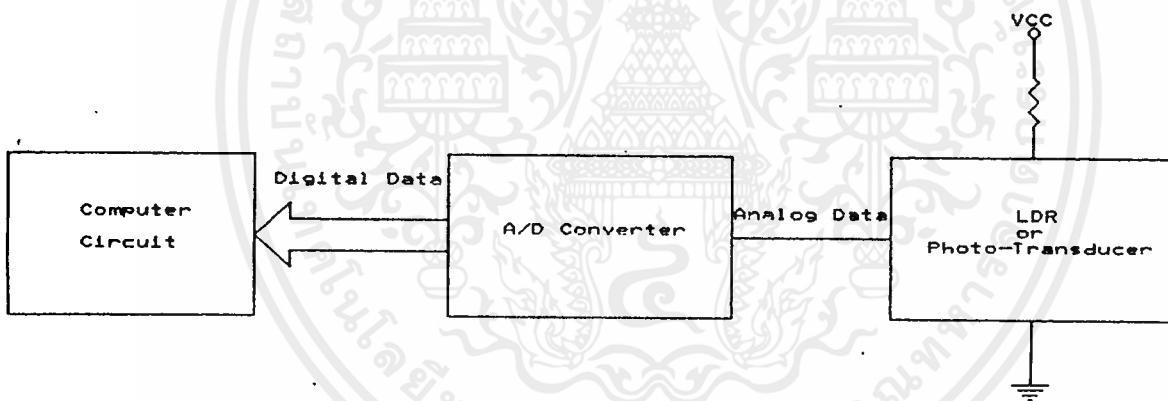
บทที่ 2

การมองภาพ

หุ่นยนต์จะมีความฉลาดมากขึ้นถ้ามีการรับรู้ข้อมูลของสิ่งรอบตัวด้วยตัวของมันเอง โดยเฉพาะอย่างยิ่งการมองด้วยตาหุ่นยนต์ มิใช่การป้อนข้อมูลจากผู้ใช้งานเท่านั้น

การมองภาพเบื้องต้นของคอมพิวเตอร์โดยทั่วไปเป็นการรับรู้ภาพ (Visual Perception) ที่ทำได้เพียงรับรู้ว่ามีวัตถุที่มองอยู่มีรูปร่างเป็นอย่างไร โดยอาศัยหลักการของการตรวจจับความเข้มแสง

ระบบนี้จะใช้วงจรที่ประกอบด้วยตัวต้านทานที่เปลี่ยนค่าตามแสง (LDR) หรือ โฟโตทรานซิสเตอร์ เพื่อแปลงระดับความเข้มแสงเป็นแรงดันไฟฟ้า แล้วจึงเปลี่ยนมาเป็นสัญญาณดิจิทัลเพื่อส่งต่อไปยังคอมพิวเตอร์ ดังแสดงในรูปที่ 2.1 การมองภาพแบบนี้ใช้กับโปรแกรมง่าย ๆ ที่บอกเพียงว่าวัตถุอยู่ในตำแหน่งที่กำหนดแล้วหรือยัง หรือกล่าวได้ว่าระบบนี้เป็นการตรวจจับวัตถุ มิใช่การมองเห็นที่แท้จริงเหมือนกับตามมนุษย์

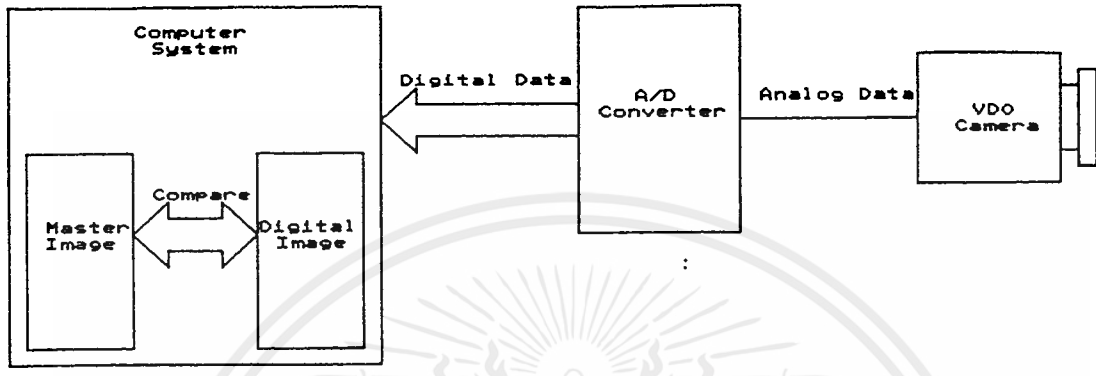


รูปที่ 2.1 การตรวจจับภาพวัตถุด้วยแสง

ระบบการมองภาพที่แท้จริง จะเกี่ยวข้องกับการแปลงภาพ (Image Transformation) การวิเคราะห์ภาพ (Image Analysis) และการเข้าใจภาพ (Image Understanding)

เริ่มจากการเปลี่ยนแปลงภาพที่ปรากฏไปเป็นสัญญาณไฟฟ้า ซึ่งกระทำโดยคอมพิวเตอร์ ระบบนี้จะใช้กล้องโทรทัศน์ กล้องโฟโตไดโอด หรือกล้องซีซีดี (Charge-Coupled Device) เพื่อทำหน้าที่แทนตา ดังแสดงในรูปที่ 2.2 เมื่อกล้องสแกนไปบนฉากเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ปรากฏจะถูกแปลงเป็นสัญญาณดิจิทัลด้วยเอ็ดจคอนเวอร์เตอร์ แล้วเก็บไว้ในหน่วยความจำด้วยรูปแบบของภาพดิจิทัล (Digital Image) จากนั้นคอมพิวเตอร์ก็จะทำการวิเคราะห์ภาพดิจิทัลนั้นว่าภาพนั้นเป็นภาพของสิ่งใด



รูปที่ 2.2 ระบบตาหุ่นยนต์ที่ใช้กล้องโทรทัศน์

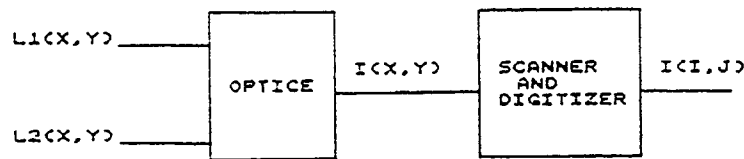
การวิเคราะห์ภาพที่ได้จากการเปรียบเทียบความเข้มแสง จะใช้ข้อมูลของขอบภาพ พื้นที่สีและลักษณะพื้นผิวเปรียบเทียบกับภาพแม่แบบ (Image Templates) เพื่อแบ่งแยก และจดจำภาพวัตถุต่าง ๆ สัญญาณภาพดิจิทัลเหล่านี้จะถูกเก็บอยู่ในหน่วยความจำของคอมพิวเตอร์ เพื่อทำการเปรียบเทียบกับภาพที่มองเห็นอยู่ จึงอาจมีปัญหาดังจำนวนหน่วยความจำที่เก็บภาพในลักษณะฉากต่าง ๆ กันไปก็อาจมีไม่เพียงพอ

ระบบที่มีความชาญฉลาดจะทำการเปรียบเทียบภาพดิจิทัลที่มองเห็นใหม่ กับภาพดิจิทัลในหน่วยความจำนี้และทำความเข้าใจภาพ แล้วจึงแก้ปัญหาของสภาพงานขณะนั้นได้ในที่สุด

การแปลงภาพจริง เป็นข้อมูลภาพ

เราพิจารณาแหล่งพื้นฐานของข้อมูลภาพ 2 ลักษณะ กรณีแรกคือการสร้างสัญญาณของสิ่งมีชีวิตจากภาพจริง และ ในกรณีของการคืนภาพเดิมจากข้อมูลที่เก็บไว้ในตัวกลางทางฟิสิกส์ ถ้าเราสังเกตจากรูป 2.3 ที่ขั้นตอน Optice ซึ่งจะสร้างสัญญาณภาพ $I(x,y)$ ป้อนให้กับส่วนของการสแกน และ digitizer เป็นส่วนของการนำข้อมูลของภาพเข้าสู่คอมพิวเตอร์ ภาพอนาล็อก $L_1(x_1,y_1)$ หรือ $L_2(x_2,y_2)$ ถูกเปลี่ยน
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นภาพสัญญาณดิจิทัล $I(x,y)$ และ บางครั้งเรียกว่าเป็นส่วนเรตินาของคอมพิวเตอร์ เราจะอธิบายถึง hardware ในส่วนที่จำเป็นสำหรับการให้รูปภาพเป็นอินพุทของคอมพิวเตอร์



รูปที่ 2.3 เป็นการแสดงตัวสแกนและ Digitizer ของอินพุท

เตอร์เรากำหนดให้ภาพดิจิทัล $I(i,j)$ เป็นฟังก์ชันจริง 2 แกนของตัวแปร discrete ค่า i,j ซึ่งเป็นค่าอ้างอิงแสงและเงา สี่ระดับที่ปรากฏบนภาพ ค่าเหล่านี้สัมพันธ์กับอินพุท $L_1(x,y)$ หรือ $L_2(x,y)$ อย่างใดอย่างหนึ่งในบริเวณ (x,y) ในบริเวณนี้เป็นค่า discrete ของจุดต่าง ๆ บนภาพและเรียกว่า pixel ในเรื่องของ hardware มีการทำงานสองขั้นตอน คือส่วนของการสุ่มข้อมูลจากภาพเป็น pixels และการทำ quantize ของระดับสีเทาขนาดแนวกว้างยาวของภาพจะเป็น 128×128 , 256×256 และ 512×512 แต่เราอาจขยายขึ้นถึง 2048×2048 ซึ่งมักจะไม่ใช่ในงานปกติของ remote sensing ระดับของ gray level นั้นมีระดับความแตกต่างกันเพียง 64 ระดับก็สามารถให้ความสมบูรณ์พอเพียง ถึงแม้ว่าในบางครั้งมีการใช้ถึง 256 ระดับ

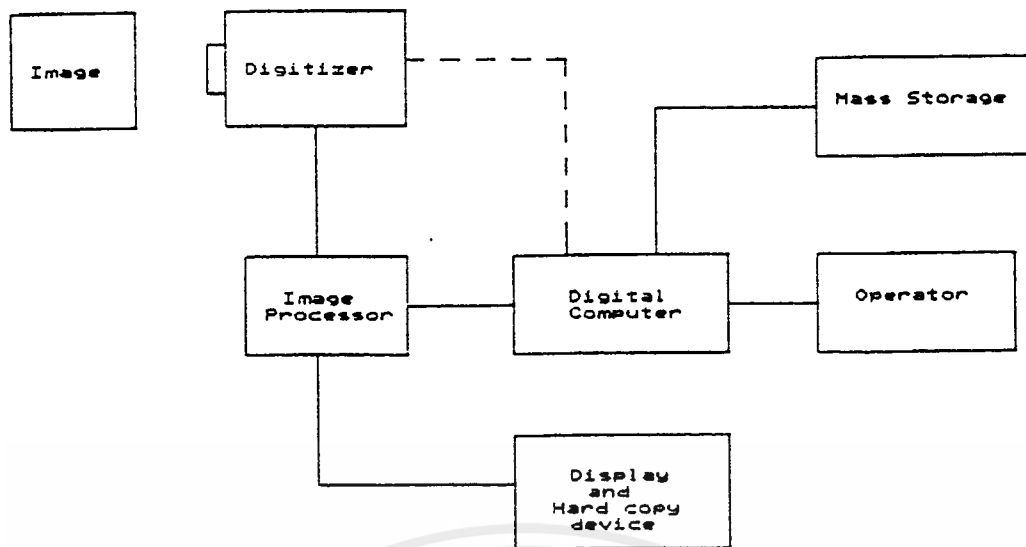
ระบบ Digital Image Processing

ส่วนประกอบพื้นฐานของระบบ Digital Image Processing แสดงไว้ในรูปที่ 2.4 ซึ่งแสดงเป็น Block diagram และจะอธิบายได้ดังต่อไปนี้

IMAGE PROCESSORS

digital image processor เป็นหัวใจของระบบ image processing ซึ่งระบบ image processor ประกอบด้วย ชุดของ hardware ซึ่งทำหน้าที่พื้นฐาน 4 อย่าง คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 Digital Image Processing block diagram

1. การรับภาพ จะมีสัญญาณ T.V. เป็นอินพุตและจะเปลี่ยนสัญญาณนี้ไปเป็นสัญญาณดิจิทัล ปัจจุบันนี้ อิมเมจโพรเซสซิ่ง (image processing) โดยมากสามารถรับภาพจาก T.V. ได้ใน 1 frame-time (ใช้เวลา 1/30 วินาที) เพราะว่าภาพ ที่รับเข้ามามักจะเป็น "frame grabber"
2. การเก็บภาพ เราเรียกว่า frame buffer เป็นหน่วยความจำที่สามารถนำภาพเข้ามาเก็บไว้ได้และยังสามารถที่จะโหลด หรือ อ่านจากสัญญาณ T.V. (เพื่อที่จะได้ภาพต่อวินาที)
3. การโพรเซสซิ่งในระดับล่าง เป็นการนำภาพที่ถ่ายเข้ามาทำการประมวลผล และการกระทำทางลอจิก (Logic) ซึ่งตัวกระทำนี้เราเรียกว่า Arithmetic - Logic Unit (ALU) ซึ่ง ALU เป็นอุปกรณ์ทางฮาร์ดแวร์ ที่ถูกออกแบบมาเป็นพิเศษ ที่ใช้ในความเร็วสูงและเป็นการโพรเซสซิ่ง ในแบบขนาน
4. การแสดงผล จะทำการอ่านจากหน่วยความจำที่เก็บภาพไว้แล้ว และทำการเปลี่ยนข้อมูลดิจิทัลไปเป็นสัญญาณอนาล็อก และเอาท์พุท ของสัญญาณนี้จะส่งไปที่ T.V. มอนิเตอร์ หรือ อุปกรณ์ วิดีโอ อื่นๆ

DIGITIZER

digitizer จะทำการเปลี่ยนภาพเป็นความเข้มของแสง (เป็นตัวเลข) ตามความเหมาะสมของสัญญาณอินพุตที่รับเข้ามา ซึ่งเป็นดิจิทัล โดยมากอุปกรณ์เหล่านี้จะเป็นเอกสารเป็นเอกสารทสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

micro - densitometers, flying spot scanners, image dissectors, vidicon cameras and photosensitive solid-state arrays. ส่วน Vidicon cameras สามารถรับภาพของธรรมชาติ เข้ามาได้โดยขึ้นอยู่กับความเข้มของแสงที่ทำการรับเข้ามาได้

DIGITAL COMPUTERS

อิมเมจ โพรเซสซิ่ง ถูกจัดมีความสามารถในการ โพรเซสซิ่ง ภายในดั่งนั้นสิ่งหนึ่งมักจะพบว่า อิมเมจ โพรเซสซิ่ง จะเชื่อมต่อกับทุก ๆ ส่วนของ คอมพิวเตอร์ซึ่งจัดให้แต่ละ โปรแกรมสามารถเข้าไปประมวลผลข้อมูลของอิมเมจได้ ระบบคอมพิวเตอร์ ที่ใช้ในการโพรเซสซิ่งของข้อมูลของอิมเมจนั้น มีตั้งแต่เครื่องไมโครคอมพิวเตอร์ไปจนถึง ระบบคอมพิวเตอร์ขนาดใหญ่ ซึ่งมีความสามารถในการคำนวณบนพื้นที่ของอิมเมจขนาดใหญ่ ตัวแปรสำคัญที่มีอิทธิพลต่อโครงสร้างของคอมพิวเตอร์ สำหรับใช้ในการโพรเซสซิ่งของข้อมูลอิมเมจ นั้นจะเป็นจุดมุ่งหมายในการเอาไปประยุกต์ใช้มากกว่า และความต้องการของการใช้งานของข้อมูลการนำไปใช้กับเครื่อง ไมโครคอมพิวเตอร์ หรือ เครื่องมินิคอมพิวเตอร์ ก็เพียงพอแล้ว

STORAGE DEVICES

ดิจิตอล อิมเมจ (digital image) ซึ่งมีขนาด 512×512 พิกเซล ซึ่งแต่ละพิกเซล ต้องการพื้นที่ 8 บิต ดังนั้นภาพทั้งภาพจะใช้เนื้อที่ประมาณ 0.25 เมกะไบต์ การเก็บภาพส่วนใหญ่จะมีอุปกรณ์อยู่ 3 ชนิด คือ magnetic disk, magnetic tapes และ optical disk

magnetic disks จะมีความจุประมาณ 700 เมกะไบต์ หรือมากกว่านี้ ดังนั้น 700 เมกะไบต์ ของดิสก์ควรจะสามารถเก็บภาพได้ 2800 ภาพ (ภาพมีขนาด 512×512)

magnetic tapes ที่มีความเข้มสูง (6400 ไบต์ต่อนิ้ว) สามารถเก็บภาพ 1 ภาพ โดยใช้พื้นที่ใน เทป ประมาณ 4 ฟุต

optical disks ซึ่งเป็นการนำเทคโนโลยีการอ่านและเขียนโดย เลเซอร์ความจุในการเก็บของ single optical disk platter มีประมาณ 4 จิกะไบต์ หรือเก็บเป็นภาพได้ประมาณ 16,000 ภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DISPLAY DEVICES

มอนิเตอร์ แบบ โมโนโครม และ คัลเลอร์ เป็นอุปกรณ์แสดงผลที่ใช้ในระบบ อิมเมจโปรเซสซึ่ง ในปัจจุบัน มอนิเตอร์จะถูกขับโดยเอาต์พุตของ image display module ใน อิมเมจโปรเซสซึ่ง ซึ่งจะส่งสัญญาณออกที่เอาต์พุตเป็นสัญญาณภาพไปที่ จอภาพ ของมอนิเตอร์ ส่วนเครื่องแสดงผลอย่างอื่นๆ จะรวมถึงจอภาพ CRT และ Printer ด้วย

ระบบจอภาพแบบ CRT ตำแหน่งของ อิเล็กเมนต์ ต่างๆ ทางแนวนอนและแนวตั้ง ของภาพเป็นการเปลี่ยนแปลง โวลเตจที่นำไปจ่ายให้กับ โยคคอยล์(Yoke coil) เพื่อ ควบคุมการเบี่ยงเบน Electro beam ของ CRT ทั้งแนวตั้งและแนวนอน

ระบบ Printer เป็นอุปกรณ์แสดงผลเบื้องต้น สำหรับงานอิมเมจที่มีความละเอียดต่ำ ๆ ระดับ Gray level ของจุดต่าง ๆ ที่พิมพ์ออกมาสามารถควบคุมจำนวนและความหนาแน่นของตัวอักษรที่จุดต่างๆได้

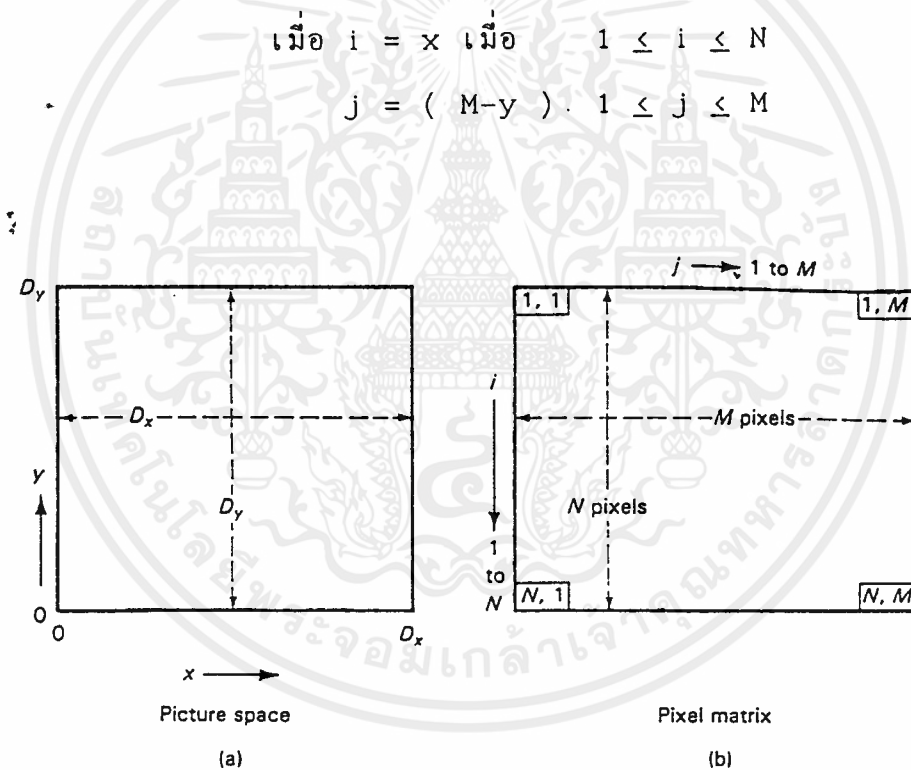


บทที่ 3

การประมวลผลภาพเบื้องต้น

3.1 Pixel

อิมเมจสามารถอธิบายได้โดยค่าของพิกเซลที่มีรูปร่างเป็นเมทริกซ์ (matrix) ขนาด $N \times M$ (ค่าอิลิเมนต์ $P(i,j)$ มีค่าเป็นบวก) ซึ่งแสดงถึงความเข้มของแสงของฟลักซ์ (Flux) บนรูปภาพอิลิเมนต์ที่ (x,y) จะแทนโดยพิกเซลดังแสดงในรูปที่ 3-1 ซึ่งความสัมพันธ์ระหว่างพิกเจอร์อิลิเมนต์ (picture element) และ พิกเซลเมทริกซ์ (pixel matrix) จุดกำเนิดของภาพ (picture) และ เมทริกซ์ (matrix) จะแตกต่างกัน x และ y เป็นคู่ลำดับ (coordinate) ในภาพ (picture) ที่มีจุดเริ่มต้นทางมุมด้านล่างซ้าย ขณะที่เมทริกซ์ (matrix) มีจำนวนพิกเซลที่มุมบนด้านซ้าย

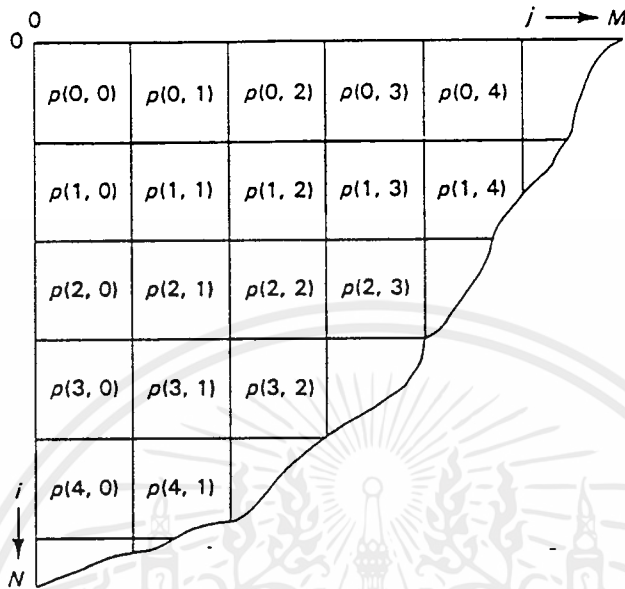


รูปที่ 3-1 Relationship of picture element (a) and corresponding pixel distribution in matrix (b)

ค่าจำนวน (numerical value) หรือขนาด (magnitude) ของพิกเซลแสดงให้ด้วยค่าเฉลี่ยความเข้มแสงบนพื้นที่พิกเซลอิลิเมนต์ (picture element) ที่แทนด้วยพิกเซลค่าของพิกเซล $P_{i,j}$ มีช่วงที่ 0 ถึง 1
 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



$x = Dx/N$ increment
 $y = Dy/M$ increment
 N = maximum number of pixels in a column
 M = maximum number of pixels in a row



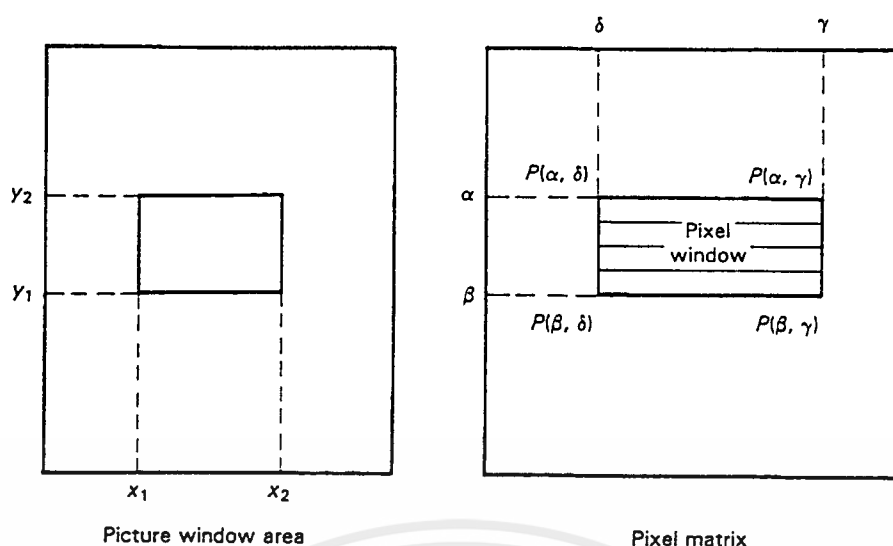
รูปที่ 3-2 Pixel indexing in an image matrix

3.2 Window

บริเวณย่อย (Subregion) ของภาพมักเรียกว่า "window" และถูกกำหนดโดยมุม 4 มุม และค่าของตำแหน่งพิกเซล $P(x, y)$, $P(x, y)$, $P(x, y)$, $P(x, y)$ ดังแสดงในรูป 3-3

3.3 Pixel location

คุณสมบัติของพิกเซลในรูปแบบพื้นฐานจะเหมือนกันหมด โดยคู่ลำดับ (coordinate) ของพิกเซลจะอยู่ในระนาบขนาด $N \times M$ ซึ่งแทนเป็นรูปภาพ พิกเซลที่ตำแหน่ง (n, m) จะมีค่าอยู่หนึ่งค่า ซึ่งถูกแทนด้วยค่าเฉลี่ยของแสงที่ตกกระทบบนพื้นที่ของภาพ (picture) แทนด้วยพิกเซล



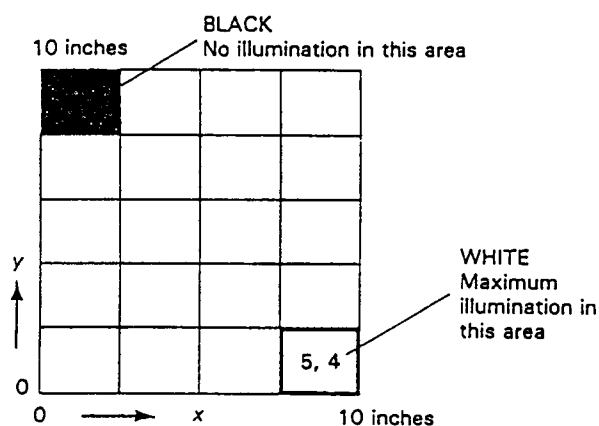
รูปที่ 3-3 Image window and corresponding pixel window

ลองพิจารณาที่ ที่ซึ่งไม่มีแสงตกกระทบบนบริเวณส่วนบนของภาพ และมีแสงที่มีความสว่างมากตกกระทบบนบริเวณส่วนล่าง ขนาด $10" \times 10"$ ของภาพ (picture) ดังรูป 3-4 ในระบบของไบนารี จะใช้การแทนค่าของแสงสว่างดังนี้ : บริเวณที่ไม่มีแสงสว่างจะแทนด้วยค่า "0" และบริเวณที่มีแสงสว่างจะแทนด้วย "1" ภาพจะแสดงโดยเมทริกซ์ (matrix) ขนาด 5×4 คือ มี 5 แถว 4 หลัก แต่ละส่วนของภาพ (picture element) มีความกว้าง $2.5"$ และ สูง $2"$ จะถูกกำหนดเป็นตัวเลข ซึ่งขึ้นอยู่กับค่าเฉลี่ยของแสงที่ตกกระทบบนพื้นที่ส่วนนั้น

พื้นที่ขนาด $2.5" \times 2"$ ที่อยู่ในส่วนบนทางมุมซ้ายมือของภาพ จะแทนด้วยตำแหน่ง (1,1) ในเมทริกซ์ (matrix) ขนาด 5×4 จะมีค่า "0" อยู่ แสดงว่าบริเวณนี้ไม่มีแสงตกกระทบบนพื้นที่ขนาด $2.5" \times 2"$ ที่อยู่มุมซ้ายส่วนล่างของภาพ ซึ่งอยู่ที่หลักที่ 4 แถวที่ 5 แทนโดยตำแหน่ง (5,4) จะมีค่าเป็น "1" อยู่แสดงว่าบริเวณนี้มีแสงสว่างสูงสุด จุดสังเกตค่าของพิกเซลที่ตำแหน่ง (1,1) จะเป็น "0" และค่าของพิกเซลที่ตำแหน่ง (5,4) จะเป็น "15" ถ้ามีระดับสีเทา (gray scale) เท่ากับ 16 ระดับ ซึ่งนำมาใช้แทนในระบบไบนารี

ไม่มีข้อมูลในค่าของพิกเซลที่เป็นค่ากลาง ๆ และ ระบบที่ออกแบบไว้จะต้องแสดงค่า "Threshole" (Threshold value) ของแสงสว่าง ซึ่งค่าจะถูกเปลี่ยนจาก 0 ไป 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



0	?	?	?
?	?	?	?
?	?	?	?
?	?	?	?
?	?	?	1

All elements would contain either a zero or a one.

รูปที่ 3-4 Nonuniform lighting on a surface (a) and resultant image pixel values for the surface (b)

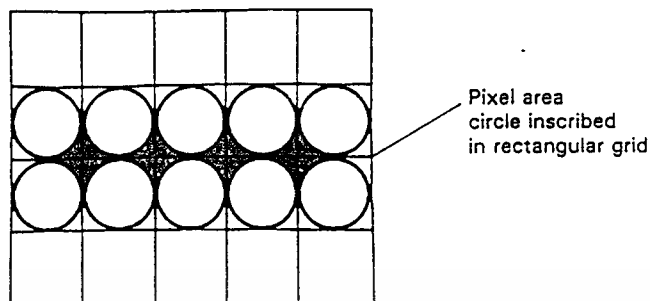
ในตัวอย่าง ส่วนของภาพ (picture element) เป็นรูปสี่เหลี่ยมผืนผ้า พื้นที่บนภาพจะเป็นรูปสี่เหลี่ยมผืนผ้าหรือวงกลมนั้นขึ้นอยู่กับตัวเซ็นเซอร์ ในกรณีที่เป็นกล้องที่ใช้หลอด (tube camera) กับพื้นที่ตัวเซ็นเซอร์เป็นแบบวงกลม อาจจะมีการเหลื่อมกัน (overlap) ดังแสดงในรูป 3-5

ในกรณีที่ไม่มีการเหลื่อมกัน (overlap) จะมีพื้นที่ของภาพระหว่างวงกลมซึ่งไม่ถูกวัดในกรณีที่มีการเหลื่อมกันของวงกลม จะมีพื้นที่ซึ่งถูกวัดเป็นสองครั้ง

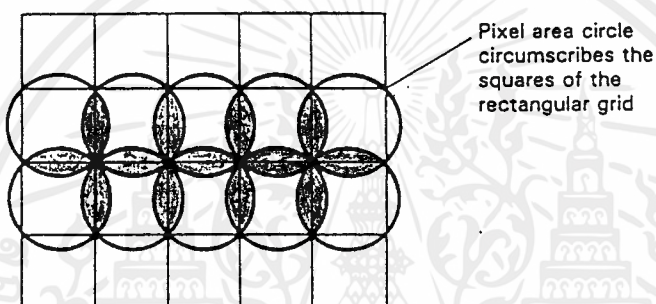
ไม่มีข้อมูลใน pixel map ซึ่งแสดงถึงรูปร่างของพื้นที่ในภาพต้นแบบ (original picture) แทนโดยพิกเซล หรือข้อมูลเกี่ยวกับแสงสว่างใน picture element หรือข้อมูลเกี่ยวกับแสงสว่างใน picture element

3.4 Gray Scale

เพื่อให้จะมีข้อมูลที่อยู่ระหว่างค่าของแสงสว่าง จำเป็นจะต้องเพิ่มจำนวนบิตในการเก็บข้อมูลแทนค่าของพิกเซล ตัวอย่างเช่น ถ้าระดับของแสงสว่าง (illumination) เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(a)



(b)

รูปที่ 3-5 Nonrectangular areas represented by pixel : round nonoverlapping (a) and round overlapping (b)

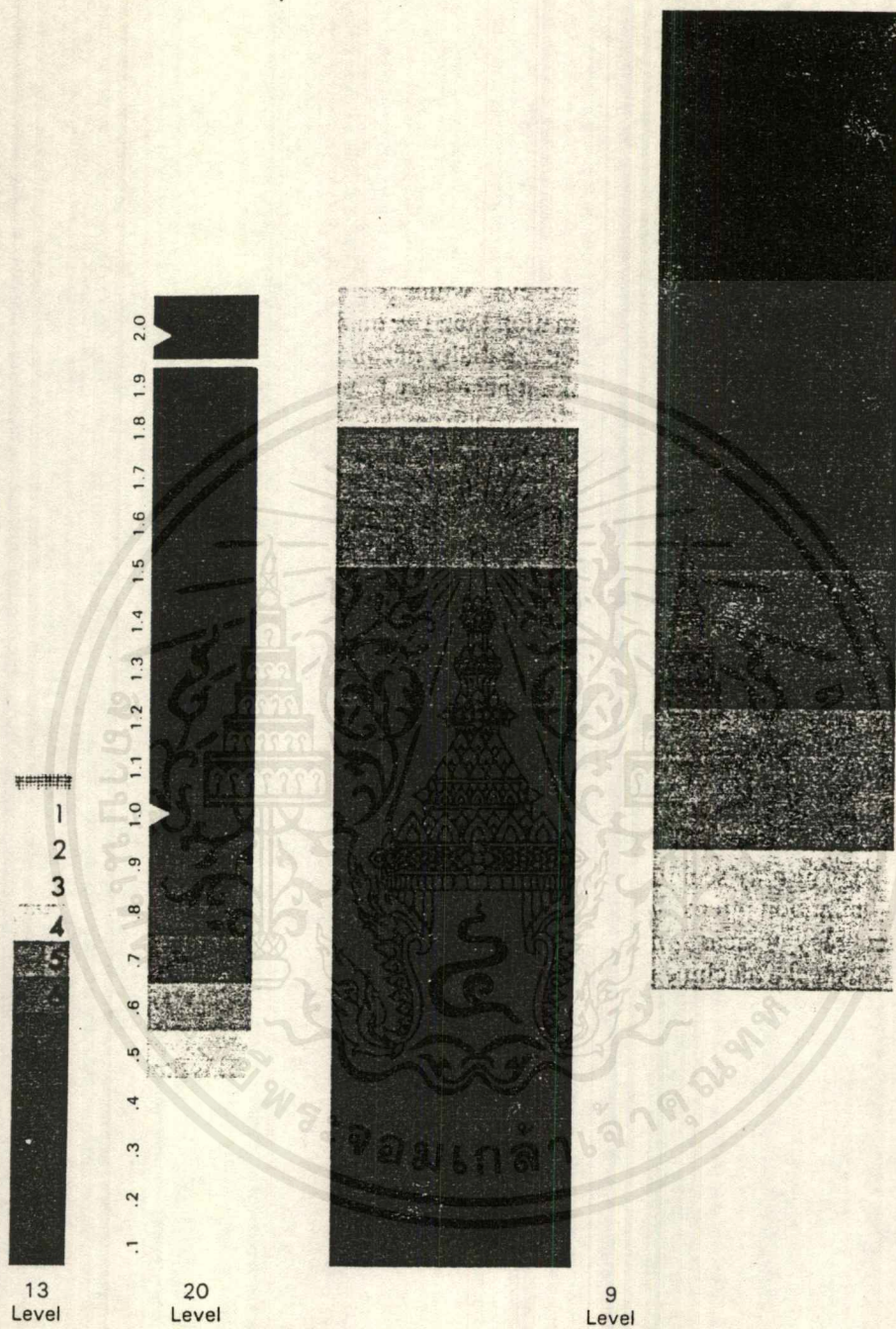
อยู่ 4 ระดับจะต้องการจำนวนบิตในการเก็บข้อมูลเพียงสองบิต จำนวนบิตเพิ่มขึ้นเป็น 4 บิต จะมีระดับแสงสว่างได้ 16 ระดับ 8 บิต จะได้ 256 ระดับ จำนวนของระดับสีเทา (gray level) โดยปกติจะเป็นสองยกกำลัง n (2^n) ค่าที่อยู่ต่ำสุดคือ "0" ซึ่งจะถูกกำหนดให้เป็นสีดำ และ ตัวเลขที่เป็น "1" หรือมากกว่าขึ้นไปจะถูกแทนด้วยระดับสีเทาส่งสุด (maximum gray level) ตัวอย่างเช่นค่า "15" ในระบบ 16 (ระบบที่มีระดับสีเทา 16 ระดับ) จะเป็นค่าสำหรับสีขาว การกำหนดค่าของพิกเซล ปกติมักจะเป็นเลขจำนวนเต็ม (integer)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GRAY SCALE			GRAY VALUE RANGE
2^1	2	ค่า	0, 1
2^2	4	ค่า	0 ถึง 7
2^4	16	ค่า	0 ถึง 16
2^8	256	ค่า	0 ถึง 256

การแทนค่าความสว่างของแสงแบบไบนารีที่นิยมใช้กันอยู่ทุกวันนี้ ในระบบการมองเห็น (vision system) เพราะมีข้อดีคือ ทำตัวเซ็นเซอร์ง่าย ข้อมูลมีความถูกต้องสูง การประมวลผล และ การเก็บข้อมูล ซึ่งในปัจจุบัน ไมโครโปรเซสเซอร์มีการประมวลผลแบบใช้ข้อมูลมากกว่า 8 บิตแล้ว ดังนั้นระดับสีเทา (gray level) ที่มีค่าเท่ากับ 16, 64 และ 256 จึงเป็นเรื่องธรรมดาในเวลานี้ได้จำกัดค่าไม่ให้เกิน 256 ระดับ เพราะว่า 64 และ 256 ระดับ ก็ให้รายละเอียดมากพอที่ตามนุษย์จะสามารถเห็น และเพียงพอที่จะนำมาใช้ในงานอุตสาหกรรมได้อย่างพอเหมาะความสามารถของตามนุษย์ในการบอกความแตกต่างระหว่างความเข้มของแสงได้นั้น อยู่ในระดับ 40 แต่สามารถบอกความแตกต่างได้เด่นชัดเพียง 10-15 ระดับเท่านั้น ระดับสีเทามีค่าเท่ากับ 16 เป็นสเกลที่ค่าสำหรับตามนุษย์ที่จะแยกแยะได้ ขณะที่ 64 และ 256 ระดับนั้น จะให้การแยกแยะได้มากกว่าตามนุษย์ที่จะแยกแยะได้

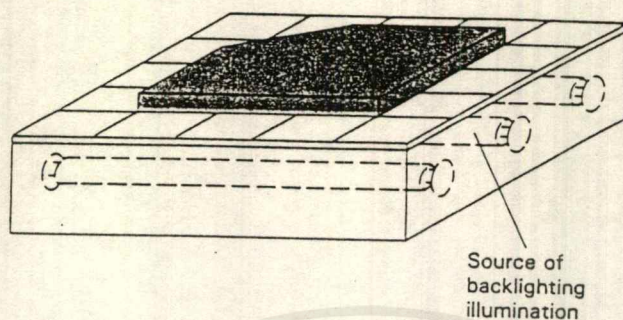
ระบบอาจจะมีความสามารถแบ่งได้ 256 ระดับ แต่ต้องประกอบรวมกับการประยุกต์ใช้งานอาจจะทำให้จะต้องใช้รายละเอียดของจำนวนระดับสีเทา ความถูกต้องหรือความต้องการที่จะพัฒนาให้ดีขึ้น โดยใช้ $1/9$, $1/13$ หรือ $1/20$ Subpixel resolution technique และระบบ Gray level มีผลต่ออิมเมจ โดยบริเวณที่มีแสงมากจะมีลักษณะเด่น (feature) ขณะที่แยกเฉพาะส่วนอธิบาย โดยทั่วไปการเพิ่มขึ้นของจำนวน gray level จะช่วยให้คุณภาพของอิมเมจดีขึ้นและจัดให้เหมาะสมที่จะทำให้ gray region ดีขึ้น ถ้าข้อมูล (information) เป็นฐานข้อมูล (data base) ขณะที่ทวิคูณระดับ gray scale image การจับ (acquisition) ภาพใน binary mode จะลดข้อมูลในการเก็บ แต่มีขีดจำกัดที่สามารถใช้เทคนิคการขยาย gray scale (gray scale expansion technique) ในการวิเคราะห์ประมวลผลของการกระทำ (operation) การเพิ่มขนาดของพิกเซลอิมเมจจากค่าเล็ก ๆ เช่น 32×32 ไปเป็น 250×250 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



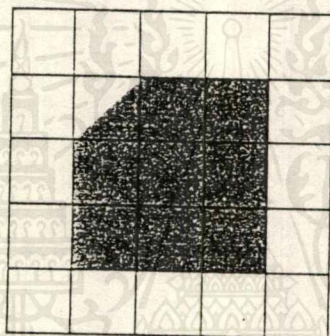
รูปที่ 3-6 Sample gray scales

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเพิ่มความละเอียด และระดับรายละเอียดของอิมเมจนี้เป็นข้อแตกต่างจากการ "zoom" ซึ่งเป็นการเพิ่มขนาดของพิกเซล



(a)



(b)

15	15	15	15	15
15	7	0	0	15
15	0	0	0	15
15	0	0	0	15
15	15	15	15	15

(c)

รูปที่ 3-7 Object on light table (a), with top view (b), and corresponding image matrix (c) where black pixel is represented by zero and white pixel is represented by fifteen

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในระบบ gray scale จะให้ค่ากลาง ๆ (means) ของลักษณะเฉพาะ บอกความแตกต่างระดับของความเข้มแสงในรูปภาพ (picture) และให้ค่ากลางของเป้าหมาย subpixel ที่เที่ยงตรงในการวัดขนาดของนิพจน์ (item) back lighting จะถูกนำมาใช้ตรวจจับในส่วนของการจักรกับระบบการมองเห็น (vision system) สำหรับช่วงทั้งหมด (overall length) ในรูป 3-7

ส่วนของมุมใน diagram จะอยู่ในตำแหน่ง (2,1), (2,4), (5,1), (5,5) ในการวิเคราะห์แบบรังผึ้ง (cellular analysis) พื้นที่ที่แทนโดยพิกเซล (2,5) จะเป็นบริเวณมืด และจะมีค่าเป็น "0" สำหรับพื้นที่ซึ่งไม่มีความเข้มของแสง พื้นที่ที่แทนโดยพิกเซล (5,2) จะมีวัตถุอยู่ครึ่งหนึ่ง ดังนั้น ค่าเฉลี่ยของแสงเหนือพื้นที่ดังกล่าวจึงเท่ากับ $(0+15)/2$ หรือ 7.5 พื้นที่ที่สว่างทั้งหมดจะมีค่าของพิกเซลเท่ากับ 15 ค่าความเข้มของแสงสว่างสูงสุดจะอยู่ในช่วง 0-15 ในระบบที่เป็น 16 ระดับ

ขณะที่ค่าของพิกเซลจะต้องเป็นจำนวนเต็ม ค่า 7.5 จำเป็นต้อง modify การ modify จำเป็นจะต้องมีกฎเกณฑ์ ในระบบใด ๆ ซึ่งจะหาว่า จะเปลี่ยน fractional value ไปเป็นจำนวนเต็ม (integer) ได้อย่างไร ตัวอย่างเช่น กฎเกณฑ์ทั่วไปจะปัดเศษขึ้นไปยังเลขจำนวนเต็มที่อยู่ถัดไป ถ้า fractional value มีค่าอยู่ระหว่าง 0.5 และ 1.0 ดังนั้นถ้าวัดค่าได้เป็น 7.5 ค่าที่ใช้จะเป็น 8 ขณะที่ในรูป 3-7 ไม่ได้ปัดเศษขึ้น ถ้าวัดค่าได้ 6.6 ค่าที่ใช้ คือ 7 ผลของ fractional value ถูกปัดเศษขึ้นมีความสำคัญ เพราะมันมีผลกระทบต่อ tolerance และเป็นสิ่งสำคัญมากที่จะต้องรู้ว่า ค่าใดที่กำลังถูกใช้ในระบบ

3.5 Quantizing Error

การปัดเศษค่า gray scale ไปเป็นจำนวนเต็ม จะได้ผลในความไม่แน่นอนของค่าที่ถูกต้อง ความไม่แน่นอนจะเป็นฟังก์ชันของกฎเกณฑ์การปัดเศษ

กฎเกณฑ์ซึ่งบอกว่า ค่าใดที่มีค่ามากกว่าค่าจำนวนเต็ม จะถูกปัดขึ้นเป็นจำนวนเต็มถัดไป ดังแสดงในรูป 3-8 สำหรับระบบที่มีค่า gray level เท่ากับ 16 การปัดเศษนี้ จะให้ผลในความไม่แน่นอนเชิงลบ (negative uncertainty)

ขนาด (dimension) และความกว้าง (tolerance) สามารถหาได้โดยการใช้ subpixel resolution technique และความรู้ในเรื่องการปัดเศษ ในตัวอย่างต่อไปนี้ ขนาด x (x dimension) ในรูป 3-9 ถูกแทนโดย 4 ช่องพิกเซล และ "1" ส่วนย่อยพิกเซลขนาด y (y dimension) ถูกแทนโดย 1 ช่องพิกเซล เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

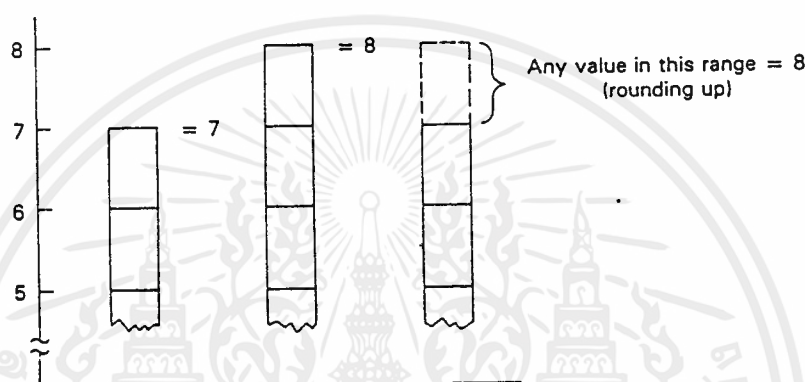
$$x = 4 \text{ pixel} \times 0.100 \text{ inch/pixel (on object)}$$

$$1 \text{ partial pixel} \times 12/16 \times 0.100 \text{ inch/pixel}$$

12/16 factor คือ = proportion of gray levels in partial pixel/total gray levels.

$$x = 0.100 \text{ inch} + 0.075 \text{ inch} = 0.475 \text{ inch}$$

$$y = 1 \text{ pixel} \times 0.100 \text{ inch/pixel} = 0.100 \text{ inch}$$



รูปที่ 3-8 Quantizing error (rounding up)

tolerance หาได้จากกฎการปิดเศษ ค่าพิกเซลของ "0" แทนด้วยค่าสูงสุดของแสงสว่าง ค่าพิกเซลของ 11 แทนด้วยค่าของแสงสว่างจาก $10.x$ ถึง 10.0 ซึ่ง x เป็นค่าจำนวนเต็มที่มีมากกว่าหรือเท่ากับ "1" ค่าพิกเซลของ 15 ถูกแทนด้วยค่าแสงสว่างจาก $14.x$ ถึง 15.0 ซึ่งพอสรุปได้ดังนี้

ดังนั้นการวัดที่คลาดเคลื่อนในพิกเซลกับการปิดเศษ ทำให้วัตถุไม่สามารถขยายได้มากกว่าที่แสดงให้เห็นโดยการคำนวณขนาด (dimension) แต่สามารถทำให้น้อยลงโดยการรวมของขนาดพิกเซล (pixel dimension) แทน โดย 1 gray level ในทิศทาง x จุดสิ้นสุดของสองพิกเซล จะช่วยให้การวัดไม่แน่นอน ขณะที่ในทิศทาง y มีเพียง 1 พิกเซลที่ทำให้เกิดความไม่แน่นอน (Uncertainty)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pixel value in image matrix		value of illunination represented by pixel value
0	=	0
1	=	$0 < \text{pixel value} \leq 1$
11	=	$0 < \text{pixel value} \leq 11$
15	=	$0 < \text{pixel value} \leq 15$

Tolerance or uncertainty of dimension

$$x = +0.0 \text{ inch} - 2 \text{ pixel} \times \frac{1}{16} \times 0.100 \text{ inch/pixel}$$
$$= -0.0125 \text{ inch}$$
$$y = +0.0 \text{ inch} - 1 \text{ pixel} \times \frac{1}{16} \times 0.100 \text{ inch/pixel}$$
$$= -0.00625 \text{ inch}$$

ขนาดของวัตถุ คือ

$$x = 0.175 + 0.0 \text{ inch} - 0.0125$$
$$y = 0.100 + 0.0 \text{ inch} - 0.00625$$

ค่าซึ่งเป็นค่าพิคเซลถูกเพิ่มขึ้นอีกหนึ่งของเลขจำนวนเต็มในการปัดเศษ และถูกลดค่าเมื่อปัดเศษลง (rounding down) ที่รู้จักกันในนาม "threshold" จะอยู่ในช่วงระหว่างเลขจำนวนเต็มสองจำนวน จะเป็นผลทั้งในทางบวกและทางลบของ tolerance กฎเกณฑ์ ซึ่งค่ามากกว่า 6.5 จะปัดเศษขึ้นเป็น 7 ค่าระหว่าง 0.6 และ 6.5 จะปัดเศษลงเป็น 6.0 จะส่งผลทั้งทางบวกและทางลบของบริเวณไม่แน่นอน (uncertainty zone) ซึ่งเท่ากับ

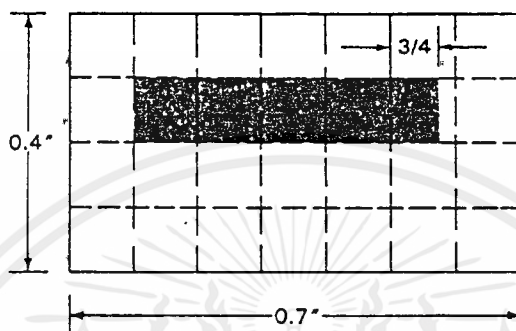
1

$$2 \times (\text{max No. gray level})$$

จำนวน gray level ที่สูง จะทำให้เกิด quantizing error น้อย เอกสารนี้เป็นเอกสารทสรงานวิชาสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

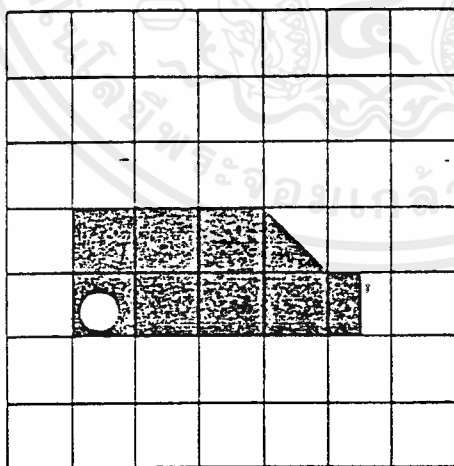
0	0	0	0	0	0	0
0	15	15	15	15	11	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

(a)



(b)

รูปที่ 3-9 Dimensional Determination from image matrix data
(a) of regular shaped object on backlighted inspection
table (b).



(a)

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	15	15	15	7	0	0
0	7	15	15	15	7	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

(b)

รูปที่ 3-10 Irregular shaped object on light table (a)

with corresponding image pixel values (b).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

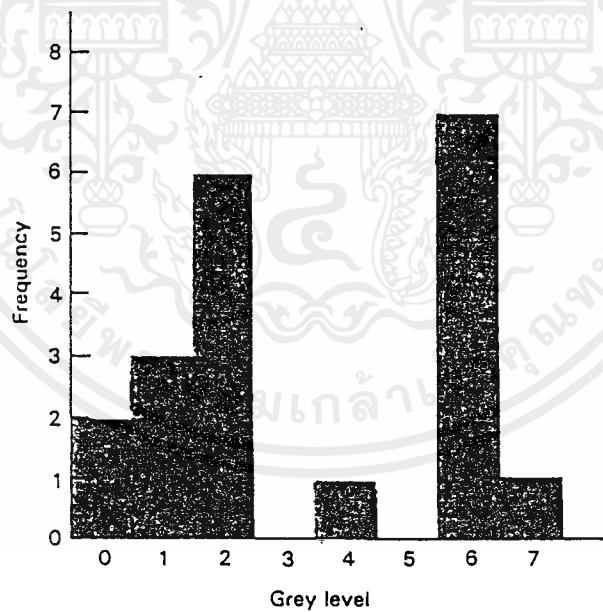
ที่แต่ละ gray level ที่เกิดขึ้นในกรอบ (frame)

ขนาด (magnitude) ของ histogram ที่ค่าของพิกเซล คือความน่าจะเป็นของค่าสีเทา (gray value) เกิดขึ้นในส่วนของภาพ (picture element) ในกรอบ (frame) ไม่มีข้อมูลที่ทำให้เกี่ยวกับตำแหน่งของพิกเซล ความน่าจะเป็นของพิกเซลค่า b เกิดขึ้น ณ ตำแหน่งจุด (x,y) ดังรูป 3-11 และมีสมการ

$$P(b) = \text{at any point } (x,y) \text{ in the picture} = \frac{\text{Value } b}{\text{total No. of pixels}}$$

ถ้า $b = 6$ ระดับสีเทาและค่าของ histogram ที่ 6 เท่ากับ 7 คือ

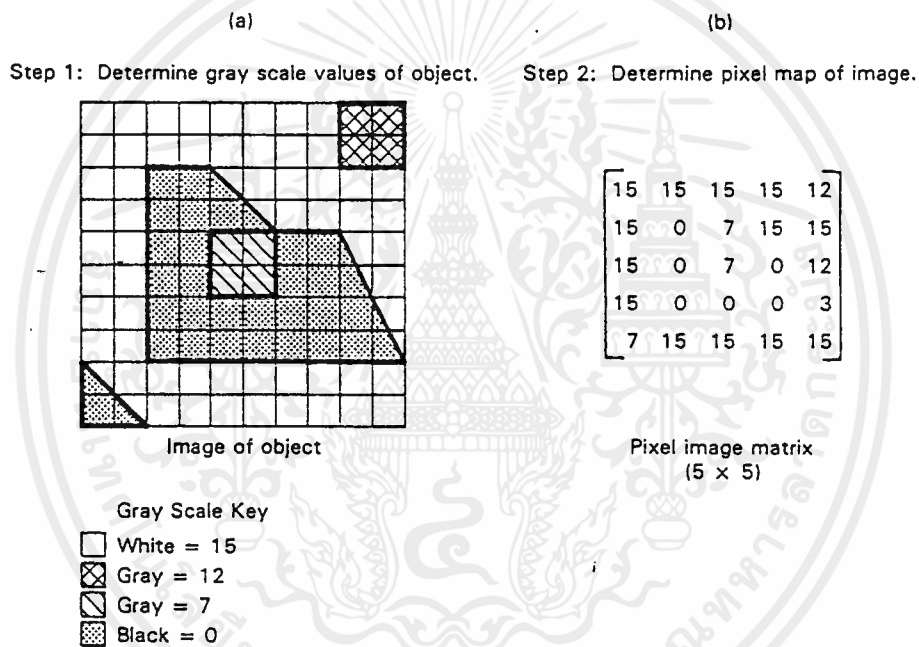
$$P(6) = 7/20 = 0.35$$



รูปที่ 3-11 Histogram for an eight gray level four by five array image.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปร่างของ histogram ที่เตรียมข้อมูลบน คุณลักษณะ (characteristic) ของภาพตัวอย่าง histogram ที่แคบบอกถึง contrast ในอิมเมจหรือรายละเอียดของค่าพิกเซลอาจแทนด้วย unique characteristic ของนิพจน์ (item) ในภาพ (picture) เช่นมีส่วนของรู histogram มีประโยชน์ในการหาค่า threshold ในการเปลี่ยน gray scale image ไปเป็นไบนารีอิมเมจ หรือ modify ส่วนของ gray scale spectrum



รูปที่ 3-12 Histogram construction : the pixel values of the object (a) determined in step one , and the pixel image matrix (b) determined in step two. A five by - five matrix is utilized to simplify the illustrative calculation.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.1 Construction

Histogram ของภาพสามารถสร้างขึ้นโดยขั้นตอนดังต่อไปนี้ ดังในรูป 3-12 และ 3-13

(1) หาจำนวนพิกเซลทั้งหมด ซึ่งจะใช้อธิบายภาพ นี่คือการหาโดยอะเรย์ matrix ตัวอย่างนี้ใช้ matrix ขนาด $M \times N$ และจำนวนพิกเซลทั้งหมดเป็นผลจาก $M \times N$ ถ้า

$$M = 10 ; N = 10 ; \text{Total pixel} = 10 \times 10 = 100$$

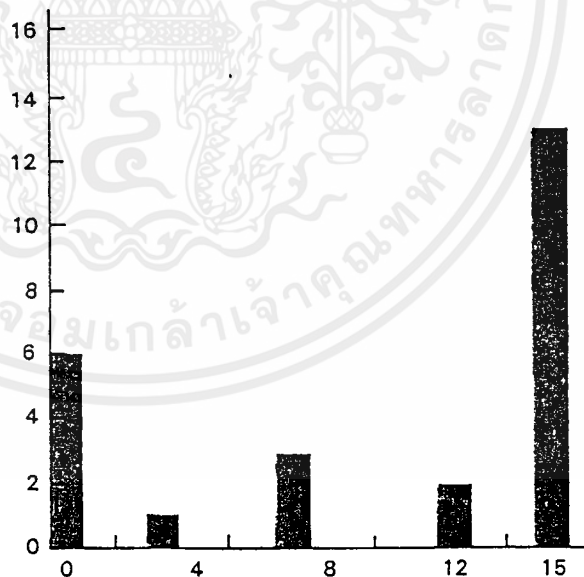
มันอาจเป็นไปได้ว่า M และ N ไม่เท่ากัน มันขึ้นอยู่กับชนิดของกล้อง และ analog-to-digital sampling rate และหน่วยความจำของระบบ ค่าที่สูงของ M และ N จะเพิ่มเป็นความถี่โดยตรง แต่ราคาก็จะเพิ่มขึ้นตามความจุและเวลาการตอบสนอง

(2) โครงสร้างของพิกเซลแมป (pixel map) ของอิมเมจในตัวอย่าง เป็นโครงสร้าง matrix ขนาด 5×5 จำนวนพิกเซลทั้งหมด คือ 25

Step 3: Tabulation of pixel values

Value	Number of Pixels
0	6
3	1
7	3
12	2
15	13
Total	25

Step 4: Graph tabulated data



รูปที่ 3-13 Histogram construction : pixel vales tabulation

(a) of image data in figure 3-12(b) and graphical representations of the tabulated data (b).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(3) ผลตารางของ pixel map นั่นคือ นับความถี่ของค่าพิกเซลแต่ละค่าที่เกิดขึ้น ตัวอย่างเช่น ค่า "0" เกิดขึ้นใน 6 พิกเซล แต่ค่า 3 เกิดขึ้นเพียงครั้งเดียว จำนวนของค่าทั้งหมดในตารางจะต้องเท่ากับ $N \times M$ ในตัวอย่างนี้เท่ากับ 25

(4) สร้าง histogram โดยพรอต bargraph สำหรับค่าแต่ละพิกเซลจาก "0" ไปถึง gray scale ลบหนึ่ง ในตัวอย่าง gray scale ของ 16 ถูกใช้สร้าง pixel map และค่าสูงสุด คือ 15 มันจะแสดงค่าสูงสุดของพิกเซลใน map ไม่มีองค์ประกอบ (factor) histogram จะประกอบด้วย bargraph สำหรับค่าของพิกเซล ตั้งแต่ 0 ถึง 15



บทที่ 4

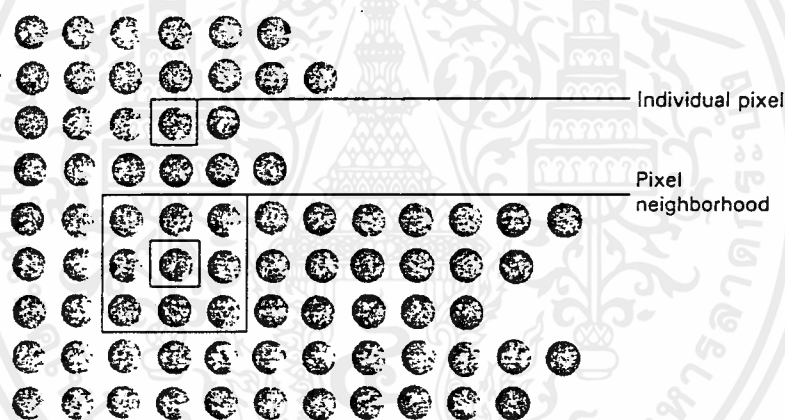
การประมวลผลการมองเห็นของ เครื่องจักร เบื้องต้น

4.1 Introduction

การประมวลผลของข้อมูลในระบบการมองเห็นสามารถแบ่งได้ 2 ประเภท

- (1) Point by point (monadic) การดัดแปลงข้อมูลบนสเกลทั้งหมด
- (2) Multiple point (dyadic) การหา elements ของอะเรย์ตัวใหม่ของอิมเมจ

รูปแบบใหม่ของพิกเซลของอิมเมจเมทริกซ์ (image matrix) จะเป็นฟังก์ชันของค่าของตำแหน่งพิกเซลเดี่ยว ๆ หรือค่าของพิกเซลในบริเวณข้างเคียงของเซลล์ (cell) เดี่ยว ๆ ดังแสดงในรูป 4-1



รูปที่ 4-1 Point and neighborhood function.

Monadic operations เป็นการสร้างอะเรย์ตัวใหม่ โดยโมดิไฟ (modify) ค่าของพิกเซลที่ตำแหน่งบนพื้นฐานเดียวกันกับกฎเกณฑ์ทั้งหมด เอาไปใช้งานทุก ๆ ตำแหน่งในอะเรย์ต้นแบบการประมวลผลการหาค่าของพิกเซลในอะเรย์ ณ ตำแหน่งที่ให้ไว้ โมดิไฟ (modify) โดย simple linear operation หรือ nonlinear operation และใส่ค่าของพิกเซลใหม่ในตำแหน่งที่สอดคล้องกันในอะเรย์ตัวใหม่ การประมวลจะถูกทำซ้ำบนค่าเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

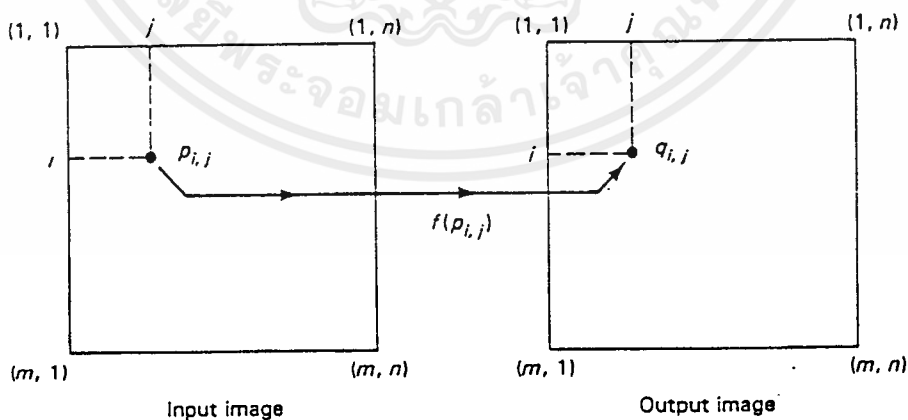
ของพิกเซลถัดไป และทำต่อไปจนครอบคลุมทั้งอะเรย์ (รูป 4-2) ที่แสดงให้เห็นในรูป 4-2 monadic operator เป็นแบบ one-to-one transformation ตัว operator f จะถูกใช้กับทุก ๆ พิกเซลในอิมเมจ หรือ ส่วนของอิมเมจและ output ขึ้นอยู่กับขนาด (magnitue) ขึ้นอยู่กับพิกเซลข้างเคียง พังก์ชัน transform ค่า gray level ของทุก ๆ พิกเซลในกรอบ (frame) และค่าใหม่ถูกกำหนดโดยสมการ

$$g(i,j) = f[p(i,j)]$$

ฟังก์ชัน f สามารถเป็นได้ linear operator หรือ nonlinear operator คณิตศาสตร์ที่ใช้ในการประมวลผล เป็นความสัมพันธ์แบบธรรมชาติ ซึ่งเกี่ยวกับการหาค่าจากผลของสองค่า ผลของอิมเมจ matrix จะมีขนาดเท่ากับอิมเมจต้นแบบ

Dyadic operator ที่จำเป็นใช้ใน procedure ที่เหมือนกัน ซึ่งอะเรย์ของอิมเมจใหม่สามารถสร้างได้ 2 ทาง

- (1) ค่าของแต่ละพิกเซลในอะเรย์ขึ้นอยู่กับ การรวมค่าของพิกเซลที่สอดคล้องกันในแต่ละตำแหน่งใน 2 หรือ มากกว่า image frame
- (2) ค่าพิกเซลในบริเวณที่อยู่ข้างเคียงที่ตำแหน่งของพิกเซล สร้างโดยการรวมกัน



รูปที่ 4-2 Monadic operator.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Procedure หลังสุดจะทำให้สำเร็จได้โดยกรรมวิธี convolution การ 'convolution' จะเกี่ยวข้องกับขนาดของ matrix แต่ 3×3 matrix เป็นขนาดที่ใช้กันโดยทั่วไป เพราะมันใช้เวลาในการคำนวณน้อย และคอมพิวเตอร์สามารถทำ matrix ขนาดใหญ่ ๆ ได้ ในรูปแบบง่าย ๆ การ convolution ของสอง matrices p และ q เกี่ยวกับการคูณของอีลิเมนต์ที่สอดคล้องกันทั้งหมด ผลของผลรวมของ element ที่ถูกคูณโดยค่าคงที่ค่าหนึ่ง (normalization constant) เพื่อรักษาผลของค่าพิกเซลให้อยู่ภายในช่วงที่กำหนด ผลของค่าพิกเซลถูกสอดแทรกในตำแหน่งที่เหมาะสมของอะเรย์ตัวใหม่ และ procedure ก็จะทำซ้ำหลังจากที่คู่ลำดับ (coordinate) ของแต่ละ element จะถูกเลื่อน (shift) ไปอีกหนึ่ง ในทิศทาง M หรือ N อย่างใดอย่างหนึ่ง จนกระทั่งตำแหน่งทั้งหมดในอะเรย์ถูกทำจนหมด

4.2 Monadic one point transformation

Monadic point-by-point operators เป็นวิธีที่ง่ายที่สุดในการประมวลผลภาพ (image processing) กรรมวิธีจะเกี่ยวกับ input image เพียง 1 อิมเมจ (p) ดังแสดงในรูป 4-2 และผลที่ได้ก็เป็น single image output, Point to point operator ซึ่งจะกล่าวในส่วนนี้จะ เป็น identity operator, inverse operator, threshold operator และ combination

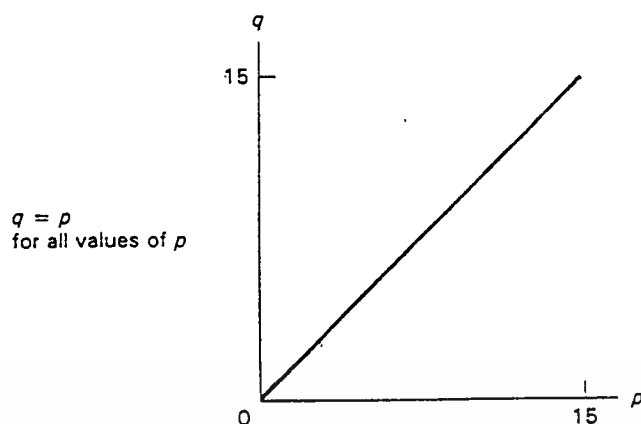
4.2.1 Identity Operator

Operator ตัวนี้จะให้ผลในการสร้างเอาท์พุทอิมเมจ ซึ่งเป็นอิมเมจแบบเอกลักษณ์ (identical) กับอินพุทอิมเมจ ดังแสดงในรูป 4-3 ค่าของแต่ละพิกเซลในอิมเมจที่สอง (q) เป็นเอกลักษณ์ (identical) กับค่าของพิกเซล p ที่สอดคล้องกันในอิมเมจที่หนึ่ง พังก์ชัน f เป็นเส้นตรงที่มีจุดเริ่มต้นที่จุดกำเนิดและต่อออกไปถึงค่าพิกเซลสูงสุดของระบบอิมเมจ procedure พื้นฐานในการประมวลผลภาพ และ operator ตัวอื่น ๆ จะทำให้สำเร็จได้โดย การ modify พังก์ชัน

4.2.2 Inverse Operator

Operator ตัวนี้จะให้ผลในการสร้างเอาท์พุทอิมเมจซึ่งเป็นส่วนกลับ (inverse) กับอินพุทอิมเมจ วิธีการก็คล้ายกับ identity operator ยกเว้นค่าพังก์ชันที่แตกต่างกัน ดังแสดงในรูป 4-4 พังก์ชัน f เป็นเส้นตรงที่มีค่า gray level สูงสุดที่ค่า gray input ต่ำสุด และเท่ากับ "0" ที่ค่า gray input สูงสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(a)

15	15	0	0	2
13	13	15	0	0
0	0	7	14	14
0	1	2	3	4
15	14	13	12	11

 p matrix

(b)

15	15	0	0	2
13	13	15	0	0
0	0	7	14	14
0	1	2	3	4
15	14	13	12	11

 q matrix

(c)

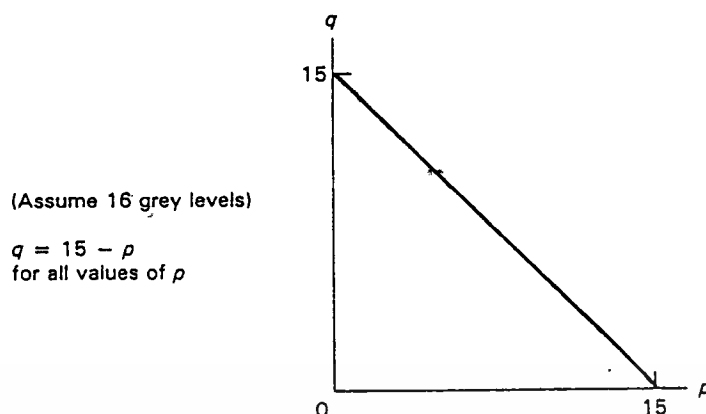
รูปที่ 4-3 Identity operator : function (a), input (b), and output (c).

4.2.3 Theshold Operator

Operator ในขั้นนี้จะให้ผลเป็นไบนารีเอ๊าท์พุทอิมเมจ จาก gray scale output image ซึ่งระดับของการส่งผ่านที่ให้โดยอินพุทพารามิเตอร์ P , ดังแสดงในรูป 4-5 และรู้จักกันในนาม "threshold" ค่าของพิกเซลทั้งหมดที่ต่ำกว่า P , จะถูกเปลี่ยนเป็น "1" operator ตัวนี้สามารถใช้ในการหา spatial information จาก histogram โดยทำ procedure ซ้ำโดยใช้ค่า theshold ที่แตกต่างกัน

4.2.4 Other Operator

Operator ที่แตกต่างกันสามารถพัฒนา โดยการรวม operator พื้นฐาน 3 ตัวคือ identity, inverse และ threshold เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(a)

$$\begin{bmatrix} 15 & 15 & 0 & 0 & 2 \\ 13 & 13 & 15 & 0 & 0 \\ 0 & 0 & 7 & 14 & 14 \\ 0 & 1 & 2 & 3 & 4 \\ 15 & 14 & 13 & 12 & 11 \end{bmatrix}$$
 p matrix
$$\begin{bmatrix} 0 & 0 & 15 & 15 & 13 \\ 2 & 2 & 0 & 15 & 15 \\ 15 & 15 & 8 & 1 & 1 \\ 15 & 14 & 13 & 12 & 11 \\ 0 & 1 & 2 & 3 & 4 \end{bmatrix}$$
 q matrix

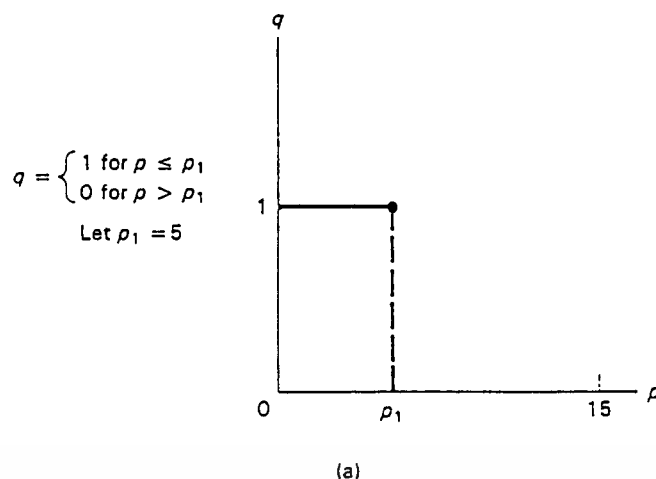
รูปที่ 4-4 Inverse operator : function , input (b) , and output (c).

Inverse threshold operator

ไบนารีเอาต์พุทอิมเมจสามารถที่จะกลับได้โดยประยุกต์ฟังก์ชัน threshold ในรูป 4-6 ไปเป็นอิมเมจค่า threshold ในรูป 4-6 แสดงให้เห็นโดยค่า P_1 และตัวอย่างการใช้ค่า threshold ของ 5 จุดที่แสดงซึ่งเป็นค่าที่ถูกนับรวมทั้งหมดทุกพิกเซลในภาพต้นแบบ ซึ่งที่ไหนที่สว่างจะกลายเป็นมืด และที่ไหนมืดจะกลายเป็นสว่าง operator ตัวนี้มีค่าได้มากที่สุด เพราะค่า threshold สามารถที่จะแปรเปลี่ยนไปเพื่อหา spatial information

Binary Threshold Interval

operator ในขั้นนี้จะให้ผลเป็นไบนารีเอาต์พุทอิมเมจ ซึ่งค่า gray input ในช่วง P_1 ถึง P_2 รวมทั้งค่า P_1 และ P_2 ด้วย จะถูกเปลี่ยนเป็น "1" และค่าที่อยู่เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



$$\begin{bmatrix} 15 & 15 & 0 & 0 & 2 \\ 13 & 13 & 15 & 0 & 0 \\ 0 & 0 & 7 & 14 & 14 \\ 0 & 1 & 2 & 3 & 4 \\ 15 & 14 & 13 & 12 & 11 \end{bmatrix}$$

p matrix

(b)

$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

q matrix

(c)

รูปที่ 4-6 Inverse threshold operator : function (a), input (b) , and output (c).

Gray scale threshold operator

operator ในขั้นนี้แสดงไว้ในรูป 4-9 ผลใน gray scale output image สำหรับค่า gray scale image ระหว่าง P_1 และ P_2 และทำค่าที่อยู่นอกเหนือจาก P_1 ถึง P_2 ให้เป็น "0"

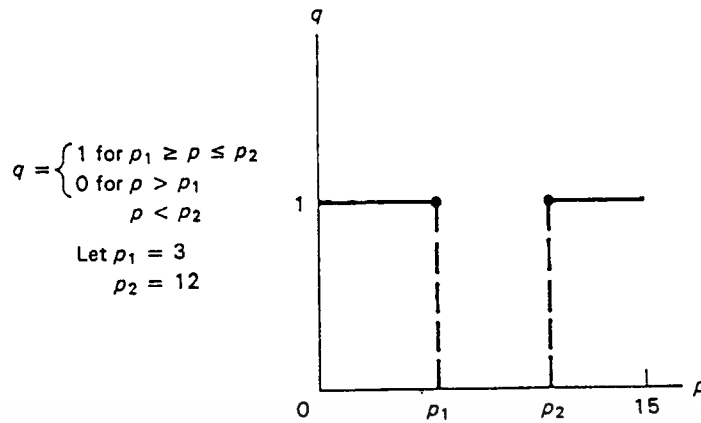
Inverse gray scale threshold operator

เอาที่พหุอิมเมจสามารถที่จะกลับได้โดยใช้ฟังก์ชันที่พิสูจน์ให้เห็นโดยรูป 4-10 operator ตัวนี้ สามารถจะให้แสงสว่างในรายละเอียดที่มีลักษณะเด่น ตัวอย่างเช่น พื้นที่ที่เหมือนกันในภาพ

Stretch Operator

operator ตัวนี้จะให้ผลใน full gray scale output image ที่สอดคล้องกับช่วงอินพุต P_1 และ P_2 และกด (suppress) ค่าทั้งหมดที่อยู่นอกเหนือจากช่วงนี้ แสดงให้เห็นในรูป 4-11

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(a)

15	15	0	0	2
13	13	15	0	0
0	0	7	14	14
0	1	2	3	4
15	14	13	12	11

p matrix

1	1	1	1	1
1	1	1	1	1
1	1	0	1	1
1	1	1	1	1
1	1	1	0	0

q matrix

(b)

(c)

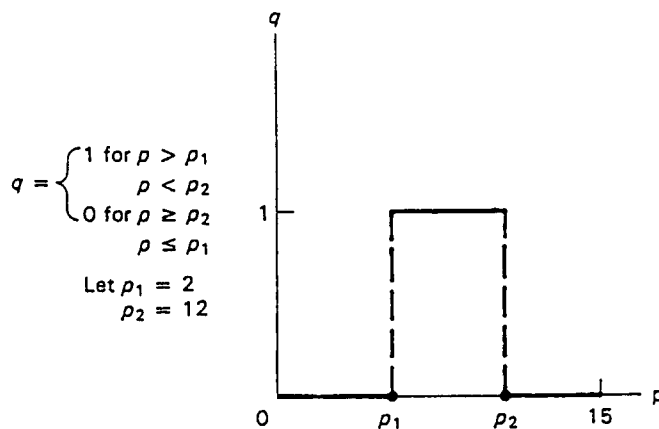
รูปที่ 4-7 Binary threshold interval operator : function (a) , input (b) , and output (c).

Gray scale reduction operator

operator ตัวนี้ให้ผลในเอาต์พุตอิมเมจ ซึ่งมีจำนวนค่า gray level น้อยกว่าจำนวนของ gray level ของอินพุตอิมเมจ ดังแสดงในรูป 4-12 ซึ่งอินพุตอิมเมจกับค่าระดับสีเทา 15 ระดับ จะถูกเปลี่ยนเป็นอิมเมจที่มีระดับสีเทา 5 ระดับ คือ q_1 , q_2 , q_3 , q_4 และ 15

4.3 Dyadic Two Point Transformation

Dyadic point-by-point operator ใช้กับข้อมูลที่มีตำแหน่งที่เหมือนกันในสองอิมเมจ two-input image matrix ในรูป 4-13 พิกเซลจาก A และ B ถูกใช้สร้างอิมเมจใหม่คือ C ขนาดของ matrix ไม่เปลี่ยนแปลง และฟังก์ชัน dyadic transformation f สามารถเป็นได้ทั้งแบบ linear หรือ nonlinear เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(a)

$$\begin{bmatrix} 15 & 15 & 0 & 0 & 2 \\ 13 & 13 & 15 & 0 & 0 \\ 0 & 0 & 7 & 14 & 14 \\ 0 & 1 & 2 & 3 & 4 \\ 15 & 14 & 13 & 12 & 11 \end{bmatrix}$$
 p matrix

(b)

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$
 q matrix

(c)

รูปที่ 4-8 Inverted binary threshold operator : function (a), input (b), and output (c).

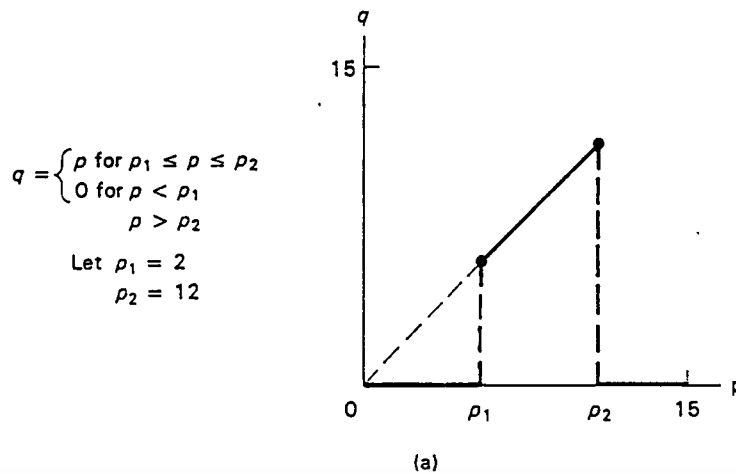
ฟังก์ชัน transformation ถูกนำไปใช้งานกับตำแหน่งของพิกเซลทั้งหมด อินพุทอิมเมจ (อินพุทอิมเมจมี 2 อิมเมจ) นั่นคือ ข้อมูลจากตำแหน่งพิกเซลในอิมเมจที่ 1 จะถูกรวมกับข้อมูลที่ตำแหน่งพิกเซลที่สอดคล้องกันในอิมเมจที่ 2 เพื่อสร้างค่าของพิกเซลที่สอดคล้องกันในอิมเมจที่ 3

คุณลักษณะของฟังก์ชันมีสมการดังนี้

$$C_{i,j} = f_D (a_{i,j} , b_{i,j})$$

ซึ่ง f_D เป็นฟังก์ชันของสองตัวแปร และ i กับ j เป็นช่วงจาก 0 ถึง m และ 0 ถึง n f_D สามารถที่จะบวก ลบ คูณ หาร ยกกำลังได้สูงสุด หรือ ฟังก์ชันอื่นซึ่งสามารถถูกหารได้ แต่ต้องระวังว่าอิมเมจทั้งสองนั้น จะต้องอยู่กับที่และคุณสมบัติการ calibration และ registration ของสองอินพุทอิมเมจ เพราะการคำนวณจะอยู่บนพื้นฐานความสอดคล้องของจุดต่อจุดในการหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



15	15	0	0	2
13	13	15	0	0
0	0	7	14	14
0	1	2	3	4
15	14	13	12	11

p matrix

(b)

0	0	0	0	2
0	0	0	0	0
0	0	7	0	0
0	0	2	3	4
0	0	0	12	11

q matrix

(c)

รูปที่ 4-9 Gray scale threshold operator : input (b) , and output (c).

ฟังก์ชันจะมีค่า factor K เป็นค่า scaling ที่เหมาะสม เพื่อรักษขนาด (magnitude) ของค่าเอาต์พุตให้อยู่ภายในช่วงไม่ให้เกิด overflow หรือสภาวะเป็นลบ (negative condition)

Transformation เกี่ยวข้องกับสองตัวแปร รวมทั้งคู่ของพิกเซลที่สอดคล้องกัน

$$R(i,j) = f[P(i,j) , q(i,j)]$$

เมื่อ p และ q เป็น matrix f เป็น functional operator และ R เป็นผลของ output matrix

4.3.1 Image Addition

image addition สามารถใช้ลดผลกระทบของสัญญาณรบกวน (noise) ในข้อมูลแสดงไว้ในรูป 4-14 ค่าของเอาต์พุต $C_{i,j}$ ซึ่งให้โดยสมการ

$$C_{i,j} = (a_{i,j} + b_{i,j}) / K$$

i, j เป็นค่าของช่วง ซึ่งค่า K เท่ากับจำนวนสั่ม (samples) image addition เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

dyadic จะทำการประมวลผลค่าเฉลี่ยข้อมูลในสอง input image matrices ถ้าหนึ่งในอินพุทอิมเมจเป็นค่าคงที่ ผลที่ได้จะสว่างขึ้นหรือมืดลงทั้งอิมเมจ ซึ่งจะปรากฏขณะเลื่อน (shift) histogram

ในตัวอย่าง special rule เหมือนการปิดเศษขึ้น จะต้องถูกหารและนำไปใช้กับฟังก์ชัน transformation ที่ให้ผลค่ากลาง ๆ มีความเที่ยงตรงที่จะทำให้ดีขึ้นและลดสัญญาณรบกวน (noise) เมื่อ procedure ถูกใช้กับจำนวน samples ที่มาก

$$C_{i,j} = (a_{i,j} + b_{i,j})/2$$

i, j เป็นช่วงของค่าจะถูกปิดขึ้น

4.3.2 Image Subtraction

การลบอิมเมจสามารถใช้ในการตรวจจับ (detect) การเคลื่อนที่ที่เกิดขึ้นระหว่างช่วงเวลา ระหว่างสองอิมเมจที่ถูกถ่ายเข้าคนละเวลา ถ้าทั้งสองอิมเมจเป็นฉาก (scene) เหมือนข้อมูลอาจจะแทนด้วย heat losses หรือ cooling ถ้า infrared spectrum เป็นข้อมูลต้น เพราะว่าการประมวลผลภาพ (image processing) ใช้เลขจำนวนบวก มันจำเป็นต้องกำหนดเอาต์พุต ซึ่งทำให้เป็นค่าบวกทั้งหมด นี้ยังสามารถปรับสเกลได้ ซึ่งค่าที่ติดลบก็จะ set ให้เป็น "0" ค่าที่ใหญ่มากก็ set ให้เท่ากับค่าสูงสุดของ gray scale level (255 ในส่วนของรูป 4-15) อีกส่วนหนึ่งที่ต้องทำให้เหมาะสม คือ การกำหนดความแตกต่างแบบไม่คิดเครื่องหมายหรือค่า absolute ของความแตกต่างดังแสดงใน matrix b

ในกรณีแรก (ตัวอย่าง a) มีความสัมพันธ์เป็น

$$C_{i,j} = K (a_{i,j} + b_{i,j})$$

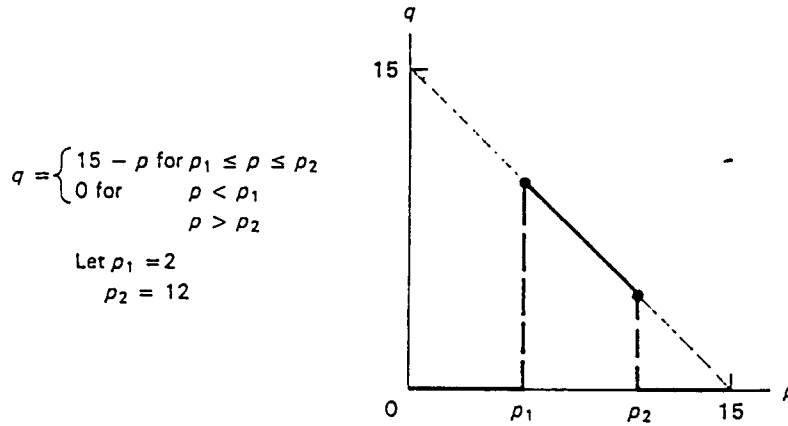
เมื่อ K เป็นฟังก์ชันแบบไม่เชิงเส้น ซึ่งค่าต่ำสุดของ $C_{i,j}$ เป็น "0" และค่าสูงสุด คือ 255

ในตัวอย่าง b ของรูป 4-15 มีความสัมพันธ์เป็น

$$R_{i,j} = K (a_{i,j} - b_{i,j})$$

เมื่อ $K = 1$, ขณะที่ค่าอยู่ในช่วง 0 ถึง 254

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(a)

$$\begin{bmatrix} 15 & 15 & 0 & 0 & 2 \\ 13 & 13 & 15 & 0 & 0 \\ 0 & 0 & 7 & 14 & 14 \\ 0 & 1 & 2 & 3 & 4 \\ 15 & 14 & 13 & 12 & 11 \end{bmatrix}$$
 p matrix

(b)

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 13 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 8 & 0 & 0 \\ 0 & 0 & 13 & 12 & 11 \\ 0 & 0 & 0 & 3 & 4 \end{bmatrix}$$
 q matrix

(c)

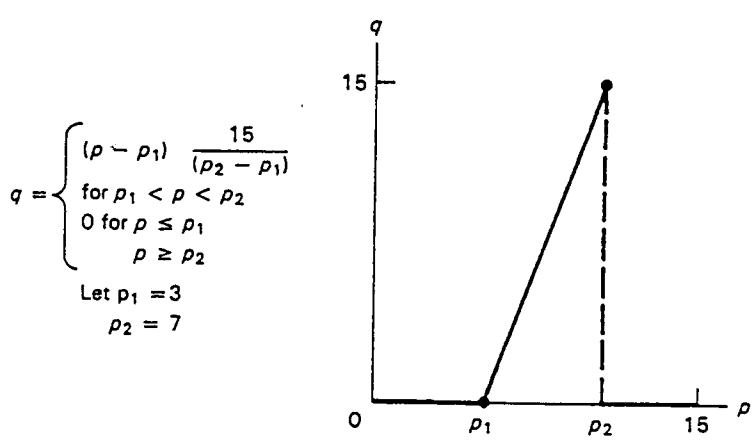
รูปที่ 4-10 Inverted gray scale threshold operator :
 function (a), input (b) and output (c).

เอาท์พุทของสองวิธีเป็นความแตกต่างโดยสมบูรณ์ ผู้ออกแบบระบบหรือผู้วิจัยต้องคำนวณหา ซึ่งวิธีการจัดเตรียมผลที่ได้ทั้งหมดสำหรับรายละเอียดการประยุกต์ใช้งาน ทั้งสองตัวอย่างแสดงให้เห็นว่า มีหลายทางที่เป็นพื้นฐานการประมวลผลของการบวก การลบ และการคูณ สามารถใช้ปรับข้อมูลใน image matrix ผู้ออกแบบจะต้องคำนวณหาชนิดของข้อมูลที่ต้องการ สำหรับการประยุกต์ใช้งานและเทคนิคการประมวลผลข้อมูล (data processing technique) ซึ่งจะทำให้ลักษณะเด่นของข้อมูลที่ค้ำจุนในข้อมูลภาพ (image data) monadic operator f จะประยุกต์ใช้ $C_{i,j}$ matrix ในรูป 5-16 ค่า $R_{i,j}$ เป็นการบดพิเศษสมการ

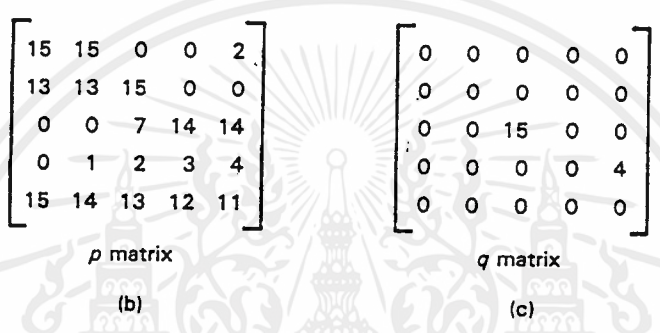
$$R_{i,j} = (C_{i,j} + 100) * 255/354$$

ถูกใช้ในการเปลี่ยนค่า $R_{i,j}$ จะเป็น "0" เมื่อ $C_{i,j}$ คือ -100 (minimum value) และ 255 เมื่อ $C_{i,j}$ เป็น 254 (maximum value)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(a)



รูปที่ 4-11 Stretch operator : function (a) , input (b) , output (c).

4.3.3 Image Multiplication

การคูณของสองอิมเมจในการประมวลผลภาพ ถูกแสดงในรูป 4-17 นี้เป็น การใช้ที่ถูกต้อง สำหรับความไม่เป็นเชิงเส้นของตัวเซ็นเซอร์ซึ่งมีช่วงว่างที่ไม่สม่ำเสมอ เนื้อพื้นที่ที่มองเห็นสิ่งนี้จะถูกต้องโดยการคูณอิมเมจ image ด้วย matrix ที่ถูกต้อง ความสัมพันธ์แสดงได้โดยสมการ

$$C_{i,j} = K [(a_{i,j} + b_{i,j}) + a_{i,j}]$$

ซึ่งค่าทั้งหมดถูกปิดเศษขึ้นเป็นจำนวนเต็มหมด ค่าสูงสุดคือ 255 และ $b_{i,j}$ เป็น correction factor

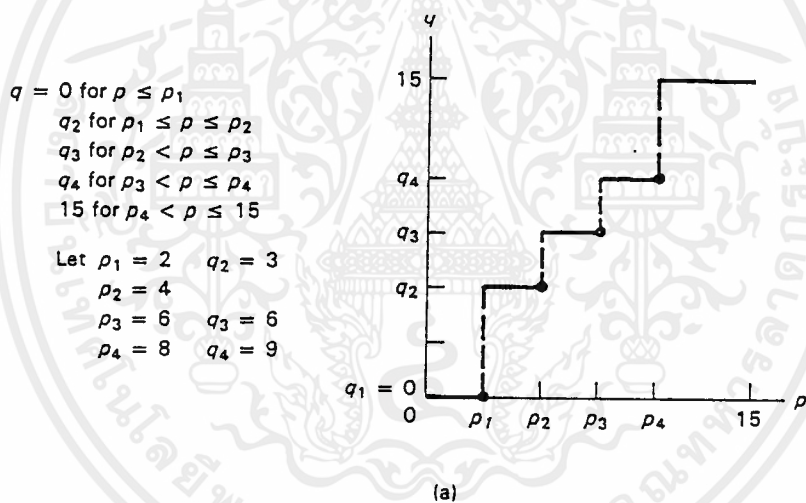
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการคูณ operator แบบอื่น ๆ จะเป็นการสร้าง window เพื่อที่จะลดการคำนวณในบริเวณที่กำลังพิจารณา correction matrix b_{ij} จะแสดงโดยสมการ

$$C_{i,j} = a_{i,j} + b_{i,j}$$

ซึ่ง $b_{i,j}$ คือ "1" สำหรับทุกค่าที่อยู่ภายในพื้นที่ window และ เป็น "0" สำหรับทุกค่า ที่อยู่ภายนอกพื้นที่ window

ขนาดของ output imagematrix จะมีขนาดเดียวกันกับ input image matrix ไม่มีแถว (row) หรือหลัก (column) หายไปในการประมวลผล แต่จะเกิดขึ้นกับ การ convolution dyadic process สำหรับการ transformation ถูกใช้ในการบวกอิมเมจ ลบอิมเมจ และการคูณอิมเมจ ค่า normalization constant อาจจะใช้ระหว่าง dyadic operation หรือ ฟังก์ชัน nonadic scaling operator ใน subsequence phase



$$\begin{bmatrix} 15 & 15 & 0 & 0 & 2 \\ 13 & 13 & 15 & 0 & 0 \\ 0 & 0 & 7 & 14 & 14 \\ 0 & 1 & 2 & 3 & 4 \\ 15 & 14 & 13 & 12 & 11 \end{bmatrix}$$

p matrix

(b)

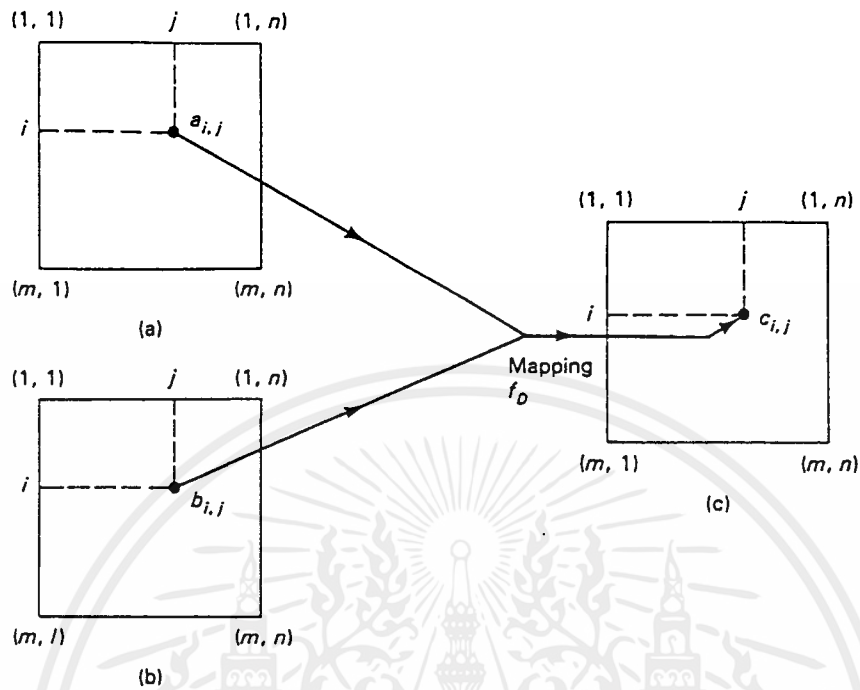
$$\begin{bmatrix} 15 & 15 & 0 & 0 & 0 \\ 15 & 15 & 15 & 0 & 0 \\ 0 & 0 & 9 & 15 & 15 \\ 0 & 0 & 0 & 3 & 3 \\ 15 & 15 & 15 & 15 & 15 \end{bmatrix}$$

q matrix

(c)

รูปที่ 4-12 Level reduction operator : function (a), input (b) , and output (c).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-13 Dyadic multipoint-to-point operator : input image A (a) and input image B (b) are mapped into a new output image C (c).

$$\begin{array}{ccc}
 \begin{bmatrix} 0 & 12 & 142 & 255 \\ 1 & 6 & 40 & 254 \\ 24 & 0 & 20 & 255 \\ 30 & 2 & 10 & 240 \end{bmatrix} &
 \begin{bmatrix} 14 & 11 & 9 & 253 \\ 3 & 5 & 39 & 254 \\ 11 & 1 & 19 & 255 \\ 18 & 2 & 11 & 256 \end{bmatrix} &
 \begin{bmatrix} 7 & 12 & 76 & 254 \\ 2 & 6 & 40 & 254 \\ 18 & 1 & 20 & 255 \\ 23 & 2 & 11 & 248 \end{bmatrix} \\
 a_{i,j} & b_{i,j} & c_{i,j} \\
 \text{input 1} & \text{input 2} & \text{output}
 \end{array}$$

รูปที่ 4-14 Image addition for $i=j=4$

4.3 Convolution : spatial transformation

อิมเมจ matrix ใหม่สามารถที่จะถูกสร้างขึ้นได้โดยค่าของพิกเซล ที่ถูกกำหนดไว้ในแต่ละ ตำแหน่งเป็นฟังก์ชันของค่าพิกเซลตำแหน่งข้างเคียง ที่แสดงในรูป 5-18 เป็น 3×3 convolution

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(a)

$$\begin{bmatrix} 0 & 12 & 142 & 255 \\ 1 & 6 & 40 & 254 \\ 24 & 0 & 20 & 255 \\ 30 & 2 & 10 & 240 \end{bmatrix} \quad \begin{bmatrix} 14 & 11 & 9 & 253 \\ 3 & 25 & 100 & 0 \\ 11 & 1 & 80 & 1 \\ 30 & 2 & 110 & 255 \end{bmatrix} \quad \begin{bmatrix} -14 & 3 & 133 & 2 \\ -2 & -19 & -60 & 254 \\ 15 & -1 & -60 & 254 \\ 0 & 0 & -100 & -15 \end{bmatrix}$$

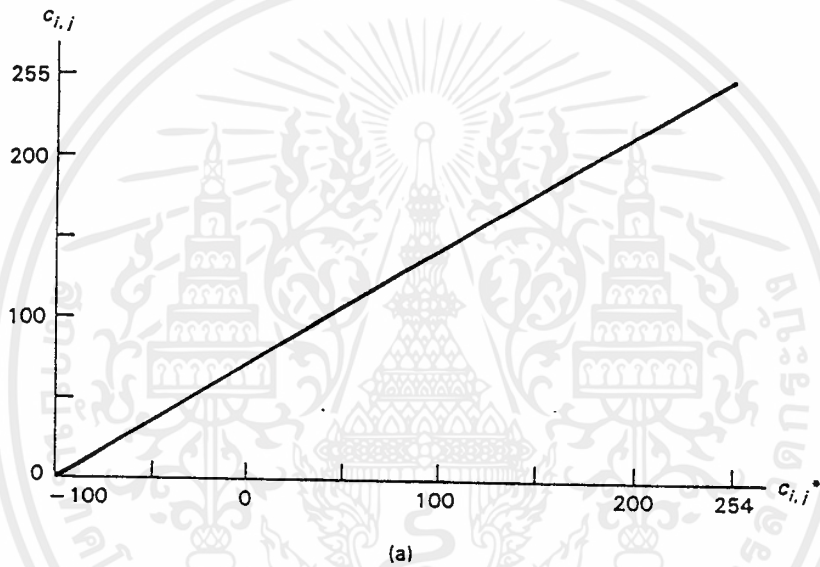
Input #1 = $a_{i,j}$ Input #2 = $b_{i,j}$ Output = $c_{i,j}^*$

(b)

$$\begin{bmatrix} 14 & 3 & 133 & 2 \\ 2 & 19 & 60 & 254 \\ 15 & 1 & 60 & 254 \\ 0 & 0 & 100 & 15 \end{bmatrix}$$

Output $c_{i,j}$

รูปที่ 4-15 Image subtraction (absolute value) ; simple ,
but useful only under very limited condition.



$$\begin{bmatrix} 62 & 74 & 168 & 74 \\ 71 & 59 & 29 & 255 \\ 83 & 72 & 29 & 255 \\ 72 & 72 & 0 & 62 \end{bmatrix}$$

Output $c_{i,j}$

(b)

รูปที่ 4-16 Image subtraction with function (a) which
generates resultant output (b) having all positive values

$$\begin{bmatrix} 0 & 12 & 142 & 255 \\ 1 & 6 & 40 & 254 \\ 24 & 0 & 20 & 255 \\ 30 & 2 & 10 & 240 \end{bmatrix} \quad \begin{bmatrix} .3 & .4 & .1 & .1 \\ .3 & 0 & 0 & .1 \\ .3 & 0 & 0 & 0 \\ .4 & .1 & 0 & .1 \end{bmatrix} \quad \begin{bmatrix} 0 & 17 & 157 & 255 \\ 2 & 6 & 40 & 255 \\ 32 & 0 & 20 & 255 \\ 42 & 3 & 10 & 255 \end{bmatrix}$$

Image matrix a_{ij} Correction matrix b_{ij} Output matrix c_{ij}

รูปที่ 4-17 Image multiplication

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานภายในเท่านั้น ไม่ควรนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \times \begin{bmatrix} + & + & - \\ - & + & + \\ + & + & - \end{bmatrix} = \begin{bmatrix} & & \\ & e' & \\ & & \end{bmatrix}$$

Where:

$$e' = +a + b - c - d + e + f + g + h - i$$

รูปที่ 4-18 Convolution (neighborhood) transformation.

ใน 3×3 convolution ค่าของพิกเซลของตำแหน่งกลางจะถูกคำนวณบนพื้นฐานค่าที่อยู่เคียง 9 ตำแหน่ง ตำแหน่งกลางจะถูกเลื่อน (shift) ไปอีกหนึ่งพิกเซล และทำการประมวลผลซ้ำ จนกระทั่งอิมเมจ matrix ถูกสร้างจนเสร็จ มันจะเป็นการแสดงความขนาดของ matrix ผลลัพธ์ ที่ถูกลดขนาดลงสองพิกเซลในแต่ละทิศทางที่เป็นขอบของอิมเมจ ถ้าไม่ต้องการลดขนาดของ matrix สามารถทำได้โดย การเติมข้อมูลที่เป็นค่าคงที่เข้าไปในแถวและหลักข้างเคียง ที่อยู่ขอบนอกของอิมเมจ matrix

บทที่ 5

ขอบภาพ

5.1 Digital Filter

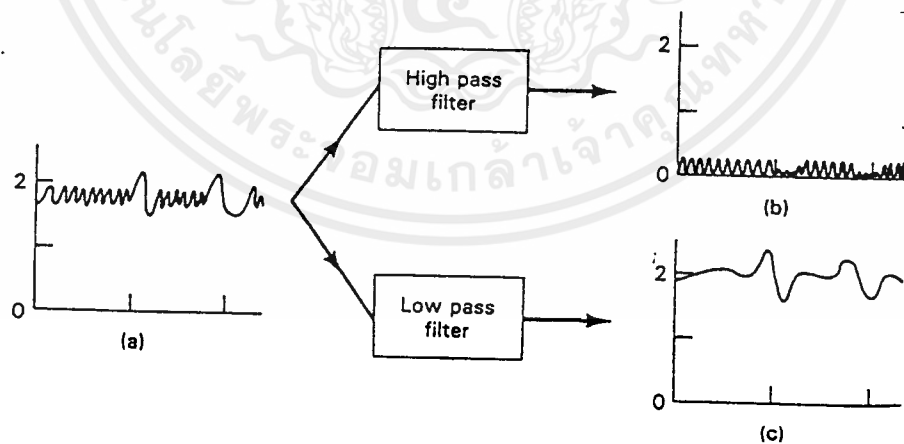
Digital Filter สามารถแบ่งออกได้เป็น low pass filter และ high pass filter ขึ้นอยู่กับส่วนของ frequency spectrum ที่มีผลกระทบ ผลของสัญญาณรวมจะมีทุกความถี่ ดังแสดงในรูป 5-1

สัญญาณเอาต์พุตของ high pass filter จะมีแต่ความถี่สูง เอาต์พุตของ low pass filter จะมีแต่ DC และความถี่ต่ำ คุณสมบัติของความถี่ cutoff สามารถเปลี่ยนแปลงโดยเลือกค่าที่จะใช้ในการ filter ได้

ความถี่ต่ำในอิมเมจจะถูกทำให้มีคุณลักษณะโดยการเปลี่ยนแปลงของความคมชัด (contrast) พิกเซลข้างเคียงอย่างช้า ๆ ในความสัมพันธ์กับความถี่สูง จะมีการเปลี่ยนแปลงความชัดอย่างรวดเร็วที่บริเวณขอบของวัตถุ

5.2 Low Pass Filter

Low pass filter จะมีผลต่อเฉพาะความถี่ต่ำในข้อมูลภาพ (image data) และ จะลดทอนความถี่สูงดังแสดงในรูป 5-2 การสุ่มจุดที่ไม่ดีในอิมเมจสามารถพิจารณาสัญญาณรบกวนและความถี่สูง เพราะว่าค่าพิกเซลที่อยู่ข้างเคียงมีการเปลี่ยนแปลงอย่างรวดเร็ว ผลของรอยต่างในอิมเมจ สามารถลดลงได้ โดยการใช้ simple averaging



รูปที่ 5-1 Results of low pass (b) and high pass (c) digital filter on a composite signal (a) containing both low and high frequencies as well as a DC bias

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

filter สัมประสิทธิ์ของ matrix สำหรับ averaging filter เป็น 3×3 convolution ซึ่งจะมี 9 element ดังแสดงในรูปข้างล่าง

$$\begin{bmatrix} f_{1,1} & f_{2,1} & f_{3,1} \\ f_{1,2} & f_{2,2} & f_{3,2} \\ f_{1,3} & f_{2,3} & f_{3,3} \end{bmatrix} = \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

รูปที่ 5-2 Coefficient values of a low pass filter matrix.

ค่าของพิกเซลในเอาต์พุต matrix ถูกคำนวณโดยการนำ convolution mask ไป convolution กับข้อมูลภาพ (image data) $P_{i,j}$ กับค่าสัมประสิทธิ์ของ matrix $f_{i,j}$ และให้ผลไว้ที่ตำแหน่งกลางของพื้นที่ในอิมเมจ matrix ใหม่ ค่าใหม่ของ $q_{2,2}$ หาได้จาก

$$q_{2,2} = f_{1,1} \times p_{1,1} + f_{2,1} \times p_{2,1} + \dots + f_{3,3} \times p_{3,3}$$

เงื่อนไขของ low pass filter คือ

- (1) ส.ป.ส. ทั้งหมดต้องเป็นบวก และ
- (2) ผลรวมทั้งหมดของ ส.ป.ส. จะต้องเท่ากับ "1"

$$\begin{bmatrix} 7 & 7 & 7 & 7 \\ 7 & (0) & 7 & 7 \\ 7 & 7 & 7 & 7 \\ 7 & 7 & 7 & 7 \end{bmatrix} \quad \begin{bmatrix} 7 & 7 & 7 & 7 \\ 7 & (7) & 7 & 7 \\ 7 & 7 & 7 & 7 \\ 7 & 7 & 7 & 7 \end{bmatrix}$$

matrix with
bad element

matrix after
low pass filter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าผลรวมของ ส.ป.ส. มากกว่า "1" จะเป็นผลทางการขยาย ถ้าผลรวมน้อยกว่า "1" จะให้ผลทางลดทอน และขนาดของค่าพิกเซลที่อยู่บนอิมเมจจะถูกลดลง ดังนั้นอิมเมจจะมีคลง

การประยุกต์ใช้งานของ ส.ป.ส. ขนาด matrix 3*3 กับพื้นที่ข้อมูล ซึ่งทุกพิกเซลมีค่าเหมือนกันหมดจะไม่มีข้อบกเว้น ซึ่งมี gradient ระหว่างพิกเซลข้างเคียง ผลของ low pass digital filter ในการลดทอนสัญญาณรบกวน สามารถพิสูจน์ให้เห็นโดยการพิจารณา กรณีที่มีหนึ่งพิกเซลใน matrix ซึ่งเป็น "0" กำหนดให้เป็น bad element ใน sensor array ที่ใช้จับ (acquire) ข้อมูลพิกเซลข้างเคียงจะมีค่าเป็น "high" หมด

ถ้าค่าของ bad pixel ยังคงเป็นค่า "high" ของ "low" เพราะผลของจุดต่างสีขาวการประยุกต์ใช้งานของ low pass filter จะมีผลกับการลดขนาดของค่า bad pixel

ผลของการเพิ่มขึ้นของขนาดของบริเวณรอบ ๆ (neighborhood) เป็น 5*5 หรือ 9*9 สามารถพิสูจน์โดยการเปรียบเทียบผลของการใช้ low pass filter กับการมีพิกเซลเสีย (bad pixel) หนึ่งตัว ("0" ในบริเวณที่อื่นมีค่าเป็น "1" หมด) ถูกแสดงไว้ในตาราง 5-1

TABLE 5-1 Effect of Filter Dimensions

Size of matrix	Convoluteds Value	Raw Value Bad Element	Relative Computative Time
3 * 3	1/9	8/9	1
5 * 5	1/25	24/25	25/9 = 2-7/9
9 * 9	1/81	80/81	81/9 = 9

พบว่า ขนาดของ matrix ที่ใหญ่จะเป็นเหตุให้ค่าที่คำนวณได้ใหม่ซึ่งเหมาะกับพิกเซลข้างเคียง แต่จะเสียเวลาในการคำนวณ ขนาด 3 * 3 ถูกนำมาใช้ในงานอุตสาหกรรมเป็นส่วนใหญ่ เพราะมันลดเวลาในการคำนวณประมวลผล เวลาประมวลผล และเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้นทุนของเครื่องมือ ผลที่ได้ จะถูกตรวจสอบให้แน่นอนว่ามันอยู่ภายใน acceptable error tolerance requirement

5.3 High Pass Filter

High pass digital filter จะกลับกันกับ low pass filter การ filter จะไม่เปลี่ยนแปลงความถี่สูงของสัญญาณ (signal)

ผลของแสงที่เป็น back ground จะมีผลใน DC bias ในระบบการมองเห็นข้อมูลเพราะ ค่าต่ำสุดของพิกเซลทั้งหมดที่อยู่เหนือบางค่าที่กำหนดให้ ผลของการใช้ high pass filter กับข้อมูลสำหรับอิมเมจขนาด 6×6 ถูกแสดงใน histogram ของรูป 5-3 ค่าพิกเซลที่สูงอาจจะเปลี่ยนแปลงหรือไม่เปลี่ยนก็ได้ ขึ้นอยู่กับจุดอิ่มตัว (degree of saturation) ของอิมเมจ

ส.ป.ส. ของ high pass filter ตั้งไว้สำหรับ matrix convolution mask ขนาด 3×3 คือ

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

- (1) ส.ป.ส. สามารถเป็นได้ทั้งบวก (positive) หรือ ลบ (negative)
- (2) ผลรวมของ ส.ป.ส. เป็น "0"

เพราะค่ารวมของ ส.ป.ส. เท่ากับ "0" ส่วนที่เป็น DC จะถูกกดไว้อย่างสมบูรณ์ ถ้าผลรวมเท่ากับ "+1" ค่าของ DC จะยังคงมีค่าเท่ากับสัญญาณต้นแบบ (original signal) การ filter ถูกใช้กับ ส.ป.ส. ของ matrix mask ขนาด 3×3 บนข้อมูลภาพ (image data) การคูณ ส.ป.ส. และข้อมูลในทุกค่าของ 9 element และผลบวกทั้ง 9 เพื่อหาค่าใหม่ของตำแหน่ง element ที่จุดกึ่งกลางของ matrix 3×3 mask จะถูกเลื่อนไปอีก "1" และ ประมวลผลซ้ำ และ หาค่าของ element ตัวถัดไป ตัวอย่างที่แสดงการประมวลผลเพื่อหาค่าในรูป 5-4 convolution ของ mask และข้อมูลให้ค่าของตรงกลาง element ที่ตำแหน่ง (2,2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{bmatrix} +1 & +1 & +1 \\ +1 & -8 & +1 \\ +1 & +1 & +1 \end{bmatrix} \quad \begin{bmatrix} 4 & 8 & 8 \\ 4 & 8 & 8 \\ 4 & 8 & 8 \end{bmatrix} \quad \begin{bmatrix} - & - & - \\ - & [] & - \\ - & - & - \end{bmatrix}$$

mask

data

element

$$\begin{aligned} (2,2) &= +(1 * 4) + (1 * 8) + (1 * 8) + (1 * 4) - (8 * 8) \\ &\quad +(1 * 8) + (1 * 8) + (1 * 4) + (1 * 8) \\ &= +4+8+8+4-64+8+4+8+8 \\ &= +52-64 \\ &= -12 \end{aligned}$$

ขอบภายในอิมเมจ matrix จะปรากฏการเปลี่ยนแปลงรูปร่างในค่าของ พิกเซลข้างเคียงดังแสดงในรูป 5-4 ตัวอย่าง B ข้อมูลมีความสัมพันธ์กับตำแหน่งและขนาด ของขอบภาพซึ่งถูกคำนวณ หรือแทนด้วย high spatial frequency ของค่าพิกเซลใน อิมเมจ

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & +9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

modified coefficient matrix

$$\begin{bmatrix} x & x & x & x & x & x & x \\ x & 4 & 4 & 4 & 4 & 4 & x \\ x & 4 & 4 & 4 & 4 & 4 & x \\ x & 4 & 4 & 4 & 4 & 4 & x \\ x & x & x & x & x & x & x \end{bmatrix} \quad \begin{bmatrix} x & x & x & x & x & x & x \\ x & 4 & -8 & +20 & 8 & 8 & x \\ x & 4 & -8 & +20 & 8 & 8 & x \\ x & 4 & -8 & +20 & 8 & 8 & x \\ x & x & x & x & x & x & x \end{bmatrix}$$

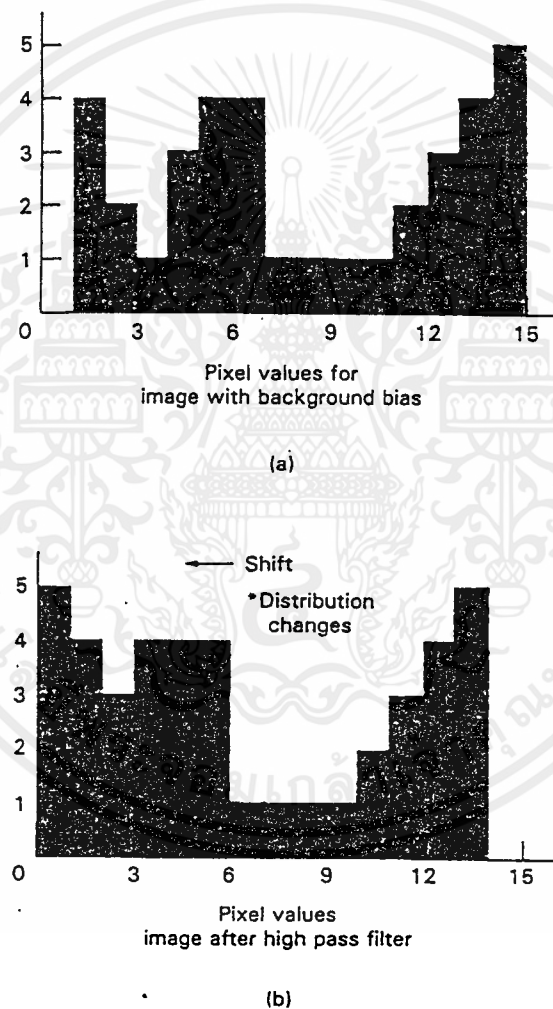
effect of modified high pass filter on examples a and b

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พบว่า high pass filter สามารถกำจัดค่าคงที่ที่ bias ในตัวอย่าง A และทำให้ขอบภาพ (edge pixel) ดีขึ้น ในตัวอย่าง B ซึ่งผลของ matrix มีเส้นของแถว และค่าพิกเซลเป็น "high"

ส.ป.ส.ของ matrix สามารถ modify ซึ่งผลรวมเท่ากับ "1" และผลลัพธ์ที่แตกต่างถูกกำหนดให้ เมื่อ filter ถูกนำไปใช้กับข้อมูลในตัวอย่าง A และ B

ผลของการ convolution ของการ modify high pass filter กับข้อมูลในอิมเมจ A และ B จัดเตรียมอิมเมจไว้ให้ ซึ่ง bias ก็ยังคงอยู่ และความถี่สูงจะถูกขยาย (amplified)

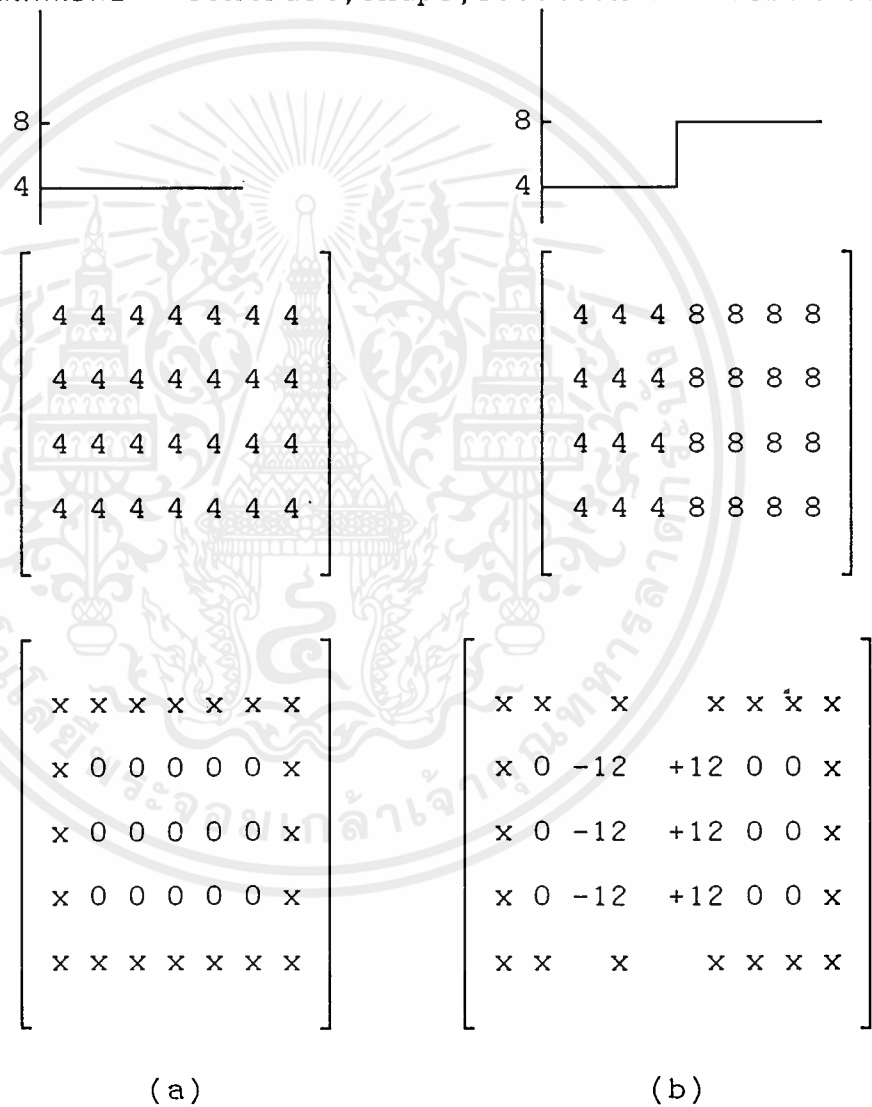


รูปที่ 5-3 Effect of a high pass filter on a histogram.

When values of pixels are shifted, the shape of the histogram changes.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

matrix ขนาด 3*3 เป็นส่วนที่เล็กที่สุดและต่ำที่สุดในการคำนวณที่ต้องทำการ convolution วิธีที่ง่าย คือ ขยายเป็น 5*5 และ 9*9 matrices แต่ผลอาจจะให้ความแตกต่างกันไม่เด่นชัด ขณะที่การคำนวณถูกขยายโดยองค์ประกอบ (factor) ของ 2.7 และ 9 ฉะนั้นการ convolution ที่มีขนาด 3*3 สามารถทำให้มีประสิทธิภาพสำหรับการใช้งานในงานอุตสาหกรรม ผลของ 9*9 สามารถทำบนระบบ 3*3 โดยการซ้ำกับการ convolution ขนาด 3*3 ค่าที่อยู่นอกเหนือส่วนนี้จะสูญหาย เพราะว่าเป็นข้อมูลที่ไม่สำคัญ high pass filter มีคุณสมบัติ ซึ่งข้อมูลในบริเวณขอบถูก modify และขอบมีผลต่อการขยาย edge detection เป็นความรู้พื้นฐานในการหาจากอิมเมจ เพราะว่าข้อมูลมีความสัมพันธ์กับ contrast, shape, location และ dimension



รูปที่ 5-4 Results of high pass filter on image not having an edge (a) and on an image containing an edge (b). x equals last value because of incomplete data.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขอบภาพ (edge) เป็นพารามิเตอร์วิกฤต (critical parameter) ของแนวความคิด และ ตำแหน่งของขอบภาพ เป็นสิ่งสำคัญยิ่ง ในระบบการมองเห็นจะใช้ความแตกต่างจาก high pass filter ไปทำให้จุดเด่น (feature) ที่กำลังพิจารณาอยู่ให้ดีขึ้น

5.4 Edge Enhancement

ขอบภาพคือบริเวณรอบ ๆ ระหว่างสองบริเวณที่มีความแตกต่างของ gray level ขอบภาพ ในอุดมคติจะเป็นฟังก์ชันแบบขั้น (step function) และสามารถที่จะเห็นความคมชัดหรือพรมัว ซึ่งความพรมัวหรือขอบภาพที่หนาสามารถพิจารณาองค์ประกอบของจำนวนตำแหน่งขอบภาพที่เล็ก ดังในรูป 5-5 ในความเป็นจริงของการใช้งานระบบมองเห็น ขอบภาพโดยปกติจะมีสัญญาณรบกวน (noise) รวมอยู่ในข้อมูลด้วย

ความถี่สูงก็ถูกรวมอยู่กับการเปลี่ยนแปลงที่ชัดเจน ในค่าความเข้มแสงของพิกเซล การทำให้ขอบภาพดีขึ้นสามารถทำได้โดย การใช้รูปแบบขอบ high pass filter กับข้อมูลภาพ (image data) ตัว operator ที่มีขนาดใหญ่ จะทำให้ขอบภาพดีขึ้นที่ยังคงรูปร่างของตัวอักษรแต่ที่นิยมใช้ในงานอุตสาหกรรม คือ

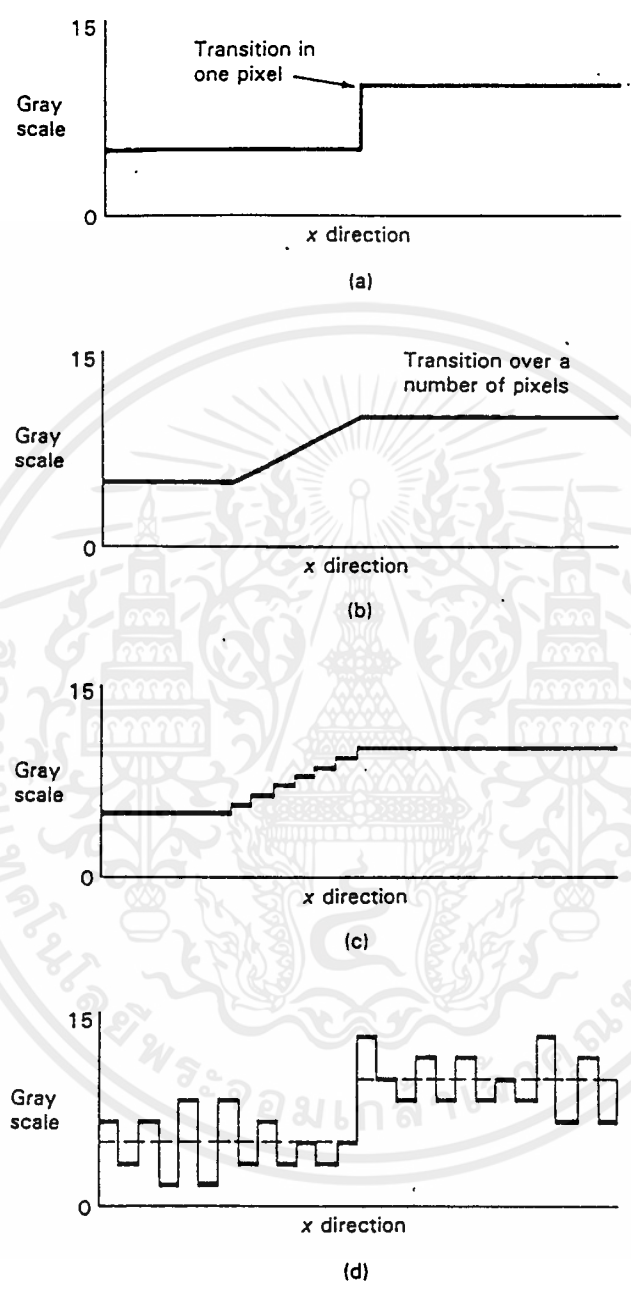
- Laplacian edge enhancement
- Robert's gradient
- Sobel edge detector

ขอบภาพแบบนั้น (step) จะแยกพื้นที่ออกเป็นสองส่วนกับความสัมพันธ์กับความสม่ำเสมอของ gray level บนแต่ละข้างของขอบภาพ เส้นที่พิจารณาจะมีขอบภาพสองขั้น (2 step) คือ อันหนึ่งเป็นบวก และอีกอันหนึ่งเป็นลบ ความกว้างของเส้นขึ้นอยู่กับการแยกระยะทางระหว่างสอง step

5.4.1 Laplacian Edge Enhancement Operator

Laplacian operator ที่ใช้ในระบบการประมวลผลเป็นพื้นฐานทางคณิตศาสตร์ อนุพันธ์ย่อยของ laplacian แสดงในรูปของฟังก์ชันต่อเนื่องได้ดังนี้

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$



รูปที่ 5-5 Types of edges in an image : ideal step (a), wide (b), break down of wide edge (c), and ideal step with noise (d).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมการคณิตศาสตร์ที่ต่อเนื่อง สามารถประมาณโดยความแตกต่างของ operator ในการ match ที่ไม่ต่อเนื่อง ของการใช้งานประมวลผลการมองเห็นแบบดิจิทัล Rotationally insensitive ของการประมวลผล laplacian จำเป็นที่จะต้องหา การเปลี่ยนแปลงความคมชัดของความเข้มแสงที่พิกเซล ในทิศทาง x และ y

$$L(i,j) = \Delta^2 p(i,j) = \Delta x^2 p(i,j) + \Delta y^2 p(i,j)$$

เมื่อ

$$\Delta x^2 = [p(i-1,j) - p(i,j)] - [p(i,j) - p(i+1,j)]$$

$$\Delta y^2 = [p(i,j+1) - p(i,j)] - [p(i,j) - p(i,j-1)]$$

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \quad \Delta x^2 = [d-e] - [e-f]$$

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \quad \Delta y^2 = [h-e] - [e-b]$$

เมื่อรวมพจน์เข้าด้วยกันจะได้ผลดังข้างล่างซึ่งเป็น Laplacian operator

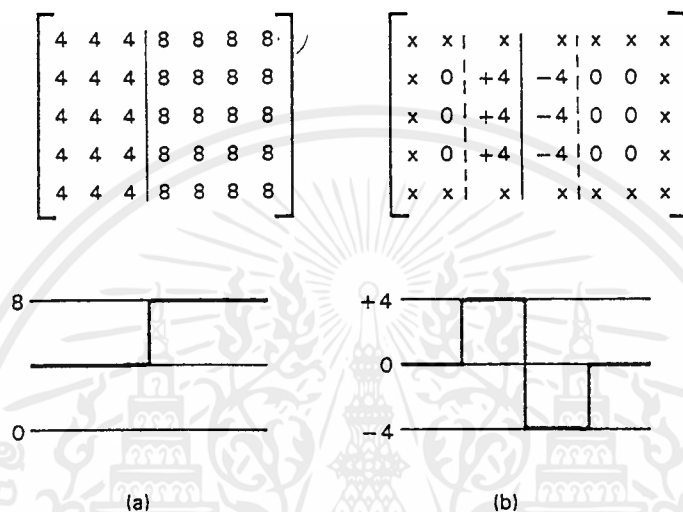
$$L(i,j) = b + d + f + h - 4e$$

$L(i,j)$ สามารถลดรูปให้เป็น mask ได้ดังนี้

$$\begin{bmatrix} 0 & +1 & 0 \\ +1 & -4 & +1 \\ 0 & +1 & 0 \end{bmatrix}$$

สรุป Laplacian operator คำนวณหาความแตกต่างระหว่าง gray level เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของจุดศูนย์กลางของพิกเซล และค่าเฉลี่ยของ gray level ของพิกเซลข้างเคียง 4 ตัว ในทิศทางแนวนอนและแนวตั้ง Laplacian operator เป็น high pass filter เพราะว่าผลรวมของ ส.ป.ส. เป็น "0" และมันมีทั้ง ส.ป.ส. ที่มีค่าเป็นบวกและลบ การใช้งานของ Laplacian กับ input matrix ของตัวอย่าง B ประกอบด้วยขอบภาพทางแนวตั้ง ดังแสดงในรูป 5-6



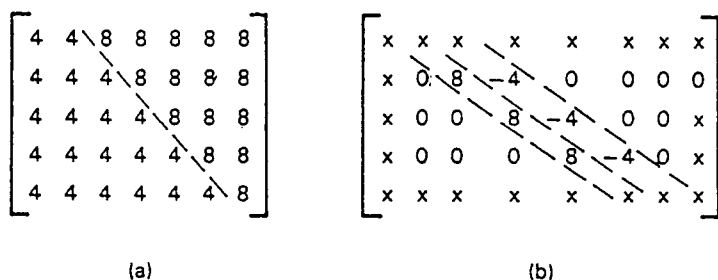
รูปที่ 5-6 Laplacian operator applied to image containing a vertical edge (a) results in zero crossing identification of edge in output (b).

การใช้งานของ Laplacian กับข้อมูลภาพและขอบภาพที่มีมุมดังรูป 5-7

การแสดงผลของขอบภาพที่เป็นบวกที่แสดงใน รูป 5-8 มีผลใน positive impulse ตามด้วย negative impulse เพราะ Laplacian operator กำลังทำอนุพันธ์กับผิว ขอบภาพจะถูกกำหนดตำแหน่งระหว่างสอง impulse พังก์ชันที่จุดซึ่ง impulse function ตัดผ่าน "ZERO" ดังนั้น Laplacian operator สามารถใช้ทำให้ขอบภาพดีขึ้นได้ และการตรวจจับการตัดผ่าน "ZERO" สามารถใช้กับตำแหน่งของขอบภาพ impulse function จาก 4 รูปแบบของขอบภาพดังแสดงในรูป 5-8

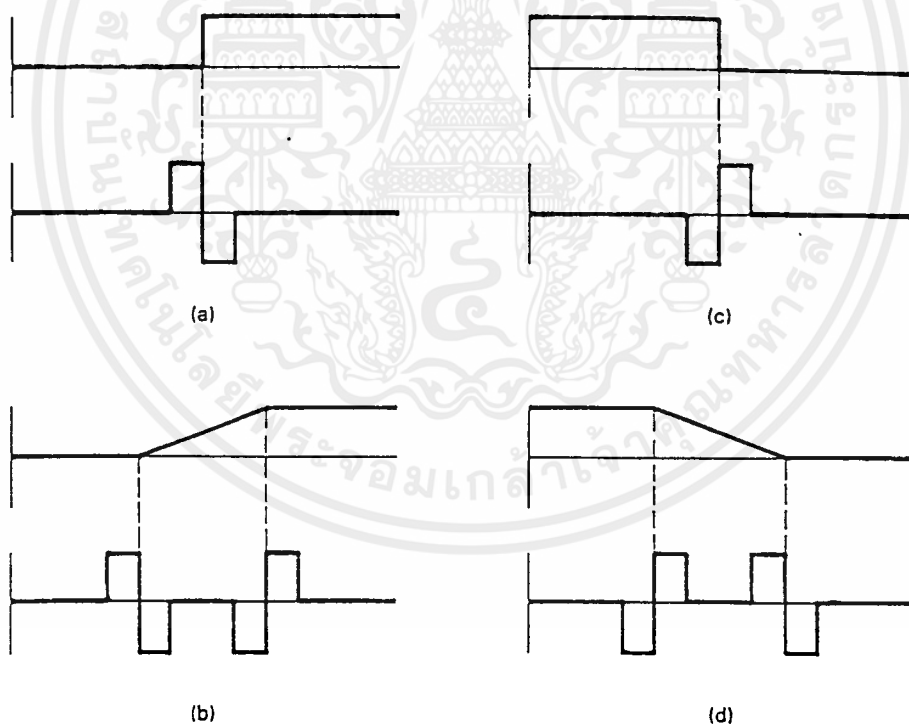
ขอบภาพและเส้นในขอบภาพ สามารถทำงานได้ในทิศทางต่างๆ มันเป็นความต้องการอย่างสูง ซึ่ง operator เป็น Isotropic ถ้ามันถูกใช้พัฒนาอิมเมจที่มีจุดสนใจไม่ชัดเจนในอิมเมจเพื่อที่จะให้เป็น Isotropic ตัว operator จะต้องเกี่ยวข้องกับ

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-7 Laplacian operator applied to image containing an edge at an angle (a) results in output (b).

เฉพาะอนุพันธ์ที่มี order เป็นเลขคู่ ผลการคำนวณจากการใช้ Isotropic operator กับอิมเมจและการหมุนเอ้าท์พุท จะเป็นเอกลักษณ์ (identical) กับการคำนวณ เมื่ออิมเมจถูกหมุนและเมื่อ operator ถูกนำมาใช้



รูปที่ 5-8 Results of Laplacian operator on four types of edges : positive edge (a), negative edge (b) , positive ramp (c) , and negative ramp (d).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

operator ที่ใช้ในการพัฒนาขอบภาพโดยปกติเกี่ยวข้องกับผลต่างกับ identity ของการเปลี่ยนแปลงความชันของค่ารวมกับค่าของพิกเซลเหมือนบริเวณผิว อนุพันธ์อันดับที่ 2 อาจจะประมาณโดยความแตกต่างอันดับหนึ่ง

$$\frac{\partial^2 f}{\partial x^2} = (f_{m,n} - f_{m+1,n}) - (f_{m+1,n} - f_{m+2,n})$$

$$\frac{\partial^2 f}{\partial x^2} = f_{m,n} - 2f_{m+1,n} + f_{m+2,n}$$

อินพุต a,b,c ประมาณของอนุพันธ์อันดับสองในทิศทาง x

$$\frac{\partial^2 f}{\partial x^2} = a - 2b + c$$

$$\begin{bmatrix} \text{Mask} \\ 1 \ -2 \ 1 \end{bmatrix}$$

การแสดง matrix ซึ่งจะใช้สำหรับ $f_{i,j}$ คือ

row : $i = i \text{ --- } > M$ (y direction)

Column : $j = j \text{ --- } > N$ (x direction)

MN matrix มี M แถว และ N หลัก

$$\begin{bmatrix} f_{1,1} & \dots & f_{1,N} \\ \vdots & & \vdots \\ f_{M,1} & \dots & f_{M,N} \end{bmatrix}$$

5.4.2 Robert's Gradient Operator

Robert's gradient operator (รูป 5-9) เป็นวิธีที่ง่ายกว่า Laplacian operator เพราะว่ามันกระทำบนพื้นที่ 2×2 ของพิกเซลของแต่ละจุด การใช้งานของข้อมูลจากขนาดพื้นที่น้อยกว่าพื้นที่ 3×3 ของ Laplacian operator จะลดเวลาในการคำนวณลง และความสามารถของระบบ operator ใช้กับอนุพันธ์ในแนวทะแยงเพื่อประมาณค่า gradient ที่เป็นจุดขนาดของ operator เท่ากับรากที่สองของผลรวมของผลต่างยกกำลังสอง มันสามารถประมาณโดยใช้ สมการคำนวณแบบง่าย ๆ ของผลรวมของค่าสมบูรณ์ของแต่ละผลต่างในแนวทะแยง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\text{Magnitude operator} = (\Delta 1^2 + \Delta 2^2)^{1/2}$$

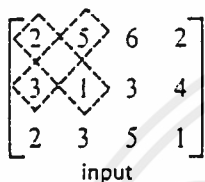
$$\text{Absolute value estimate} = |\Delta 1| + |\Delta 2|$$

where $\Delta 1 = p(i, j) - p(i+1, j+1)$ and $\Delta 2 = p(i+1, j) - p(i, j+1)$

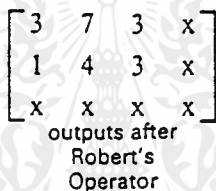


$$\text{New } q(1,1) = |\Delta 1| + |\Delta 2|$$

where $\Delta 1 = (a-d)$ and $\Delta 2 = (b-c)$



$$\text{New } q(1,1) = (2-1) + (5-3) = |1| + |2| = 3$$



รูปที่ 5-9 Robert's gradient operator.

5.4.3 Sobel Edge Detector Operator

Sobel Operator (ในรูป 5-10) เป็นการคำนวณแบบไม่เป็นเชิงเส้นของขนาดขอบภาพที่เป็นจุด แต่มันจะไม่ใช้ค่าที่จุดของมันเองในการคำนวณ ค่าของพิกเซลกำหนดให้โดย

5.4.4 Other Local Operator

วิธีการพัฒนาแทนในส่วนที่กล่าวมาก่อนหน้านั้นเป็นแบบพื้น ๆ และพบใน standard processing package ของโปรแกรมย่อย (software routines) ใช้กับระบบมองเห็นทางตัวเลข Convolution mask รูปแบบต่าง ๆ ดังแสดงไว้ให้ดูข้างนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

where

$$S = (\Delta x^2 + \Delta y^2)^{1/2}$$

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \quad \Delta x = (a + 2d + g) - (c + 2f + e)$$

$$\Delta y = (g + 2h + i) - (a + 2b + c)$$

input

$$\Delta x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad \Delta y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

convolution masks for sobel operator

$$\begin{bmatrix} 3 & 4 & 2 & 5 & 1 \\ 2 & \textcircled{1} & 6 & 4 & 2 \\ 3 & 5 & 7 & 1 & 3 \\ 4 & 2 & 5 & 7 & 1 \\ 2 & 5 & 1 & 3 & 2 \end{bmatrix} \quad \begin{matrix} x & x & x & x & x \\ x & q_{2,2} & & & \end{matrix}$$

input

$$q_{2,2} = (\Delta x^2 + \Delta y^2)^{1/2} \text{ around pixel } 2,2$$

$$\Delta x = 3 + 4 + 3 - 2 - 12 - 7 = +11$$

$$\Delta y = -3 - 8 - 2 + 3 + 10 + 7 = +17$$

$$q_{2,2}^2 = [(-11)^2 + (+17)^2]^{1/2}$$

$$= [121 + 49]^{1/2}$$

$$= (170)^{1/2} \approx 13$$

รูปที่ 5-10 Sobel edge detector operator.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Convolution mask

CONVOLUTION MASKS

1. Low pass filters
(coefficients positive and added to one)

$$\begin{bmatrix} 0.1 & 0.1 & 0.1 \\ 0.1 & 0.2 & 0.1 \\ 0.1 & 0.1 & 0.1 \end{bmatrix} \quad \begin{bmatrix} 1/16 & 1/8 & 1/16 \\ 1/8 & 1/4 & 1/8 \\ 1/16 & 1/8 & 1/16 \end{bmatrix}$$
2. High pass filters
(coefficients added to zero)

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$
3. High pass filter with DC Bias
(coefficients added to one)

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$
4. Gradient-directional
(coefficients added to zero)

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix}$$

N NE E
5. Shift and difference filters (coefficients added to zero)

$$\begin{bmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Vertical edges Horizontal edges Horizontal and vertical edges
6. Blurs

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Horizontal Vertical Diagonal
7. Difference filters

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix}$$

Vertical Horizontal
8. Vertical differentiation filters
Take absolute value difference (7).
9. Horizontal differencing and vertical smoothing filters

$$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$
10. Laplacian filters
(coefficients added to zero)

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$
11. Bright region expansion
Maximum of

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

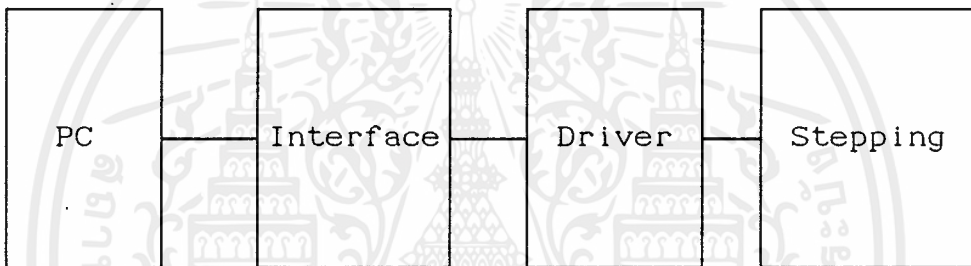
Convolution mask

12. Medium filter
(reduce camera noise) Fifth largest of $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$
13. Enhance line segment
(coefficients added to zero) $\begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix}$ Vertical $\begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix}$ Horizontal $\begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix}$ L-R Diagonal
14. Edge detector $\text{Max}(A) - (B)$
where: $A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ $B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
15. Line detector $q_{2,1} = A - \text{Min}(\text{Min}(\text{Max}(\text{Max}(B))))$
where: $A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ $B = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$
16. Template matching
(Cross correlation) $\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$

บทที่ 6

สเตปปีงมอเตอร์ (Stepping motor)

สเตปปีงมอเตอร์เป็นอุปกรณ์ จำพวกเชิงกลทางไฟฟ้าที่มีอินพุตเป็นกลุ่มของไบนารีโลเตจและเอาต์พุตเป็นลักษณะของการเคลื่อนที่แบบเชิงมุม หรือหมุนไปตามสเตป (แต่ละสเตปในช่วง 0.1 ถึง 90 องศา ขึ้นอยู่กับโครงสร้างของสเตปปีงมอเตอร์) ตามสัญญาณพัลส์ที่ป้อนให้กับขดสเตเตอร์ จึงจะเกิดแรงผลักดันต่อโรเตอร์ทำให้มอเตอร์หมุนไป แต่ลักษณะของสเตปปีงมอเตอร์จะมีขดของสเตเตอร์อยู่หลายขดซึ่งเรียกว่า "เฟส" นั้นเมื่อป้อนสัญญาณซึ่งเป็นพัลส์ในลักษณะซีควเอนซ์ของเลขไบนารีโดยผ่านวงจรไดรเวอร์ (Driver) จะทำให้โรเตอร์หมุนได้อย่างต่อเนื่องกันบล็อกไดอะแกรมรูปที่ 6.1



รูปที่ 6.1 บล็อกไดอะแกรมแสดงการควบคุมสเตปปีงมอเตอร์

คุณสมบัติของสเตปปีงมอเตอร์

ในระบบควบคุมตำแหน่งที่ใช้สเตปปีงมอเตอร์นั้น มีข้อดีอยู่หลายประการ คือ

1. เป็นลักษณะการควบคุมแบบไม่ต้องการการป้อนกลับไม่ว่าจะเป็นการควบคุมตำแหน่งหรือความเร็ว
2. ความผิดพลาดเกี่ยวกับตำแหน่งแทบไม่มีเลยเนื่องจากการเคลื่อนที่ของสเตปปีงมอเตอร์นั้นเคลื่อนที่เป็นสเตปด้วยจำนวนองศาที่มีค่าแน่นอน
3. สเตปปีงมอเตอร์ จะถูกนำมาใช้กับเครื่องมือที่ต้องการความละเอียดแม่นยำและใช้อยู่ในเครื่องมือประเภทดิจิทัล เช่น เครื่องวาดรูป เครื่องคอมพิวเตอร์ นิวเมอริกอลคอนโทรล (Computer Numerical Control) หรือ CNC
4. ไม่จำเป็นต้องใช้วงจรแปลงดิจิทัลเป็นอนาลอกเมื่ออินเตอร์เฟสกับไมโครคอมพิวเตอร์

5. สเตปปีงมอเตอร์ เป็นอุปกรณ์ที่เปลี่ยนสัญญาณดิจิทัลไปเป็นการเคลื่อน เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ทางกล ดังนั้นการติดต่อกับอุปกรณ์ดิจิทัลก็เป็นไปได้โดยง่าย และวงจรขยายกำลังจากสัญญาณดิจิทัล (digital power amplifier) ที่ใช้ก็มีราคาถูกกว่าวงจรขยายกำลังเชิงเส้น (Linear power amplifier) อีกด้วย

6. การออกแบบวงจรควบคุมสเตปป์มอเตอร์สามารถทำได้ง่ายกว่าวงจรควบคุมมอเตอร์แบบเซอร์โว และยังสามารถออกแบบวงจรในสเตปป์มอเตอร์ทำงานหรือหยุดได้แบบทันทีทันใด

โหมดการทำงานของ STEPPING MOTORS

ถ้าจะแบ่งโหมดการทำงานของ Stepping Motor ตามอัตราเร็วของ Step จะแบ่งออกได้เป็น 2 โหมด คือ หมุนเป็นสเตป (Discrete Stepping Mode) และหมุนแบบต่อเนื่อง (Slewing mode)

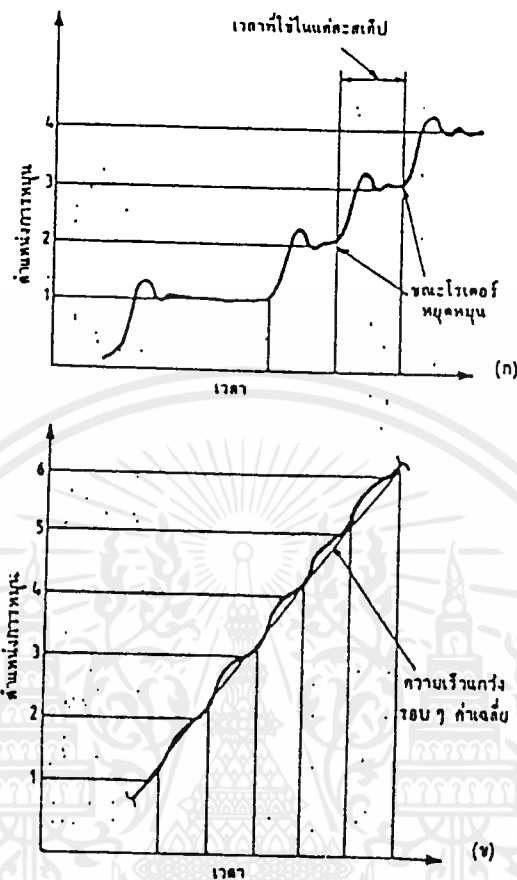
โดยถ้าการหมุนเป็นแบบสเตป และมีเวลาหยุดนิ่งก่อนที่จะเปลี่ยนเป็นสเตปต่อไป ก็จะเรียกการทำงานในโหมดนี้ว่าการหมุนเป็น Step ดังรูป 6.2(ก) สำหรับตัวอย่างของเครื่องใช้ที่ทำงานในโหมดนี้คือ เครื่องเจาะบัตร การทำงานต่าง ๆ ก็คือ Stepping Motor จะเป็นตัวส่งแถบกระดาษเข้าไปยังเครื่องปฏักกระดาษ เพื่อบันทึกข้อมูลลงในกระดาษ ซึ่งเป็นตัวส่งแถบกระดาษเข้าไปยังเครื่องปฏักกระดาษ เพื่อบันทึกข้อมูลลงในแถบกระดาษ ซึ่งการหมุนของ Stepping Motor จะหมุนไปแล้วหยุดชั่วขณะเพื่อปฏักกระดาษให้เรียบร้อยก่อน แล้วจึงค่อยหมุนต่อไปยังตำแหน่งเจาะใหญ่

ถ้าเพิ่มอัตราเร็วในแต่ละ Step ให้เร็วขึ้น และเป็นไปอย่างต่อเนื่องไม่มีการหยุดนิ่ง จะเรียกกำหนดการทำงานนี้ว่า การหมุนแบบต่อเนื่อง ดังแสดงในรูป 6.2(ข) ซึ่งสามารถหาความสัมพันธ์ระหว่างความเร็วรอบของมอเตอร์ (n) กับอัตราเร็วของ Step (f) และจำนวนสเตปทั้งหมด (s) ได้ดังสมการ

$$N = 60 \frac{f}{s}$$

กราฟคุณลักษณะของ STEPPING MOTOR

กราฟคุณลักษณะของ Stepping Motor จะเป็นการแสดงความสัมพันธ์ระหว่างอัตราเร็วของ Step และแรงบิดดังแสดงในรูปสำหรับกราฟเส้นประเรียกว่า Start-step curved หรือ single step load curve เป็นกราฟที่อยู่ในโหมดการเอ็กสตรีนเป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 6.2 กราฟแสดงสเต็ปของการหมุนในโหมดการทำงาน

(ก) หมุนแบบ Step

(ข) หมุนต่อเนื่อง

หมุนเป็น Step และเป็นกราฟที่แสดงถึงย่านของแรงบิดที่มอเตอร์สามารถเริ่ม และหยุด หมุนได้ โดยปราศจากความผิดพลาดแม้ที่อัตราเร็วของ Step ต่างๆ กัน และกราฟ อีกเส้นคือ slew curve ซึ่งทำงานอยู่ในโหมดการหมุนแบบต่อเนื่อง จะเป็นกราฟที่ แสดงถึงค่าแรงบิดสูงสุดที่ Stepping Motor สามารถจะกระทำได้ที่อัตราเร็วของ Step ต่างๆ กัน ถ้ามีการใช้งาน Stepping Motor โหมดกราฟนี้ก็อาจทำให้เกิดความผิดพลาดได้ในทางตรงกันข้าม ถ้าใช้งานอยู่ภายใต้กราฟนี้แม้จะควบคุมแบบระบบเปิดก็มั่นใจ ได้ว่าทั้งตำแหน่งและความเร็วมีความเที่ยงตรงแน่นอนของ Motor สามารถคำนวณได้จากสมการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

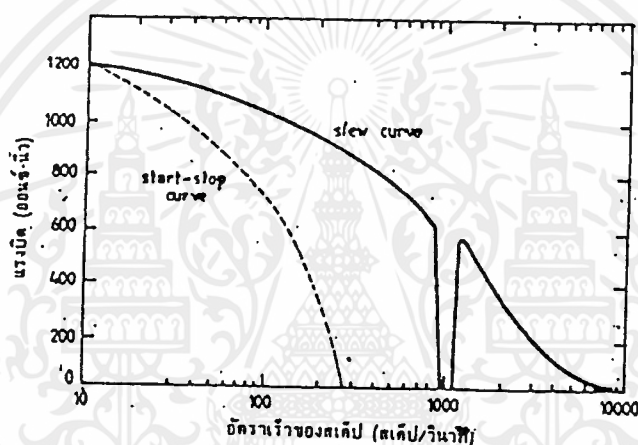
มุมที่เปลี่ยนไป = มุมใน 1 Step x จำนวนพัลส์ที่ป้อนให้

ส่วนความเร็วคำนวณหาได้จากสมการ

$$n = 60 \frac{r}{s}$$

s

สำหรับช่วงที่เส้นกราฟขาดหายไปของ slew curve นั้นเป็นย่านเรโซแนนซ์ (Resonant region) ซึ่งเป็นจุดอ่อนสำคัญที่ต้องคำนึงถึงในการพิจารณาใช้งาน เพราะเป็นช่วงที่ไม่เสถียร และควบคุมไม่ได้



รูปที่ 6.3 กราฟแสดงความสัมพันธ์ระหว่างอัตราเร็วของ Step กับแรงบิดของการทำงานทั้ง 2 โหมดของ Stepping Motor

โหมดของการกระตุ้นเฟส (MODES OF EXCITATION)

การใช้งาน Stepping Motor อย่างเป็นต่อเนื่อง จำเป็นต้องจ่ายพัลส์เป็นสำคัญอย่างต่อเนื่อง วิธีการกระตุ้นเฟสมีหลายอย่างดังนี้

1. การกระตุ้นแบบเฟสเดี่ยว (Single - phase excitation)

ตาราง 6.1 แสดงลำดับพของการกระตุ้นแบบเฟสเดี่ยวสำหรับ MOTOR แบบ 3 และ 4 เฟสจากตารางเมื่อต้องการให้มอเตอร์หมุนตามเข็มนาฬิกาสามารถทำได้โดยกระตุ้นเฟส ph-1ph2-ph3 ตามลำดับในทางกลับกันถ้าต้องการทวนเข็มนาฬิกากระตุ้น ph3-ph2-ph1 ตามลำดับเช่นเดียวกับการกระตุ้น ลักษณะนี้เป็นลักษณะ "ONE PHASE ONE DRIVE"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Excitation sequence in the single-phase-on operation.

(1) Three-phase motor

	R	1	2	3	4	5	6	7	8
Phase 1									
Phase 2									
Phase 3									

(2) Four-phase motor

	R	1	2	3	4	5	6	7	8
Phase 1									
Phase 2									
Phase 3									
Phase 4									

Pulses

Note: Symbol R indicates 'reset'.

ตารางรูป 6.1 แสดงลำดับการกระตุ้นแบบเฟสเดี่ยว

2. การกระตุ้นแบบเฟสคู่ (Two-phase excitation)

ตารางที่ 6.2 แสดงลำดับการกระตุ้นแบบเฟสคู่สำหรับ Motor

แบบ 3 และ 4 เฟส จากตารางจะพบว่าเมื่อ ph2 off และ ph1 on จะพบว่า ph3 ยังคงถูกกระตุ้นอยู่ ลักษณะของการกระตุ้นแบบนี้เป็นดังรูป 6.4

Excitation sequence in the two-phase-on operation.

(1) Three-phase motor

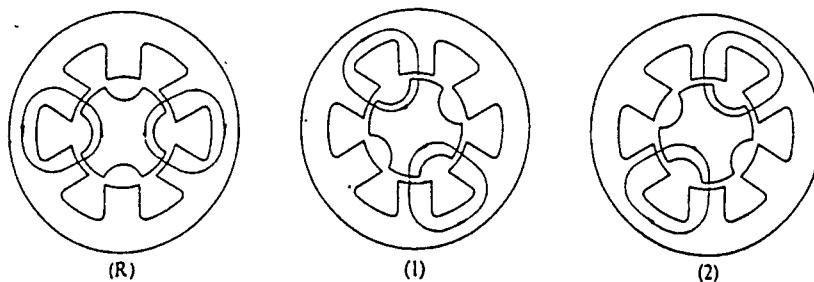
Clock state	R	1	2	3	4	5	6	7	8
Phase 1									
Phase 2									
Phase 3									

(2) Four-phase motor

Clock state	R	1	2	3	4	5	6	7	8
Phase 1									
Phase 2									
Phase 3									
Phase 4									

ตาราง 6.2 แสดงลำดับการกระตุ้นแบบเฟสคู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Change in field pattern as a step proceeds in the two-phase-on mode in a monofilar three-phase VR motor.

รูปที่ 6.4 ลักษณะของการเปลี่ยนเฟสของการกระตุ้นแบบเฟสคู่

3. การกระตุ้นแบบกึ่งสเตป (Half-step mode)

เป็นการรวมรูปแบบการกระตุ้น ระหว่างการกระตุ้นแบบเฟสเดี่ยวและเฟสคู่ ตารางที่ 6.3 เป็นการแสดงลำดับการกระตุ้นแบบกึ่งสเตป การกระตุ้นจะต้องจ่ายสัญญาณพิก้า 2 State คือ A, B โดยที่ A จะจ่ายลักษณะกระตุ้นเฟสเดี่ยวและ B เป็นการจ่าย Clock ในลักษณะกระตุ้นแบบเฟสคู่

Excitation sequence in the half-step operation (three-phase motor).

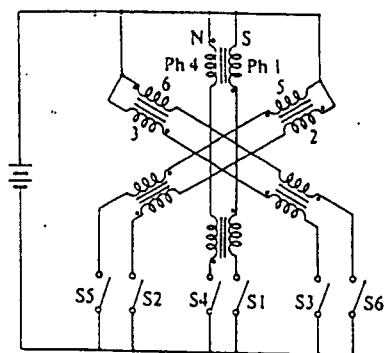
Clock state (A)	R	1	2	3	4	5				
Clock state (B)	R	1	2	3	4	5	6	7	8	9
Phase 1										
Phase 2										
Phase 3										

ตาราง 6.3 แสดงลำดับการกระตุ้นแบบกึ่ง Step

4. Two-phase-on drive of bifilar-wound three phase VR motor

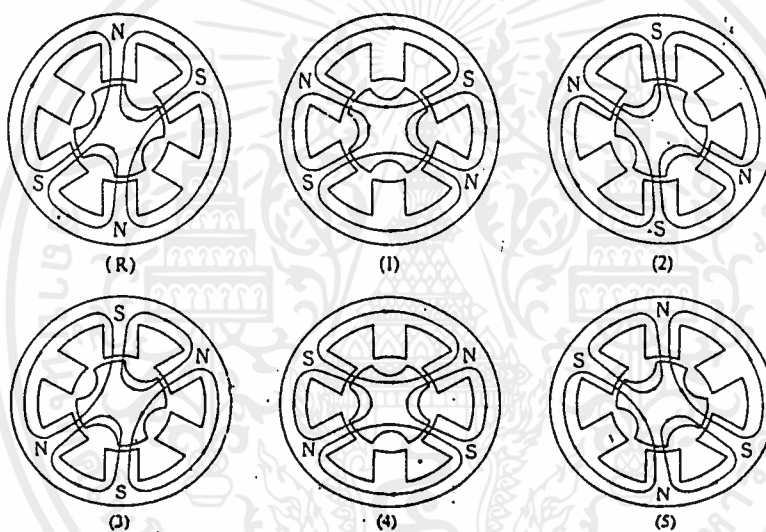
ความต้องการสำคัญในการออกแบบ Motor คือ ต้องการให้ขนาดของเครื่องจักรเล็ก และถ้าเป็นไปได้จะต้องมีความสามารถทำงานพิเศษ ดังนั้นจึงทำให้เกิดการกระตุ้นแบบนี้ขึ้น รูป 6.5 เป็นลักษณะไดอะแกรมของ bifilar wound coil ใน VR Motor 3 เฟสในรูป 6.6 แสดงลักษณะของฟลักซ์แม่เหล็กเมื่อมีการกระตุ้น ตามตาราง 6.4 ข้อดีของการกระตุ้นแบบนี้คือ แรงบิดของเครื่องจักรจะดีมาก

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ยืมได้เห็นใบใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Schematic diagram of the bifilar-wound coils in the three-phase VR motor.

รูปที่ 6.5 ลักษณะไดอะแกรมของ bifilar wound coil ใน VR Motor 3 เฟส



Field pattern in the bifilar-wound three-phase motor. Symbol (R) denotes Reset in the table in Fig. 5.31 (p. 141).

รูปที่ 6.6 แสดงลักษณะของฟลักซ์แม่เหล็ก

Excitation sequence in the two-phase-on operation for a bifilar-wound three-phase VR motor.

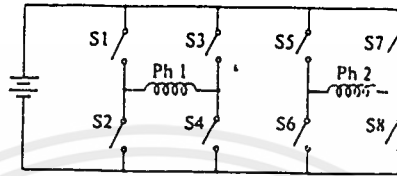
Clock state	R	1	2	3	4	5	6	7	8
Ph1 (S1)									
2 (S2)									
3 (S3)									
4 (S4)									
5 (S5)									
6 (S6)									

ตาราง 6.4 แสดงลำดับการกระตุ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. การกระตุ้นด้วยวงจรบริดจ์

วงจรบริดจ์แสดงในรูป 6.7 เป็นวงจรสำหรับขับมอเตอร์ใน bipolar mode ลักษณะการกระตุ้นวงจรตามลำดับในตาราง 6.5 จากตารางและวงจรที่พบในขดลวดแต่ละขด จะนำกระแสสลับอย่างสม่ำเสมอ ซึ่งลักษณะนี้จะปรับปรุง 25-30% ของแรงบิดต่อหน่วยของ I/P ซึ่งเป็นไปได้เมื่อใช้กับการไคร์ฟ bipolar ลักษณะของ Voltage ที่ป้อนให้ วงจร Bipolar drive แสดงในรูป 6.8



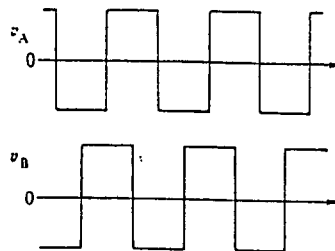
Bridge driver scheme for a two-phase stepping motor.

รูป 6.7 วงจร Bridge drive สำหรับ Stepping Motor 2 เฟส

Excitation sequence in the bridge operation for a two-phase motor.

Clock state	R	1	2	3	4	5	6	7	8
S1									
S2									
S3									
S4									
S5									
S6									
S7									
S8									

ตาราง 6.5 แสดงลำดับการกระตุ้นสำหรับวงจรบริดจ์



Square-wave voltage for the bipolar drive.

รูป 6.8 Squar-wave Voltage สำหรับวงจรไคร์ฟ Bipolar

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

การจัดแอดเดรสสำหรับหน่วยความจำและ I/O ของเครื่อง PC

ในบทนี้จะกล่าวถึงการจัดแอดเดรสสำหรับหน่วยความจำและพอร์ท I/O ต่างๆ ภายใน IBM/PC ซึ่งจะแสดงถึงแอดเดรสต่างๆ ที่ถูกใช้งานโดยพอร์ท I/O <Input/Output Port> และ หน่วยความจำ นอกจากนี้จะได้กล่าวถึงเทคนิคการดีโค้ด <Decode> แอดเดรสในแบบต่างๆ ด้วย

การจัดแอดเดรสสำหรับพอร์ท I/O ใน IBM/PC

ในหัวข้อนี้จะกล่าวถึงวิธีการอ้างและใช้งานแอดเดรสต่าง ๆ ของพอร์ท I/O ที่ใช้งานอยู่ใน IBM/PC

การอ้างแอดเดรสของพอร์ท I/O

ในการควบคุมและตรวจสอบสภาวะการทำงานรวมทั้งการอ่านข้อมูลจากอุปกรณ์ที่เป็นชิพพอร์ทหรือการ์ดต่างๆ ที่ใช้ในระบบของ IBM/PC นั้น จะกระทำโดยผ่านทางพอร์ท I/O ของระบบ ดังนั้นในการที่จะใช้งานหรือควบคุมการทำงานของอุปกรณ์เหล่านี้ จึงจำเป็นต้องศึกษาถึงวิธีการควบคุมพอร์ท I/O ต่างๆ ของระบบด้วย และเนื่องจากการควบคุมหรือติดต่อกับพอร์ทเหล่านี้ต้องกระทำโดยการอ้างถึงแอดเดรสของพอร์ท I/O เหล่านั้นโดยตรง เราจึงจำเป็นต้องศึกษาถึงหลักการอ้างแอดเดรสของ 8088 ใน IBM/PC ด้วย

สำหรับแอดเดรสของพอร์ท I/O ต่าง ๆ นั้น จะเป็นแอดเดรสที่ถูกสร้างขึ้นโดย 8088 ซึ่งแอดเดรสเหล่านี้เป็นแอดเดรสที่จัดไว้สำหรับพอร์ท I/O โดยเฉพาะ คือ แยกจากแอดเดรสของหน่วยความจำโดยเด็ดขาด ส่วนการส่งข้อมูลให้กับพอร์ทเหล่านี้จะทำได้โดยการใช้คำสั่ง OUT ของ 8088 ส่งข้อมูลนั้นไปยังแอดเดรสของพอร์ทที่ต้องการ และสำหรับการตรวจสอบหรือการอ่านข้อมูลจากพอร์ท ก็จะได้ทำได้โดยการใช้คำสั่ง IN ของ 8088 อ่านข้อมูลจากแอดเดรสของพอร์ทที่ต้องการเช่นกัน

ภายในไมโครโพรเซสเซอร์เบอร์ 8068 นี้จะมีแอดเดรสสำหรับใช้กับพอร์ท I/O อยู่ทั้งสิ้น 65,536 หรือ 64K แอดเดรส (ในขณะที่มีแอดเดรสสำหรับหน่วยความจำอยู่ 1 Mbyte) ซึ่งทำให้การอ้างแอดเดรสของพอร์ท I/O ที่ทำงานร่วมกับ 8088 นั้น ต้องใช้จำนวนเส้นแอดเดรสในบัสแอดเดรสในบัสแอดเดรสทั้งสิ้น 16 เส้น คือ A0-A15 แต่สำหรับใน IBM/PC นี้ถูกออกแบบมาให้ใช้เส้นแอดเดรสเฉพาะ 10 เส้นล่างคือ A0-A9

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เท่านั้น ดังนั้นในการอ้างถึงแอดเดรสของพอร์ทของอุปกรณ์ หรือชิพพอร์ทใด ๆ ที่ใช้ร่วมกับ IBM/PC จึงใช้จำนวนเส้นแอดเดรสเพียง 10 เส้นด้วย โดยเส้นแอดเดรสที่เหลือคือ A10-A15 นั้นจะไม่ถูกนำไปใช้งาน อย่างไรก็ตามถึงแม้ว่าเส้นแอดเดรส A10-A15 นี้จะไม่ถูกนำไปใช้งาน แต่ค่าแอดเดรสบนเส้นแอดเดรสเหล่านี้ยังคงเปลี่ยนแปลงตามค่าแอดเดรสของพอร์ทที่กำหนดไว้ในคำสั่ง OUT หรือ IN อยู่ด้วย เพียงแต่ไม่ได้ถูกนำมาตีโค้ดร่วมกับแอดเดรส A0-A9 เท่านั้น ตัวอย่างเช่น ในการใช้คำสั่ง OUT ส่งข้อมูลไปยังพอร์ทที่ตรงกับแอดเดรส 0010H นั้นจะให้ผลเหมือนกับการส่งข้อมูลไปยังพอร์ทที่ตรงกับแอดเดรส 0401H, 0810H, 0C10H, ทั้งนี้ เนื่องจากแอดเดรส 6 บิตบนไม่ได้ถูกใช้งาน จึงทำให้การเปลี่ยนแปลงค่าแอดเดรสบนเส้นแอดเดรส A10-A15 นั้นไม่ทำให้เกิดความแตกต่างใด ๆ ขึ้น

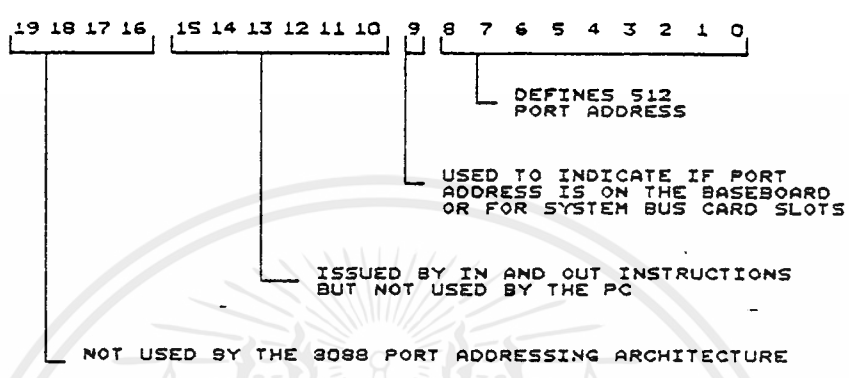
เนื่องจากใน IBM/PC ได้ใช้งานเส้นแอดเดรสเพียง 10 เส้น (คือ A0-A9) ดังนั้นจึงสามารถที่จะอ้างแอดเดรสของพอร์ทได้สูงสุดเพียง 1024 พอร์ท (จากจำนวน 64K พอร์ท) เท่านั้น นอกจากนี้ในกรณีที่เป็นกรอ่านข้อมูลจากพอร์ทของ IBM/PC ข้อมูลในบิต A9 จะถูกจัดให้มีหน้าที่ในการแบ่งพอร์ททั้ง 1024 พอร์ทออกเป็น 2 ส่วน (ส่วนละ 512 พอร์ท) อีกด้วย กล่าวคือถ้าข้อมูลในบิต A9 เป็น "0" แล้ว เราจะทำกรอ่านข้อมูลได้เฉพาะจากพอร์ทของอุปกรณ์หรือชิพพอร์ทต่างๆ ที่อยู่บนเมนบอร์ด <Main Board> ของ IBM/PC เช่น 8253-5, 8237-5 หรือ 8259A เท่านั้น แต่ถ้าข้อมูลในบิต A9 นี้เป็น "1" ก็จะทำกรอ่านข้อมูลได้เฉพาะจากพอร์ทที่อยู่บนการ์ดต่างๆ เท่านั้น

จากที่ได้กล่าวมานั้นจะสรุปได้ว่าพอร์ทบน IBM/PC ทั้ง 1024 พอร์ท ถูกแบ่งออกเป็น 2 กลุ่มโดยที่กลุ่มแรกเป็นกลุ่มของพอร์ทที่อยู่บนเมนบอร์ด และกลุ่มที่สองเป็นกลุ่มที่จัดเตรียมไว้สำหรับพอร์ทที่อยู่บนการ์ดต่างๆ

สำหรับในกรณีของการส่งข้อมูลให้กับพอร์ททั้ง 1024 พอร์ท เราสามารถที่จะเลือกส่งไปยังพอร์ทใด ๆ ใน IBM/PC ได้ ดังนั้นการเลือกแอดเดรสสำหรับพอร์ทที่อยู่บนการ์ดจึงสามารถทำได้โดยสะดวก แต่อย่างไรก็ตามสิ่งหนึ่งที่จะต้องคำนึงถึงก็คือถ้าแอดเดรสที่เราเลือกให้กับพอร์ทนี้ตรงกับค่าแอดเดรสเดิมที่มีอยู่บนเมนบอร์ดแล้ว เมื่อเราทำการส่งข้อมูลให้กับพอร์ทที่อยู่ในตำแหน่งแอดเดรสนี้ ก็จะเท่ากับเป็นการส่งข้อมูลให้กับทั้งพอร์ทที่อยู่บนเมนบอร์ดและพอร์ทที่อยู่บนการ์ดด้วย ซึ่งในกรณีเช่นนี้อาจจะก่อให้เกิดความผิดพลาดขึ้นได้เช่นกัน ดังนั้นในการกำหนดค่าแอดเดรสให้กับพอร์ทที่ถูกสร้างขึ้นบนการ์ดต่างๆ จึงควรจะใช้ค่าแอดเดรสที่แอดเดรสบิต A9 มีค่าเป็น "1" คือ แอดเดรส OFE00H จนถึง OFFF0H เท่านั้น (แอดเดรสบิต A10-A15 ไม่ถูกใช้ในการตีโค้ดแต่เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อความสะดวกจึงกำหนดให้มีค่าเป็น "1" ในฐานสองทั้งหมด แต่ในการใช้งานจริงอาจเปลี่ยนให้แอดเดรส A10-A15 แต่ละบิตมีค่าเป็น "1" หรือ "0" ก็ได้)

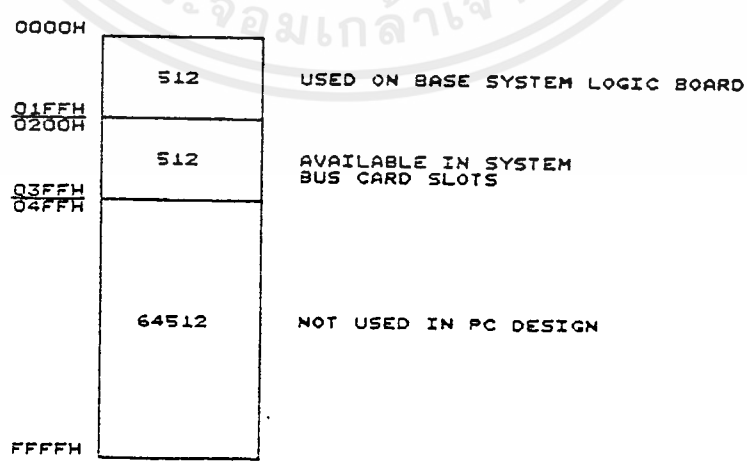
สำหรับรูปที่ 7.1 นี้ จะแสดงถึงการใช้งานแอดเดรสบิตต่างๆ ในการอ้างแอดเดรสของพอร์ทใน IBM/PC



รูปที่ 7.1 การใช้แอดเดรสบิตต่าง ๆ ในการอ้างแอดเดรสของพอร์ทใน IBM/PC

การใช้งานแอดเดรสสำหรับพอร์ท I/O ใน IBM/PC

จากที่ได้กล่าวไว้ในหัวข้อที่ผ่านมานั้นพอร์ท I/O ทั้ง 1024 พอร์ทใน IBM/PC จะถูกแบ่งออกเป็น 2 กลุ่ม ๆ ละ 512 พอร์ท สำหรับในหัวข้อนี้จะกล่าวถึงการใช้งานพอร์ทต่าง ๆ เหล่านี้โดยจะแบ่งเป็น 2 กลุ่มตามที่ได้อธิบายไว้ในหัวข้อที่ผ่านมามา ดังนี้

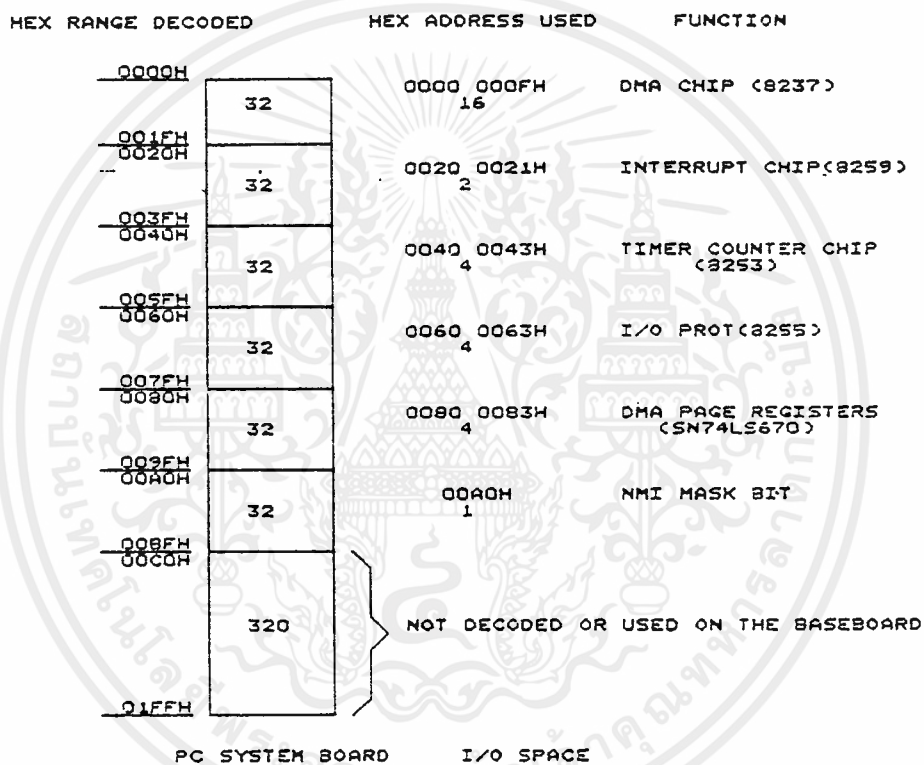


รูปที่ 7.2 การใช้งานแอดเดรสของพอร์ทบน IBM/PC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ในกลุ่มแรกนี้เป็นกลุ่มของพอร์ท I/O ที่อยู่บนเมนบอร์ด IBM/PC ซึ่งจะมีแอดเดรสอยู่ในตำแหน่ง 0000H จนถึง 01FFH (ขอให้ระลึกอยู่เสมอว่า A10-A15 นั้นไม่ถูกใช้งาน) หรือแอดเดรสที่มีบิต A9 เป็น "0" นั่นเอง

สำหรับแอดเดรสของพอร์ท I/O ในกลุ่มนี้จะใช้ในการอ้างแอดเดรสของชิพพอร์ทและอุปกรณ์ที่เป็น I/O ต่างๆ บนเมนบอร์ดของ IBM/PC เช่น แอดเดรส 0000H จนถึง 000FH จะถูกใช้เป็นแอดเดรสสำหรับ 8237-5 DMA Controller เป็นต้น ในรูปที่ 7.3 จะแสดงถึงการใช้งานแอดเดรสต่าง ๆ ตั้งแต่ 0000H จนถึง 01FFH ในการอ้างแอดเดรสของชิพพอร์ทและอุปกรณ์ต่าง ๆ ที่ทำหน้าที่เป็น I/O บนเมนบอร์ดของ IBM/PC



รูปที่ 7.3 การใช้งานแอดเดรสต่างๆ สำหรับพอร์ท I/O ของ IBM/PC

จากรูปข้างต้นจะเห็นว่าแอดเดรส 00C0H จนถึงแอดเดรส 01FFH นั้นไม่ได้ถูกใช้งานบนเมนบอร์ดของ IBM/PC ดังนั้นในกรณีนี้เราก็สามารถที่จะใช้งานแอดเดรสต่าง ๆ เหล่านี้ได้ แต่อย่างไรก็ตามแอดเดรสเหล่านี้ยังคงถูกตีโค้ดให้เป็นแอดเดรสที่ใช้ในการอ่านข้อมูลจาก พอร์ท I/O บนเมนบอร์ดเท่านั้น ดังนั้นการใช้ค่าแอดเดรส 00C0H-01FFH กับพอร์ท I/O I/O บนการ์ดหรือวงจรรีโมตที่เรารวบรวมขึ้นนั้น ต้องเป็นพอร์ทเอาต์พุตเพียงชนิดเดียวเท่านั้น กล่าวคือ จะทำการอ่านข้อมูลจากพอร์ท I/O (ที่ไม่ได้อยู่บนเมนบอร์ด) ที่มีค่าแอดเดรสอยู่ในช่วง 00C0H-01FFH ไม่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ในกลุ่มที่สองนี้ จะเป็นกลุ่มของพอร์ท I/O ที่ถูกใช้งานอยู่บนการ์ดที่ใช้เสียบบนสล็อตต่างๆ ของ IBM/PC สำหรับแอดเดรสของพอร์ทเหล่านี้จะเริ่มต้นจากแอดเดรส 0200H จนถึง 03FFH ซึ่งก็คือแอดเดรสที่มีบิต A9 เป็น "1" นั่นเอง สำหรับการใช้งานแอดเดรสของพอร์ท I/O ในกลุ่มนี้จะแสดงได้ดังรูป 7.4

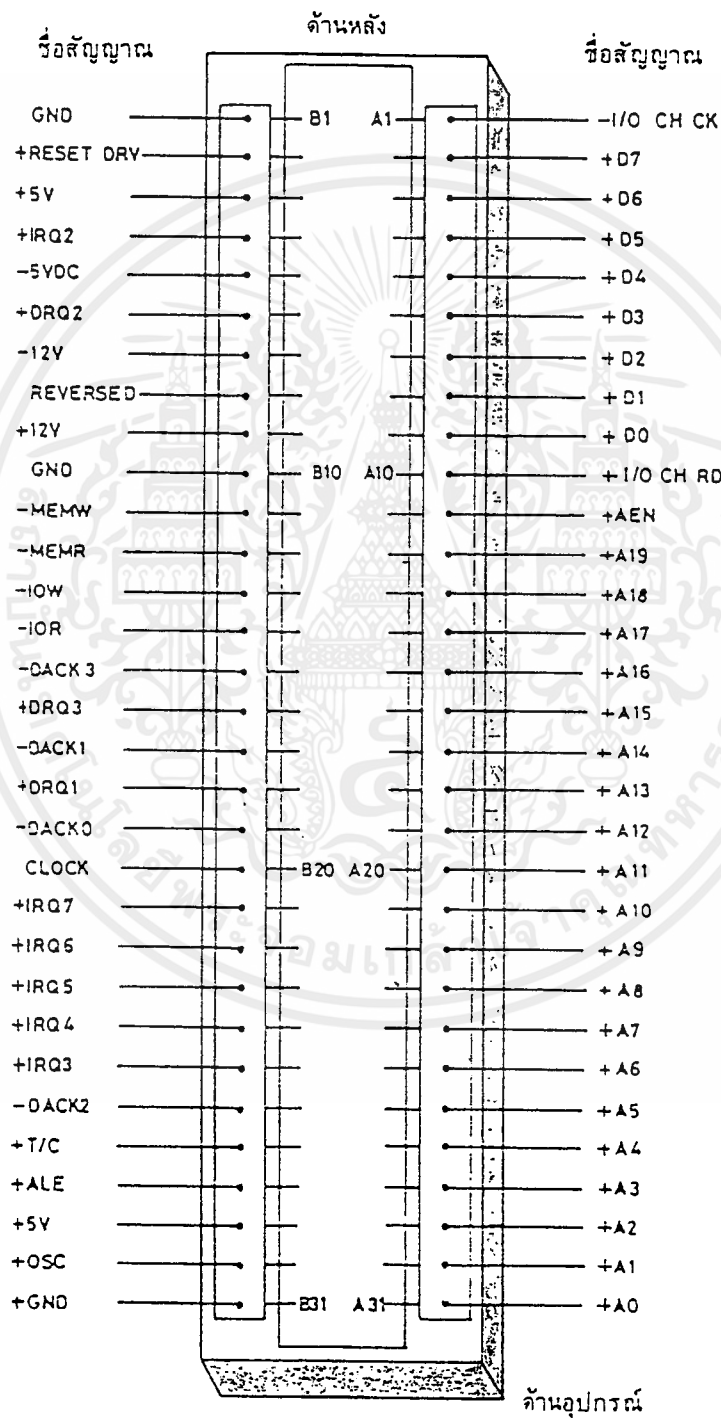
USES		
0200H 0201H 0202H	1	NOT USED
	1	GAME CONTROL ADAPTER
	119	NOT USED
0277H 0278H 027FH 0280H	8	SECOND PRINTER PORT ADAPTER
	120	NOT USED
02F7H 02F8H 02FFH 0300H	8	SECOND SERIAL PORT
	120	NOT USED
0377H 0378H 037FH 0380H	8	PRINTER PORT ADAPTER
	48	NOT USED
03AFH 0380H	16	MONOCHROME AND PRINTER ADAPTER
03BFH 03C0H	16	NOT USED
03CFH 03D0H	16	COLOR GRAPHICS
03DFH 03E0H	16	NOT USED
03EFH 03F0H	8	5 1/4" DISK DRIVE
03F7H 03F8H 03FFH	8	SERIAL PORT ADAPTER

รูปที่ 7.4 การใช้งานแอดเดรสสำหรับพอร์ท I/O บนการ์ดต่างๆ

อย่างไรก็ตามการใช้งานแอดเดรสในกลุ่มนี้อาจจะเปลี่ยนแปลงไปได้ ทั้งนี้ขึ้นอยู่กับการใช้งานการ์ดต่าง ๆ ร่วมกับ IBM/PC โดยการ์ดที่ถูกออกแบบผลิตขึ้นใหม่นั้นอาจจะใช้ค่าแอดเดรสต่างๆ ที่เหลืออยู่นี้ได้ ดังนั้นก่อนที่จะทำการออกแบบวงจรอินเทอร์เฟสที่จำเป็นต้องใช้ค่าแอดเดรสสำหรับพอร์ท I/O จึงควรจะตรวจสอบดูก่อนว่าการ์ดต่างๆ ที่ใช้อยู่ในระบบของ IBM/PC ที่เราใช้งานอยู่นั้นมีคาร์ดิโคบ้าง และการ์ดเหล่านั้นใช้งานแอดเดรสใดบ้าง จากนั้นจึงทำการออกแบบวงจรอินเทอร์เฟสโดยเลือกใช้เฉพาะแอดเดรสที่ยังไม่ถูกใช้งาน

สัญญาณต่าง ๆ บนสล็อตของ IBM/PC

ซึ่งแต่ละสล็อตจะมีจำนวนขาทั้งสิ้น 62 ขา แบ่งออกเป็น 2 ข้าง ๆ ละ 31 ขา ส่วนการเรียกตำแหน่งขาของสล็อตเหล่านี้จะขึ้นอยู่กับว่าขานั้นอยู่ข้างใด (ซ้ายหรือขวา) ของสล็อต โดยขาที่อยู่ทางด้านซ้ายของสล็อตจะเรียกโดยใช้อักษร "B" นำหน้าเลขตำแหน่งของขา (นับจากทางด้านท้ายของเครื่อง) ส่วนขาที่อยู่ทางด้านขวาของสล็อตจะเรียกโดยใช้อักษร "A" นำหน้าเลขตำแหน่งของขา (นับจากทางด้านท้ายของเครื่อง)



รูปที่ 7.5 แสดงสายสัญญาณเชื่อมต่อที่สล็อต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดเกี่ยวกับสัญญาณ

A0-A19 <Address Bus; ขา A31-A12> :

ขาสัญญาณทั้ง 20 ขานี้เป็นเอาต์พุต ซึ่งใช้สำหรับกำหนดแอดเดรสของหน่วยความจำ หรือ อุปกรณ์ที่ I/O ที่ 8088 ต้องการติดต่อกับ โดยที่สัญญาณ A0 จะมีนัยสำคัญต่ำสุด <Least Significant Bit> และ A19 จะมีนัยสำคัญสูงสุด <Most Significant Bit> สำหรับค่าแอดเดรสบนบัสแอดเดรส A0-A19 นี้ จะถูกกำหนดโดย 8088 ในระหว่างขบวนการอ่าน/เขียนข้อมูลลงในหน่วยความจำหรืออุปกรณ์ I/O แต่ในช่วงของขบวนการ DMA นั้น DMA-Controller จะเป็นผู้กำหนดค่าแอดเดรสบนบัสแอดเดรสเอง (ในระหว่างนี้ 8088 จะถูกตัดออกจากระบบ)

จะเห็นได้ว่าจำนวนเส้นแอดเดรสจะมีอยู่ 20 เส้น ซึ่งสามารถที่จะอ้างแอดเดรสของหน่วยความจำได้ถึง 1 Mbyte แต่อย่างไรก็ตามจะมีแอดเดรสบางแอดเดรสที่ถูกใช้งานโดย IBM/PC อยู่ก่อนแล้ว คือ แอดเดรสของหน่วยความจำ RAM บนเมนบอร์ด ที่ถูกใช้โดยระบบ จำนวน 64 Kbyte (สำหรับ IBM PC/XT จะเป็นจำนวน 25 Kbyte) และ แอดเดรสสำหรับหน่วยความจำ ROM อีก 48Kbyte ซึ่งถูกจัดในช่วงของแอดเดรสบนบัสใน 1 Mbyte คือ 0FC00H จนถึง 0FFFFH (สำหรับ IBM PC/XT จะเป็น 64 Kbyte)

สำหรับการอ้างแอดเดรสของพอร์ท I/O นั้น จะใช้เส้นแอดเดรสเพียง 16 เส้น คือ A0 - A15 ซึ่งจะทำให้อ้างแอดเดรสของพอร์ทได้ 64K พอร์ท โดยผ่านทางชุดคำสั่ง IN และ OUT ส่วนเส้นแอดเดรสที่เหลือคือ A16-A19 นั้นจะไม่ถูกใช้งาน อย่างไรก็ตามภายใน IBM/PC จะใช้เส้นแอดเดรสในการอ้างแอดเดรสของพอร์ทเพียง 10 เส้น คือจาก A0-A9 และค่าแอดเดรสที่ใช้งานจะต้องอยู่ในช่วง 0200H จนถึง 13FFH เท่านั้น

D0-D7 <Data Bus; ขา A9-A2> :

ขาสัญญาณนี้จะเป็นแบบ Bi-Directional ซึ่งต่อกับบัสข้อมูลของระบบ เพื่อทำหน้าที่ในการส่งผ่านข้อมูลระหว่างพอร์ท I/O กับ IBM/PC โดยบิต D0 จะมีนัยสำคัญต่ำสุด และบิต D7 จะมีนัยสำคัญสูงสุด

สำหรับในบัสไซเคิลของการเขียนข้อมูลที่สร้างขึ้นโดย 8088 นั้นข้อมูลจะถูกส่งออกมาบนบัสของมูล ก่อนที่สัญญาณ IOW (ในกรณีที่ต้องการส่งข้อมูลให้กับพอร์ท) หรือเอกสารเป็นเอกสารสองหน้าไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MEMW (ในกรณีที่ต้องการส่งข้อมูลให้กับหน่วยความจำ) จะเปลี่ยนจากลอจิก "0" เป็นลอจิก "1" (ขอขาขึ้น) ซึ่งโดยทั่วไปขอขาขึ้นของสัญญาณ IOW หรือ MEMW นี้จะถูกใช้เพื่อส่งให้พอร์ท I/O หรือหน่วยความจำที่มีแอดเดรสตรงกับค่าแอดเดรสบนบัสแอดเดรสนั้นรับข้อมูลไปเก็บไว้

สำหรับในบัสไซเคิลของการอ่านข้อมูลที่สร้างขึ้นโดย 8088 นั้น พอร์ท I/O หรือหน่วยความจำที่ถูกอ้างถึงจะต้องส่งข้อมูลออกมาบนบัสข้อมูล ก่อนที่สัญญาณ IOR (ในกรณีที่ต้องการอ่านข้อมูลจากพอร์ท) หรือ MEMR (ในกรณีที่ต้องการอ่านข้อมูลจากหน่วยความจำ) จะเปลี่ยนจากลอจิก "0" เป็น "1" (ขอขาขึ้น)

IOR <I/O Read; ขา B14> :

ขาสัญญาณนี้เป็นเอาต์พุตแอคทิฟที่ลอจิก "0" ที่สร้างขึ้นโดย 8288 Bus Controller เพื่อใช้ในการแสดงว่าบัสไซเคิลที่เกิดขึ้นนี้ เป็นบัสไซเคิลของการอ่านข้อมูลจากพอร์ท I/O เพื่อให้พอร์ท I/O ที่มีแอดเดรสตรงกับแอดเดรสบนบัสแอดเดรสนั้นส่งข้อมูลออกมาบนบัสข้อมูล โดยข้อมูลจะต้องถูกส่งออกมาบนบัสข้อมูลก่อนขอขาขึ้นของสัญญาณ IOR ประมาณ 30 nanosec เพื่อให้มั่นใจได้ว่า 8088 สามารถรับข้อมูลได้ถูกต้อง สำหรับในขบวนการ DMA 8237A DMA Controller จะทำการสร้างสัญญาณ IOR เอง โดยที่ค่าแอดเดรสที่อยู่บนแอดเดรสจะเป็นค่าแอดเดรสของหน่วยความจำ (แทนที่จะเป็นแอดเดรสของพอร์ท I/O) ที่พอร์ท I/O ที่ขอ DMA ต้องการจะนำข้อมูลไปเก็บไว้ที่พอร์ทใดจะส่งข้อมูลออกมาบนบัสข้อมูลนั้น จะอาศัยสัญญาณ DACK จาก DMA Controller เป็นตัวกำหนดเช่นกรณีที่สัญญาณ DACK1 แอคทิฟก็จะแสดงว่าพอร์ท I/O ที่จะต้องส่งข้อมูลออกมาบนบัสข้อมูลก็คือพอร์ท I/O ที่ขอ DMA ผ่านทางแชนแนลที่ 1 <DRQ1> เป็นต้น

IOW <I/O Write; ขา B13> :

ขาสัญญาณนี้เป็นเอาต์แอคทิฟที่ลอจิก "0" ซึ่งถูกสร้างขึ้นโดย 8288 Bus Controller เพื่อใช้แสดงว่าบัสไซเคิลที่เกิดขึ้นนี้เป็นบัสไซเคิลของการเขียนข้อมูลลงบนพอร์ท I/O เพื่อให้พอร์ท I/O ที่มีแอดเดรสตรงกับแอดเดรสบนบัสแอดเดรสนั้น รับข้อมูลที่อยู่บนบัสข้อมูลไปเก็บไว้ อย่างไรก็ตามเนื่องจากในช่วงเวลาที่สัญญาณ IOW นี้แอคทิฟ (ลอจิก "0") นั้นข้อมูลบนบัสข้อมูลอาจจะยังไม่สมบูรณ์ ดังนั้นในการออกแบบจึงควรจะใช้ขอขาขึ้นของสัญญาณ IOW แทนขอขาลงในการทำให้พอร์ท I/O ที่เกี่ยวข้องรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลไปเก็บไว้ เพื่อให้ข้อมูลบนบัสข้อมูลสมบูรณ์เสียก่อน สำหรับในขบวนการ DMA นั้น DMA-Controller จะทำการสร้างสัญญาณ IOW เอง โดยที่ค่าแอดเดรสที่อยู่บนบัส แอดเดรสจะเป็นค่าแอดเดรสของหน่วยความจำที่พอร์ท I/O ที่ขอ DMA ต้องการจะอ่านข้อมูล)

AEN <Address Enable; ขา A11> :

สัญญาณนี้เป็นเอาต์พุตที่ใช้ในการแสดงว่า บัสไซเคิลที่เกิดขึ้นในช่วงเวลาที่สัญญาณ AEN แอคทีฟ (ลอจิก " 1 ") นั้น เป็นบัสไซเคิลของขบวนการ DMA

สำหรับบนเมนบอร์ดของ IBM/PC นั้น จะใช้สัญญาณนี้ในการดิสเอเบิล <Disable> 8288 Bus Controller และจะใช้ดิสเอเบิลพอร์ท I/O ต่างๆ ที่ไม่เกี่ยวข้องกับขบวนการ DMA ที่เกิดขึ้นนี้ ที่จำเป็นต้องทำเช่นนี้ก็เพราะในระหว่างขบวนการ DMA นั้น 8237A-5 จะส่งแอดเดรสของหน่วยความจำออกมาบนบัสแอดเดรส และจะทำให้สัญญาณ IOR หรือ IOW แอคทีฟด้วย ดังนั้นถ้าไม่ทำการดิสเอเบิลพอร์ท I/O ที่ไม่เกี่ยวข้องไว้ ก็อาจจะทำให้พอร์ท I/O ที่มีแอดเดรสตรงกับค่าแอดเดรสบนบัสแอดเดรส (ซึ่งเป็นแอดเดรสของหน่วยความจำ)นั้น ทำการอ่านหรือส่งข้อมูลออกมาบนบัสข้อมูลทำให้เกิดความผิดพลาดขึ้นได้

บัสของแหล่งจ่ายไฟของระบบ

+5 Vdc <ขา B3 และ B29> :

ขาทั้งสองนี้ต่อกับแหล่งจ่ายไฟ DC +5V ของระบบโดยจะมีค่าความเที่ยงตรง <Regulator> + 5% คืออยู่ในช่วง +4.75 ถึง +5.25 Vdc

+12 Vdc <ขา B9> :

ขานี้จะต่อกับแหล่งจากไฟ DC +12V ของระบบโดยจะมีค่าความเที่ยงตรง <Regulator> + 5% คืออยู่ในช่วง +11.4 ถึง +12.6 Vdc

GND <ขา B1, B10 และ B31> :

ขาทั้งสามนี้จะต่อเข้ากับกราวด์ (Ground) ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรทางค่าน Hardware

การตีโค้ดแอดเดรสสำหรับพอร์ท I/O

สำหรับในหัวข้อนี้จะกล่าวถึงวิธีการตีโค้ดแอดเดรสและลักษณะการทำงานของของสัญญาณของ CPU จะอธิบายเฉพาะสัญญาณที่เรานำมาทำการตีโค้ดเท่านั้น

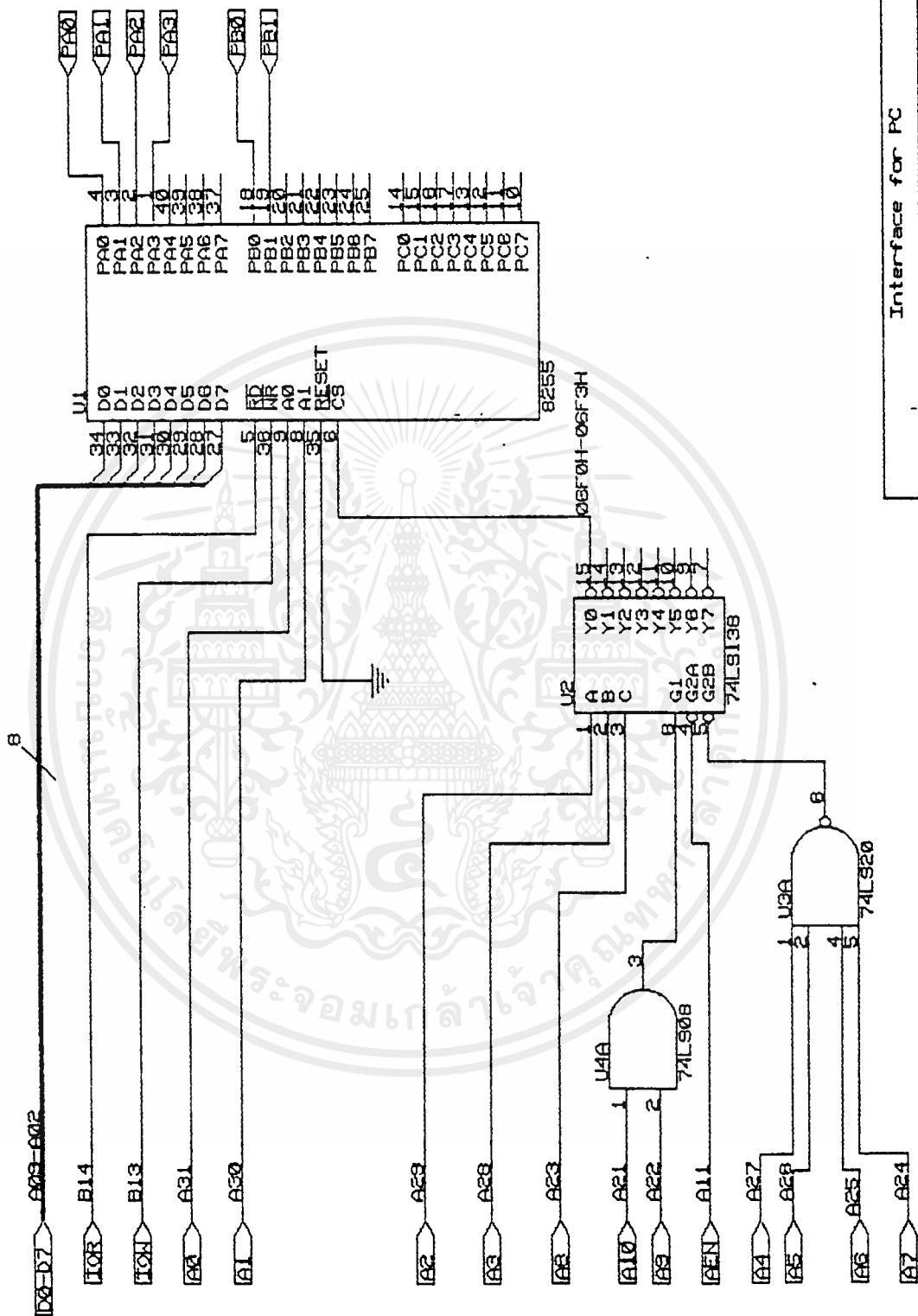
วงจรการ DECODE ADDRESS ของพอร์ท I/O

วิธีการตีโค้ดเป็นวิธีที่ง่ายและสะดวกในการตีโค้ดแอดเดรสหรือ กลุ่มของแอดเดรสของพอร์ท I/O ซึ่งวิธีนี้เป็นการกำหนดจำนวนของแอดเดรสที่เราต้องการใช้ จากนั้นจึงทำการเลือกบล็อกของแอดเดรสที่ยังไม่ถูกใช้งานโดยคาร์ดหรือวงจรอินเทอร์เฟสอื่นๆ แล้วจึงออกแบบวงจรที่ทำการตีโค้ดแอดเดรสที่เราต้องการ

สำหรับในวงจร Interface for PC นี้จะเลือกใช้การตีโค้ดแอดเดรส 02F0H จนถึง 02F3H คือ ใช้สัญญาณเอาต์พุต (สัญญาณ GROUPSELECT) จากขา YO (ขา 15) ของ 74LS138 ชุดที่เป็น WRITE REG ซึ่งจะแอกทิฟ (ลอจิก "0") เมื่อ CPU ต้องการจะส่งข้อมูลให้กับวงจรภายนอก (สัญญาณ IOW แอกทิฟ) และ ชุดที่เป็น READ REG ซึ่งจะแอกทิฟเมื่อ CPU ต้องการจะอ่านข้อมูลจากวงจรภายนอก (สัญญาณ IOR แอกทิฟ) สัญญาณ WRITE REG และ READ REG นี้โดยทั่วไปจะนำไปเป็น สัญญาณสโตรบ <Strobe> ให้กับวงจรภายนอกที่เกี่ยวข้อง เพื่อให้สามารถส่งหรือรับข้อมูลจาก CPU ได้ในช่วงเวลาที่เหมาะสม เพื่อให้ CPU สามารถส่งหรือรับข้อมูลจากอุปกรณ์ภายนอกได้เมื่อแอดเดรสในกลุ่มนี้ถูกเลือก สำหรับทิศทางของข้อมูลจะถูกควบคุมโดยสัญญาณ IOR ส่วนสัญญาณ AEN จะถูกนำมาใช้ในการดิสเอเบิลวงจรตีโค้ดโดยถ้าสัญญาณ AEN เป็น "1" ซึ่งเป็นช่วงเวลาของขบวนการ DMA นั้น 74LS138 จะถูกดิสเอเบิลทันที ทั้งนี้ก็เพื่อป้องกันความผิดพลาดที่อาจเกิดขึ้น เนื่องจากการตีโค้ดแอดเดรสของพอร์ทใน ระหว่างขบวนการ DMA นั้นเอง (ในระหว่างนี้แอดเดรสบนบัสแอดเดรสจะเป็นแอดเดรสของหน่วยความจำ คือ สัญญาณ MEMW หรือ MEMR จะแอกทิฟ แต่ในขณะที่เดียวกันสัญญาณ IOR หรือ IOW ก็แอกทิฟด้วย ดังนั้นถ้าไม่ดิสเอเบิลวงจรตีโค้ดไว้แล้ว อาจจะทำให้วงจรตีโค้ดคิดว่าแอดเดรสบนบัส แอดเดรสเป็นแอดเดรสของพอร์ท I/O ก็ได้)

ในการตีโค้ดแอดเดรสของพอร์ท I/O เราจะต้องคำนึงถึงช่วงเวลาของสัญญาณที่เกิดขึ้นในขบวนการอ่านหรือเขียนข้อมูลลงบนพอร์ท I/O ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Interface for PC

Size	Document Number	Image_1
9		

Date: September 30, 1992 Sheet 1 of 1

Date: September 30, 1992 Sheet 1 of 1

1. ในช่วงเริ่มต้นของบัสไซเคิลที่เกี่ยวกับพอร์ท I/O นั้น ถ้าสัญญาณจาก วงจรตีโค้ดมีการหน่วงเวลา <Delay> มากเกินไป อาจจะทำให้สัญญาณตีโค้ดนี้เกิดขึ้นหลังจากที่สัญญาณ IOR หรือ IOW แอคทิฟ และเนื่องจากค่าแอดเดรสบนบัสแอดเดรสนั้น วงจรตีโค้ดจะได้รับค่าแอดเดรสอื่น ๆ อยู่ ซึ่งถ้าหากวงจรตีโค้ดมีการหน่วงเวลามากเกินไปแล้ว สัญญาณตีโค้ดแอดเดรสที่ไม่ถูกต้องนี้อาจจะถูกหน่วงเวลาจนเกิดขึ้นในช่วงเวลาที่สัญญาณ IOR หรือ IOW เกิดขึ้นแล้วก็ได้ ทำให้ข้อมูลบนบัสข้อมูลนั้นถูกส่งไปยังพอร์ทที่ไม่ถูกต้อง สำหรับใน IBM/PC จะถูกออกแบบมาให้การหน่วงเวลาในวงจรตีโค้ดนั้นมีค่าไม่เกิน 92 nanosec

2. ในช่วงท้ายของบัสไซเคิลในการเขียนข้อมูลลงบนพอร์ท I/O นั้น ถ้าสัญญาณ IOW มีการหน่วงเวลาออกไป และวงจรตีโค้ดมีความเร็วในการทำงานสูงแล้ว อาจจะทำให้ข้อมูลในบัสไซเคิลนี้ถูกส่งให้กับพอร์ท I/O ที่มีแอดเดรสตรงกับค่าแอดเดรสในบัสไซเคิลต่อไปก็ได้สำหรับใน IBM/PC สัญญาณ IOW จะมีการหน่วงเวลาไปไม่เกิน 200 nanosec

อย่างไรก็ตามช่วงเวลาที่ต้องสนใจมากอีกช่วงเวลาหนึ่ง ก็คือ ช่วงเวลา ระหว่างขอบขาขึ้นของสัญญาณ IOW กับช่วงเวลาที่ข้อมูลที่ต้องถูกส่งออกมาบนบัสข้อมูล ถ้าสัญญาณ IOW ถูกหน่วงเวลาไปเกินกว่า 120 nanosec แล้ว อาจจะทำให้พอร์ท I/O ได้รับข้อมูลที่ไม่ถูกต้องก็ได้ และสำหรับสัญญาณ IOR นั้นถ้ามีการหน่วงเวลาเกิดขึ้นแล้ว ก็จะทำให้ความเร็วในการอ่านข้อมูลถูกลดลง

คุณสมบัติของแผงวงจร DigiIMAGE รุ่น DZ-II การถ่ายภาพ

1. บนแผงวงจรมีหน่วยความจำสำหรับภาพขนาด 192K ไบต์ ซึ่งไมโครคอมพิวเตอร์ สามารถติดต่อกับหน่วยความจำนี้ได้โดยตรง
 2. วงจรถ่ายภาพเป็นแบบ REAL TIME สำหรับภาพขาวดำ และเกือบ REAL TIME สำหรับภาพสี
 3. สามารถดูภาพ LIFE TIME จากจอ VGA ได้ขณะถ่าย
 4. การปรับแต่งภาพสามารถควบคุมความสว่าง (brightness), ความชัดเจน (contrast) และความเข้มของสี (saturation) ได้จากซอฟต์แวร์
 5. รับสัญญาณอินพุท แบบ COMPOSITE ขาวดำ หรือสี (ระบบ PAL) ก็ได้ เช่นสัญญาณจากกล้องถ่ายวิดีโอ
- เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความละเอียดของภาพ

1. ความละเอียดของภาพสีได้ถึง 320*200 จุด 250 สี (สีจริง)
2. ความละเอียดของภาพขาวดำได้ถึง 320*200 จุด 64 ระดับเทา
3. ภาพที่ถ่ายได้สามารถเก็บในรูปแบบ PIC ของ PC STORY BOARD

PLUS หรือ TIFF

หลักการทำงานของแผงวงจร DigiIMAGE

แผงวงจร DigiIMAGE ประกอบด้วยส่วนต่าง ๆ ที่สำคัญ ดังนี้ คือ

1. หน่วยความจำสำหรับเก็บภาพอิมเมจ (RAM) 3 Bank ละ 64K ไบต์
2. ส่วนแปลงสัญญาณวิดีโอจากอนาล็อกให้เป็นข้อมูลดิจิทัล (A/D)
3. ส่วนแยกสัญญาณสี
4. ส่วนปรับ contrast, brightness, saturation
5. ส่วนควบคุมการติดต่อระหว่างส่วนต่าง ๆ จากคอมพิวเตอร์

เมื่อถ่ายภาพสีสัญญาณสีจากกล้องวิดีโอ จะเข้ามาทางอินพุตของวงจรไปเข้าส่วนแยกสัญญาณสี ออกมาเป็นสี 3 สีคือ แดง (R), เขียว (G), น้ำเงิน (B) แล้วจึงส่งออกไปส่วนแปลงสัญญาณอนาล็อกเป็นดิจิทัล (A/D) เพื่อนำสัญญาณภาพไปเก็บในหน่วยความจำ (RAM) ในส่วนของวงจรแยกสีนี้ เราสามารถควบคุมระดับความเข้มสีจากไม่มีสีไปจนถึงมีสีเข้มมากได้เรียกว่า การปรับ SATURATION ซึ่งควบคุมได้ด้วยซอฟต์แวร์ ส่วนในวงจร A/D เราสามารถควบคุมความชัดเจนของภาพได้ด้วยการปรับ Reference+ (Ref+) และ Reference- (Ref-) ซึ่งโดยทั่วไป Ref+ จะอยู่สูงกว่าสัญญาณภาพ และ Ref- จะอยู่ต่ำกว่าสัญญาณภาพ ซึ่งเราควบคุมได้ด้วยซอฟต์แวร์

ลักษณะของการนำสัญญาณภาพ ที่ถูกแปลงเป็นดิจิทัลแล้วไปเก็บลงในหน่วยความจำนั้น จะทำการเก็บทีละสีลงในหน่วยความจำสีละ BANK จนครบ 3 BANK ตามลำดับ

เมื่อถ่ายภาพขาวดำ (ส่งจากซอฟต์แวร์) สัญญาณภาพจะนำไปเข้าส่วนวงจรแปลงอนาล็อกเป็นดิจิทัลเลย ดังนั้นการเก็บข้อมูลภาพจึงอาจเก็บไว้ในหน่วยความจำเพียง BANK เดียวเท่านั้น

ไมโครคอมพิวเตอร์สามารถติดต่อกับหน่วยความจำ (RAM) บน card ได้ โดยการส่งผ่านพอร์ตควบคุม ปกติหน่วยความจำภาพนี้จะแยกออกจากระบบของไมโครคอมพิวเตอร์ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เตอร์ ในการถ่ายภาพ ผู้ใช้สามารถเลือกติดต่อกับหน่วยความจำ BANK ใดก็ได้เพื่อทำการเก็บข้อมูลภาพที่ถ่ายนั้น ในทำนองเดียวกันผู้ใช้อีกก็สามารถเลือกติดต่อกับหน่วยความจำ BANK ใดก็ได้เพื่อทำการอ่านข้อมูลที่เก็บไว้แล้วมาใช้ แต่จะต้องสั่งให้วงจรหยุดการถ่ายภาพเสียก่อน (freeze) จึงจะทำการอ่านข้อมูลได้ และเมื่ออ่านข้อมูลจากหน่วยความจำเรียบร้อยแล้ว จึงสามารถสั่งให้หน่วยความจำนั้นกลับไปอยู่โหมดการถ่ายภาพเพื่อถ่ายภาพใหม่ได้อีก

2.3 การติดตั้งระบบฮาร์ดแวร์

แผงวงจร DigiIMAGE รุ่น DZ-11 สามารถกำหนดตำแหน่งของ พอร์ต และ หน่วยความจำได้โดยผู้ใช้ ด้วยการตั้งตำแหน่งของ jumper J1 ตำแหน่งแอดเดรสที่เลือกไว้จะมีดังนี้

	a	b	c	d	e
1	0	0	0	0	0
Jumper J1	0	0	0	0	0
0	0	0	0	0	0

แอดเดรสของพอร์ตเลือกที่ jumper a,b

แอดเดรสของหน่วยความจำเลือกที่ jumper c,d,e

I/O Address	a	b	Memory Address	c	d	e
210-21Fh	0	0	C400-C3FFh	0	0	0
2A0-2AFh	0	1	C800-CBBFh	0	0	1
310-30Fh	1	0	CC00-CFFFh	0	1	0
3A0-3AFh	1	1	D000-D3FFh	0	1	1
			D800-DBFFh	1	0	0
			DC00-DFFFh	1	0	1
			E000-E3FFh	1	1	0
			Not select	1	1	1

โดยปกติแผงวงจรรุ่น DZ-II จะถูกติดตั้งแอดเดรสของพอร์ตไว้ที่ 210H และ แอดเดรสของหน่วยความจำไว้ที่ D000H หากต้องการเลือกแอดเดรสของพอร์ตไว้ที่ ตำแหน่งอื่น ให้เลือกตำแหน่งของ jumper J1 ตรง a,b หากต้องการเลือกแอดเดรสของหน่วยความจำใหม่ ก็ให้เลือกตำแหน่งของ jumper J1 ตรง c,d,e ผู้ใช้สามารถเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการติดตั้งระบบฮาร์ดแวร์ด้วยตัวเองได้โดยการเสียบแผงวงจร DigiIMAGE ลงในช่องขยาย (Expansion Slot) ช่องใดช่องหนึ่งของเครื่องไมโครคอมพิวเตอร์สำหรับเครื่องคอมพิวเตอร์ทั่วไป ไม่ว่าจะเป็นระบบ PC,XT,AT หรือ AT 386 ที่ประกอบด้วยแผงจอภาพ (ภาษาไทย 25 บรรทัด ,Mono, HGC, CGA, EGA, VGA) แผงควบคุมดิสก์ไดรฟ์, ฮาร์ดดิสก์, และแผงวงจร Multi I/O (หรือ Multi Function) จะสามารถติดตั้งแผงวงจร DigiIMAGE ลงไปได้ทันที

หากเครื่องคอมพิวเตอร์มีแผงวงจรอื่นนอกจากที่กล่าวมาแล้วติดตั้งอยู่ เช่น แผงวงจร LAN, แผงวงจรสแกนเนอร์, แผงวงจร EMS และวงจรพิเศษอื่น ๆ รวมทั้ง วงจรพิเศษที่มีการติดตั้งอยู่บนเมนบอร์ด เช่น วงจร EMS (สำหรับเครื่อง AT บางรุ่น) หากแอดเดรสของพอร์ตหรือหน่วยความจำของวงจรพิเศษเหล่านี้ ถูกเลือกไว้ตรงกับของแผงวงจร DigiIMAGE ก็จะต้องมีการย้ายตำแหน่งแอดเดรสของแผงวงจรอันใดอันหนึ่งใหม่ เพื่อให้ตำแหน่ง Address ของแผงวงจร DigiIMAGE ไปชนกับ address ของแผงวงจรอื่น

ส่วนตำแหน่ง jumper ที่ J2 และ J3 ใช้สำหรับการเลือกสัญญาณจาก input เพื่อให้เหมาะสมกับแผงวงจร DigiIMAGE ความหมายและตำแหน่งที่เลือกมีดังนี้

	1	2	3
J2	0	0	0
J3	0	0	0

J2 เชื่อมระหว่าง

1,2 = เลือกใช้กับกล่องที่ทำให้สัญญาณออกมาเป็น ขาว/ดำ

2,3 = เลือกใช้กับกล่องที่ทำให้สัญญาณออกมาเป็น สี

J3 เชื่อมระหว่าง

1,2 = สำหรับสัญญาณวิดีโอที่ต้องการ INPUT IMPEDANCE สูง

2,3 = สำหรับสัญญาณวิดีโอที่ต้องการ INPUT IMPEDANCE 75 โอห์ม

โดยปกติตำแหน่งของ jumper J2 จะตั้งไว้ที่ 2,3 และ J3 จะตั้งไว้ที่ 1,2

เมื่อทำการติดตั้งแผงวงจร DigiIMAGE ลงไปในเครื่องเรียบร้อยแล้ว นำสัญญาณ VIDEO ป้อนเข้าที่แจ็ค VDO-IN จากนั้นให้เรียกโปรแกรม DZ2DI.COM เลือก menu ไปที่ Digitize now เพื่อนำสัญญาณภาพ video ไปแสดงบนจอ VGA ตามขั้นตอนต่อไป เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดต่อกับ card DZ-II

บนแผงวงจร DogoO<AGE DZ-II มีพอร์ต Output Port อยู่ 4 Port, Input port อยู่ 2 Port ซึ่งมีหน้าที่ต่าง ๆ ดังนี้
Output Port.4 Port มีดังนี้

CtrlPort : เป็นพอร์ตคำสั่งในการถ่ายภาพ รวมทั้งการเลือก BANK, PAGE ของหน่วยความจำ และสิ่งที่ต้องการถ่าย ตำแหน่งของพอร์ตนี้ อยู่ที่ 210h, 2a0h, 310h, 3a0h ตามการตั้งตำแหน่งของ jumper ความหมายของ พอร์ตนี้ มีดังนี้

d7 เป็นคำสั่งในการถ่ายหรือหยุดภาพ

0 = สั่งให้หยุด digitize

1 = สั่งให้ digitize

d6 สัญญาณนี้ต่อออกไปที่ช่อง J-out สามารถนำไปควบคุมอะไรก็ได้ ระดับของสัญญาณจะเหมือนกับสัญญาณที่ออกมาจาก IC TTL

d5,d4 เลือกให้สัญญาณที่ต้องการถ่าย ไปเข้าวงจร A/D

00 = เลือกถ่ายสี Y (สัญญาณขาวดำ)

01 = เลือกถ่ายสี R (แดง)

10 = เลือกถ่ายสี G (สีเขียว)

11 = เลือกถ่ายสี B (น้ำเงิน)

d3,d2 เลือกหน่วยความจำ BANK ที่ต้องการเก็บภาพที่ถูกแปลงเป็นดิจิทัล หรือเลือกหน่วยความจำ BANK ที่ต้องการอ่านหรือเขียนข้อมูล ซึ่งมี 3 BANK

00 = เลือก RAM BANK 0

01 = เลือก RAM BANK 1

10 = เลือก RAM BANK 2

d1,d0 เลือก PAGE ของหน่วยความจำที่ต้องการอ่านหรือเขียน โดยในหน่วยความจำ

แต่ละ BANK จะติดต่อกับ CPU เป็น PAGE ซึ่งมี BANK ละ 4 PAGE

แต่ละ PAGE มีหน่วยความจำอยู่ 16 K byte

00 = เลือก RAM BANK 0

01 = เลือก RAM BANK 1

10 = เลือก RAM BANK 2

11 = เลือก RAM BANK 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- RefhPort** : มีตำแหน่งอยู่ที่ CtrlPort+1 เป็นค่าของ referent+ สำหรับควบคุมการทำงานของวงจรแปลงอนาล็อกเป็นดิจิตอล
d5-d0 = ค่าตั้งแต่ 0-63
- Ref1Port** : มีตำแหน่งอยู่ที่ CtrlPort+2 เป็นค่าของ referent- สำหรับควบคุมการทำงานของวงจรแปลงอนาล็อกเป็นดิจิตอล
d5-d0 = ค่าตั้งแต่ 0-63
- SatPort** : มีตำแหน่งอยู่ที่ CtrlPort+3 เป็นค่าของ referent+ สำหรับควบคุมการทำงานของวงจรแยกสี
d5-d0 = ค่าตั้งแต่ 0-63

Input Port 2 Port มีดังนี้

- StatPort** : มีตำแหน่งอยู่ที่ CtrlPort เป็นพอร์ทอ่านสัญญาณที่ใช้ในการตรวจสอบเมื่อจะถ่ายภาพ รวมทั้งอ่านข้อมูลของ CtrlPort บางอย่างกลับมาด้วย
d7 อ่านมาจาก d7 ของ CtrlPort
d6 ตรวจสอบสัญญาณจากช่อง J-IN เพื่อนำไปใช้ในโปรแกรมต่างๆ ผู้ใช้สามารถต่อไปใช้งานได้ สัญญาณที่ต้องการจะต้องมีระดับแรงดันเท่ากับที่อินพุทของ IC TTL ต้องการ ปกติจะถูกดึงขึ้นไฟบวก 5 Volt ด้วย ค่าความต้านทาน 4.7K Ohm
d5 ตรวจสอบสัญญาณการถ่ายภาพ digi_stat (เป็น PULSE)
d4 ตรวจสอบสัญญาณ Hor. sync (เป็น PULSE)
d3-d0 อ่านมาจาก d3-d0 ของ CtrlPort
- SWPort** : มีตำแหน่งของพอร์ทอยู่ที่ CtrlPort+1 เป็นพอร์ทที่ใช้อ่านตำแหน่งของ Jumper J1 ตัว c,d,e ที่ได้ตั้งไว้ พอร์ทนี้มีคุณสมบัติพิเศษคือ จะต้องอ่านอย่างน้อย 2 ครั้ง โดยครั้งแรกจะได้ข้อมูลที่ผิดแต่ครั้งต่อไปจะถูก
d3,d2,d1 อ่านจากตำแหน่งของ Jumper J1 ตรง c,d,e ตามลำดับ
d0 มีค่าเป็น 0 (สำหรับตรวจสอบว่าเป็นแผงวงจรสี)

ตัวอย่าง การติดต่อกับ DZ-II

สั่งให้ DZ-II ถ่ายภาพ (digitize)

out CtrlPort+2,(63-brightness) ;Ref-

out CtrlPort+1,(reference-) + ((63-reference-)*(63-contrast))/63 ;Ref+

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือกสัญญาณที่ต้องการถ่ายไปไว้ในวงจรมัลติเพล็กซ์เป็นดิจิทัล และ เลือกหน่วยความจำ BANK ที่ต้องการ

3. ตรวจสอบสัญญาณ `digi_stat` (ที่บิต d5 ของ `SatPort`) จะต้องเป็น PULSE (1) ภายในไม่เกิน 20ms แล้วรอสัญญาณนี้จนตกเป็น 0 หมายถึงถ่ายภาพครบ 1 frame
4. หากต้องการสั่งให้หยุด `digitize` (freeze) ให้ส่งคำสั่งหยุดภาพออก `CtrlPort` (บิต d7 เป็น 0)
5. สำหรับการถ่ายภาพในโหมด `color RGB` จะต้อง สั่ง `digitize` ภาพสี ลง หน่วยความจำ BANK 0,1,2 ตามลำดับ ดังนั้นจึงต้องสั่ง `digitize` ทั้งหมด 3 ครั้ง และในแต่ละครั้งจะต้องรอสัญญาณ `digi_stat` ตกจาก PULSE 1 เป็น 0 เสมอ
6. เมื่อต้องการอ่านข้อมูลที่อยู่บนหน่วยความจำของการ์ด DZ-II มาลงยังหน่วยความจำของคอมพิวเตอร์ เราสามารถอ่านได้ครั้งละ 16K Byte หรือเท่ากับ 1 PAGE ฉะนั้นการอ่านภาพสี 65 K Byte 3 BANK จะต้องอ่านถึง 12 เทียว ซึ่งในการอ่านแต่ละเทียวจะต้องส่งคำสั่งออก `CtrlPort` เพื่อเลือก BANK และ PAGE ที่ต้องการอ่าน

ขั้นตอนต่างๆ สามารถเขียนเป็นกระบวนการความได้ดังนี้

1. การตรวจสอบสัญญาณของการ์ดและสแกนหา Address ของ `CtrlPort`

เป็นการตรวจหาพอร์ททั้ง 4 ตำแหน่ง คือ 210h, 2a0h, 310h และ 3a0h โดยการตรวจดูสัญญาณที่บ่งบอกว่าเป็นการ์ด DZ-II สัญญาณที่ตรวจจะอยู่ที่พอร์ททางออกไป 8001h เช่น 21h จะตรวจที่ 8211h เป็นต้น

PROCEDURE `Chk_ID_Card`

`PortBase [ARRAY 1 to 4] = 210h, 2a0h, 310h, 3a0h`

`Index = 1`

FOR `Count = 1 to 4` do

`CtrlPort = PortBase[Index]`

`SWPort = CtrlPort + 1`

`ChkPort = CtrlPort OR 8001h`

IN PORT `SWport` TO `Chk1` (Skip first byte)

IN PORT `SWPort` TO `Chk1`

`Chk1 = ((Chk1 XOR 0fh) AND 0fh)`

`Chk2 = Chk1`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

FOR Time = 0 TO 200 DO
    IN PORT ChkPort to Chk1
    Chk1 = ((Chk1 XOR 0fh) AND 0fh)
    IF ( Chk <> Chk2)
        GOTO NextBase
    END IF
    Chk2 = ((Chk2 Xor 0fh) AND 0fh)
END FOR
EXIT (with CtrlPort)
NextBase:
    Index = Index + 1
END FOR
PRINT "Card DZ-II Not Found II"
END PROCEDURE

```

2. การตรวจสอบสัญญาณ Digi_Stat (ที่บิต d5 ของCtrlPort)

จะต้องตรวจสอบหลังจากส่งคำสั่งถ่ายภาพออกไปแล้ว ซึ่งเมื่อออกจาก PROCEDURE นี้แล้ว ถึงถ่ายภาพได้ครบ 1 เฟรม แต่ถ้าสัญญาณ Digi_Stat ไม่เคยเป็น 1 เลย ภายในช่วงเวลาที่กำหนดแสดงว่าไม่มีสัญญาณ Video ส่งมาที่การ์ด

```

PROCEDURE Chk_Digi_Stat
    FOR Pulse = 1 DOWN TO 0
        Time = 0
        Time_Out = 4000
        WHILE digi_stat <> Pulse
            Time = Time + 1
            IF(Time > Time_Out)
                PRINT "VIDEO Signal Not Found."
                EXIT
            END IF
        END WHILE
    END FOR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

END WHILE

END FOR

END PROCEDURE

3. การสั่งถ่ายภาพ

เราสามารถเลือกถ่ายได้ทั้ง โหมดสีและขาว-ดำ สำหรับ PROCEDURE Digitize_RGB จะเป็นการสั่งถ่ายภาพสี R,G,B ขนาด 256*256 จุดลงหน่วยความจำ BANK 0,1,2 ตามลำดับ และ PROCEDURE Digitize_B&W เป็นการสั่งถ่ายภาพ ขาว-ดำ ขนาด 256*256 จุดลงหน่วยความจำ BANK0 เพียง BANK เดียว จะเห็นว่าขั้นตอนในการสั่งถ่ายแต่ละครั้งจะต้องทำการตรวจสอบสัญญาณ Digi_Stat เสมอ เพื่อเป็นการบอกให้รู้ว่าภาพที่กำลังถ่ายอยู่นั้นถ่ายครบเฟรมหรือยัง เมื่อกระบวนการถ่ายเสร็จสิ้นไปแล้ว เราจึงสั่งให้หยุดถ่าย เป็นการถ่ายครบ 1 ภาพ และเมื่อต้องการ ถ่ายภาพต่อไปก็ให้ทำตามขั้นตอนของกระบวนการความนี้

การสั่งถ่ายภาพสี

PROCEDURE

DigiCom_R = 90h (10010000 b)

DigiCom_G = A4h (10100100 b)

DigiCom_B = B8h (10111000 b)

DigiComStop = 0 (00000000 b)

OUT CtrlPort,DigiCom_R

CALL Chk_Digi_Stat

OUT CtrlPort,DigiCom_G

CALL Chk_Digi_Stat

OUT CtrlPort,DigiCom_B

CALL Chk_Digi_Stat

OUT CtrlPort,DigiComStop

END PROCEDURE

การสั่งถ่ายภาพขาวดำ

PROCEDURE Digitize_B&W

DigiCom_Y = 80h (1000000 b)

DigiComStop = 0 (0000000 b)

OUT CtrlPort,DigiCom_Y

CALL Chk_Digi_Stat

OUT CtrlPort,DigiComStop

END PROCEDURE

4.การปรับค่าของ Contrast,Brightness และ Saturation

สามารถเปลี่ยนค่านี้ได้ตั้งแต่ 0-63 ระดับ และ ค่าที่ได้จากการเปลี่ยนแปลง
เราจะนำมาโปรแกรม ตามขั้นตอนของ กระบวนการดังนี้

PROCEDURE Con_Bri_Sat_Level

Contrast = 32 (Vary from 0 to 63)

Brightness = 50 (Vary from 0 to 63)

Saturation = 32 (Vary from 0 to 63)

Reference_ = 63 - Brihtness

RefhPort = CtrlPort + 1

Ref1Port = CtrlPort + 2

SatPort = CtrlPort + 3

OUT Ref1Port,Reference_

OUT RefhPort,Reference_+[[63-Reference-]*[63-Contrast]]/63

OUT SatPort,Saturation

END PROCEDURE

5.การอ่านตำแหน่ง Address ของหน่วยความจำบน Card DZ-II

ทำได้โดยการอ่านค่าของ J1 จาก SWPort แล้วจึงเปิดตารางหาตำแหน่ง

ของ Address

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROCEDURE Read_Address

SWPort = CtrlPort + 1

IN SWPort TO ChkSW (LATCH DATA)

IN SWPort TO ChkSW (GET DATA)

ChkSW = ChkSW AND 0eh (00001110 b)

CASE ChkSW DO

0 : CardSeg = C400h

2 : CardSeg = C800h

4 : CardSeg = CC00h

6 : CardSeg = D000h

8 : CardSeg = D800h

10 : CardSeg = DC00h

12 : CardSeg = E000h

14 : CardSeg = NULL

END CASE

END PROCEDURE

6.การติดต่อกับหน่วยความจำของ DZ-II

เราจะแยกเป็น PROCEDURE Read_RGB สำหรับอ่านข้อมูลจากหน่วยความจำ BANK 0,1,2, ของ Card DZ-II ซึ่งเป็นหน่วยความจำที่เก็บข้อมูล ของภาพสี R,G,B ตามลำดับ ลงหน่วยความจำของระบบ ที่จองไว้ 64 K * 3 และ PROCEDURE Read_Y จะเป็นการอ่านข้อมูลจากหน่วยความจำ BANK 0 ที่เก็บข้อมูลของ ภาพขาว-ดำไว้เพียง BANK เดียว เพื่อนำลงหน่วยความจำของระบบที่จองไว้ 64 K byte ส่วน PROCEDURE Read_RAM_Bank จะเป็นโปรแกรมย่อย เพื่อให้เรียกใช้โดยจะต้องส่ง พารามิเตอร์ RamBank และ BufSeg ไปให้

การติดต่อกับหน่วยความจำสำหรับภาพสี

PROCEDURE Read_RGB

Bank0_R =10h (00010000 b)

Bank0_G =24h (00100100 b)

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้ในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Bank0_B =38h (00111000 b)

BufSeg_R = 64 Kbyte  ALLOCATE  MEMORY
BufSeg_G = 64 Kbyte  ALLOCATE  MEMORY
BufSeg_B = 64 Kbyte  ALLOCATE  MEMORY

CALL Read_RAM_Bank WITH Bank0_R,BufSeg_R
CALL Read_RAM_Bank WITH Bank0_G,BufSeg_G
CALL Read_RAM_Bank WITH Bank0_B,BufSeg_B

END PROCEDURE

```

การติดต่อกับหน่วยความจำสำหรับภาพขาวดำ

```

PROCEDURE Read_Y
    Bank0_Y = 0 (00000000 b)
    BufSeg_Y = 64 Kbyte ALLOCATE MEMORY
    CALL Read_RAM_Bank WITH Bank0_Y,BufSeg_y
END PROCEDURE

```

โปรแกรมย่อยสำหรับอ่านหน่วยความจำจากคาร์ดลงหน่วยความจำของระบบ

```

PROCEDURE Read_RAM_Bank <RamBank,BufSeg>
(cardSeg from PROCEDURE Read_Address)
CardOffs = 0
BufOffs = 0

For Count = 1 TO 4
    OUT CtrlPort,RamBank
    MOVE DATA FROM CardSeg:CardOffs TO BufSeg:BufOffs
                                LENGTH 16Kbyte

    RamBank = RamBank + 1 (to next page)
    BufOffs = BufOffs + 16K

END FOR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

END PROCEDURE

ลักษณะของข้อมูลภาพ

ภาพขาวดำที่เก็บอยู่ในหน่วยความจำนั้นจะมี ขนาด 256×256 จุด โดยใช้จำนวนบิตทั้ง 6 บิตบน คือ ตั้งแต่บิต 2 ถึง บิต 7 แบ่งระดับเทาเป็น 0 - 63 ระดับ และแต่ละจุดจะต้องใช้เนื้อที่ขนาด 1 ไบต์เรียงกันไปในหน่วยความจำขนาด 64K ไบต์ ของ BANK 0 โดยข้อมูลภาพจุดแรกจะอยู่ในหน่วยความจำแอดเดรสแรกคือ 0 ข้อมูลภาพจุดถัดมา (ตามแนวนอน) ก็จะอยู่ที่แอดเดรส 1 เรียงกันไปจนจบข้อมูลภาพจะมีแอดเดรสเป็น 65535 หรือ OFFFHH

สำหรับภาพสี จะใช้หน่วยความจำ 3 BANK แบ่งเป็นสี R,G,B สีละ BANK แต่ละ BANK สามารถจุได้ทั้งหมด 256×256 จุด แต่ละจุดจะมีข้อมูลภาพอยู่ 6 บิตบน คือ ตั้งแต่บิต 2 ถึง บิต 7 ดังนั้นบิต 0 กับ บิต 1 จะเป็นข้อมูลที่ไม่ได้ใช้ และ การเก็บข้อมูลในแต่ละ BANK จะเหมือนกับการเก็บข้อมูลของภาพขาวดำ

โดยปกติข้อมูลบิต 0 และ 1 ของทุกๆ ไบต์ที่เก็บอยู่ในหน่วยความจำจะเป็นตัวบอกว่า ข้อมูลนี้เป็นสีอะไร คือ

D1	D0	
0	0	= ข้อมูลจากสัญญาณ Y
0	1	= ข้อมูลจากสัญญาณ R
1	0	= ข้อมูลจากสัญญาณ G
1	1	= ข้อมูลจากสัญญาณ B

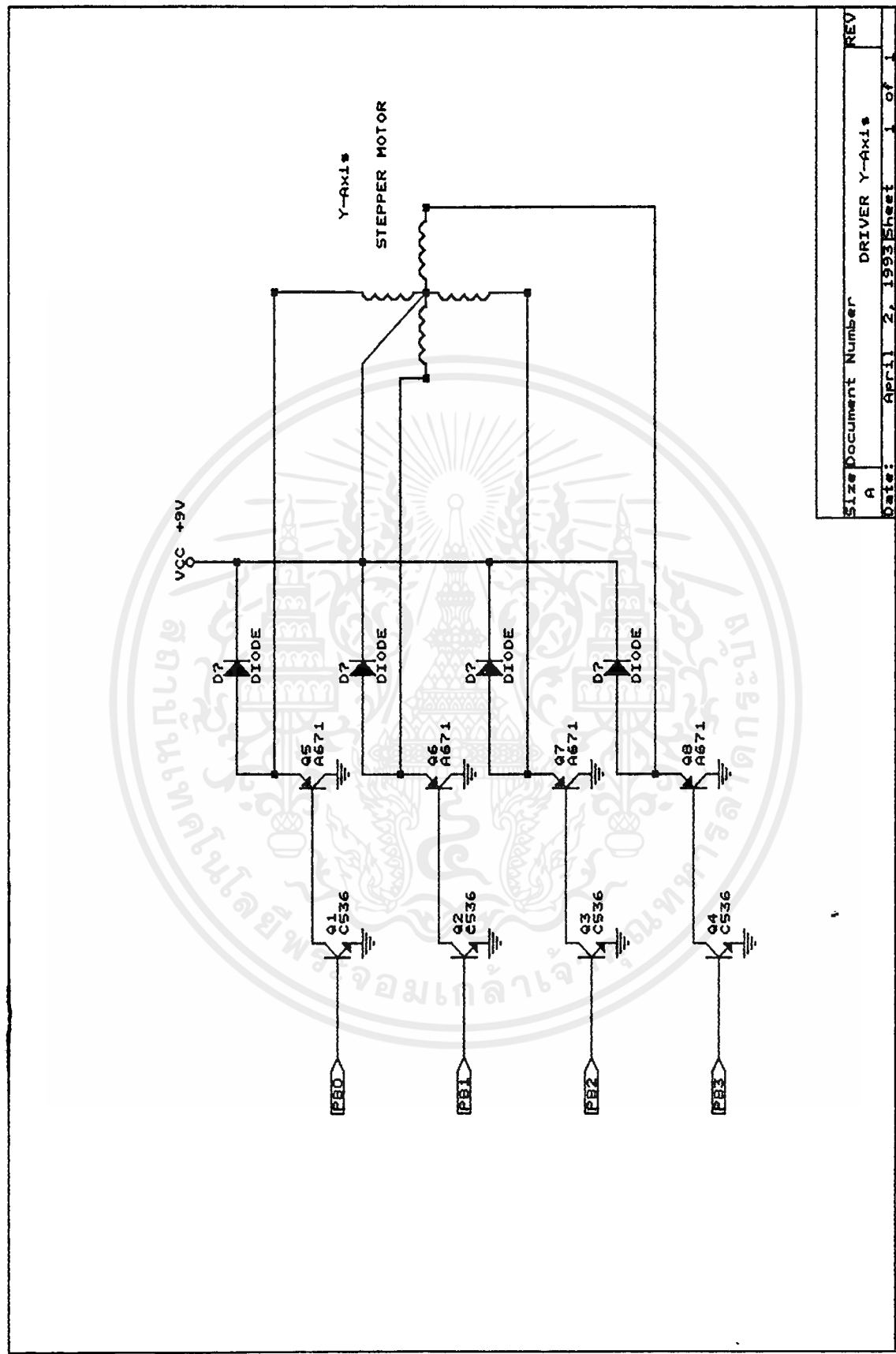
วงจร Driver

ชุด Driver ประกอบด้วย Q1 - Q8 สัญญาณ Drive Phase ของ Stepping Motor จะมาจาก IC#8255 โดย Q1 - Q4 จะเป็น Pre-driver ให้กับ Q5-Q8 อีกที ทั้งนี้เพื่อเพิ่มกระแส Drive

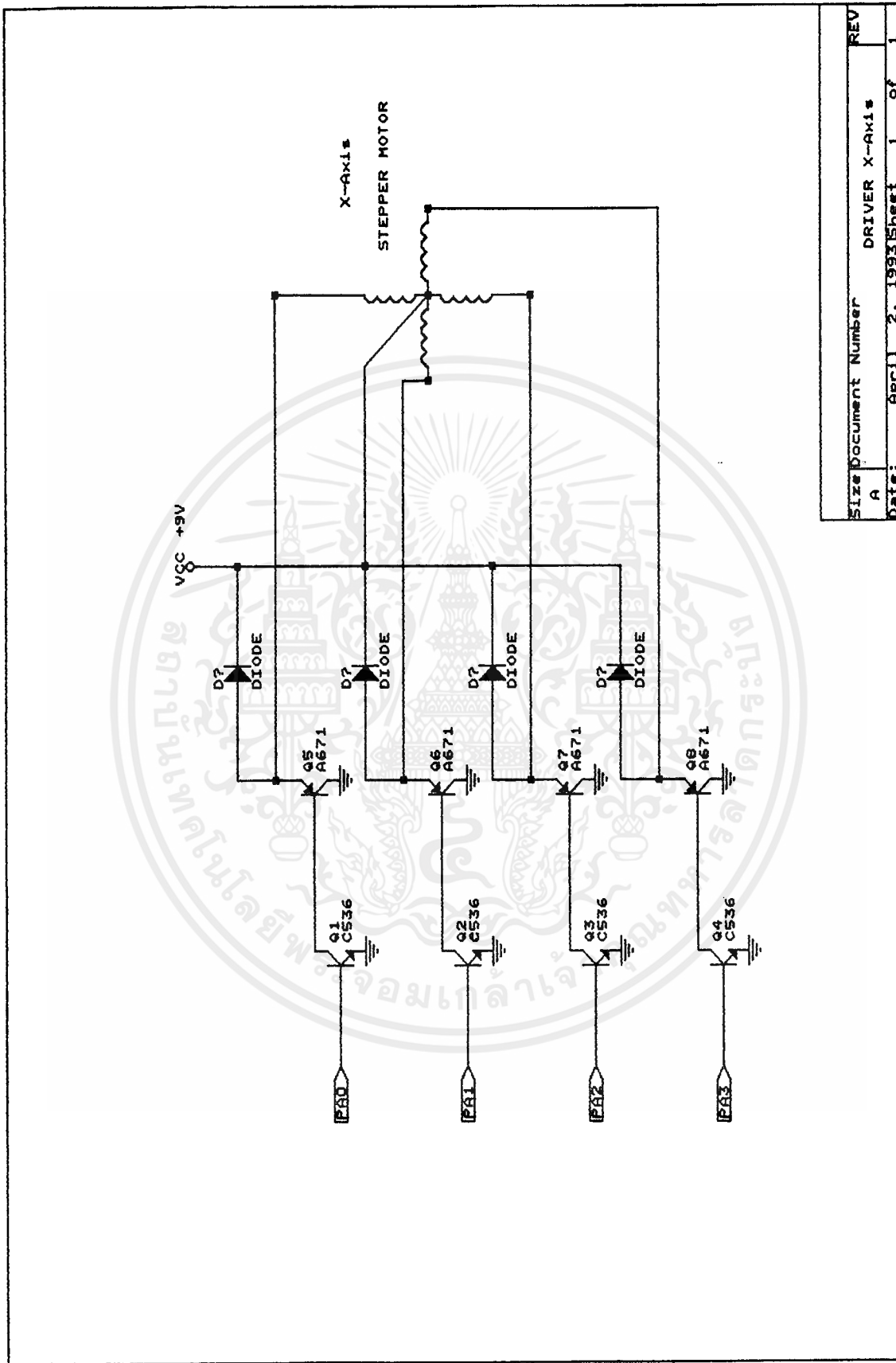
ส่วนไดโอด D_1-D_4 ที่ต่อक्रमขดลวดของสเต็ปมอเตอรืนั้น ทำหน้าที่ป้องกันแรงดันย้อนกลับ ซึ่งเกิดขึ้นจากป้องกันแรงดันย้อนกลับ ซึ่งเกิดขึ้นจากขดลวดของมอเตอร์ ขณะที่ทรานซิสเตอร์หยุดทำงาน เพื่อป้องกันไม่ให้ทรานซิสเตอร์เสียหาย สำหรับชุดจ่ายไฟวงจรในส่วนเลี้ยงมอเตอร์สามารถเปลี่ยนได้ตามขนาดของมอเตอร์ว่าใช้ไฟเลี้ยงกี่โวลต์

Power supply

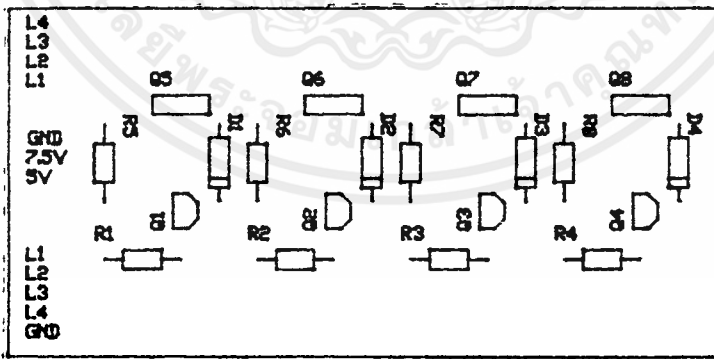
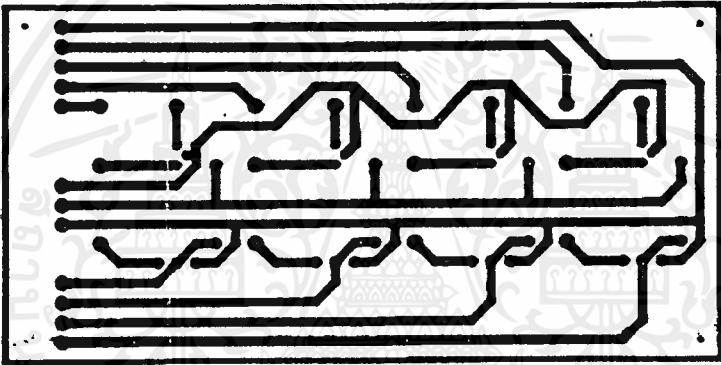
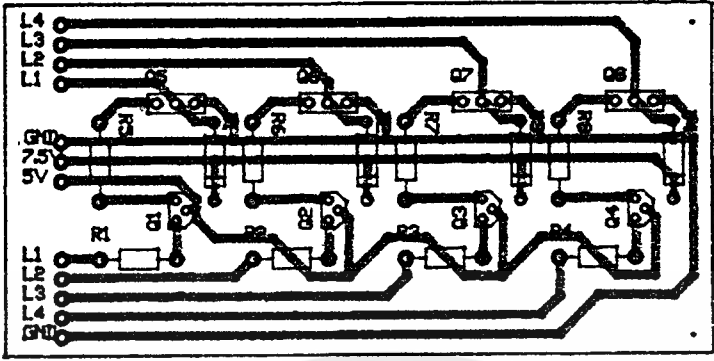
วงจร Power Supply ที่เป็นตัวกำเนิดแหล่งจ่ายไฟ DC +7.5 Volt มาเลี้ยงในระบบนี้โดยมีหลักการทำงานของแหล่งจ่าย Power Supply คือ มีชุด Transformer เป็นหม้อแปลงไฟ AC 220 VAc to 24 VAc โดยผ่านวงจร Full Wave Rectifier โดยใช้ diode #IN4004 และผ่าน capacitor Filter 4700 μ F /50V เป็นไฟ DC ที่ค่อนข้างจะราบเรียบและส่งผ่านไปยัง IC Regulator #LM 350K ให้เหลือแรงดัน DC เพียง +7.5 VDC เท่านั้น เพื่อส่งไปเลี้ยงให้แก่วงจรภายในระบบ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

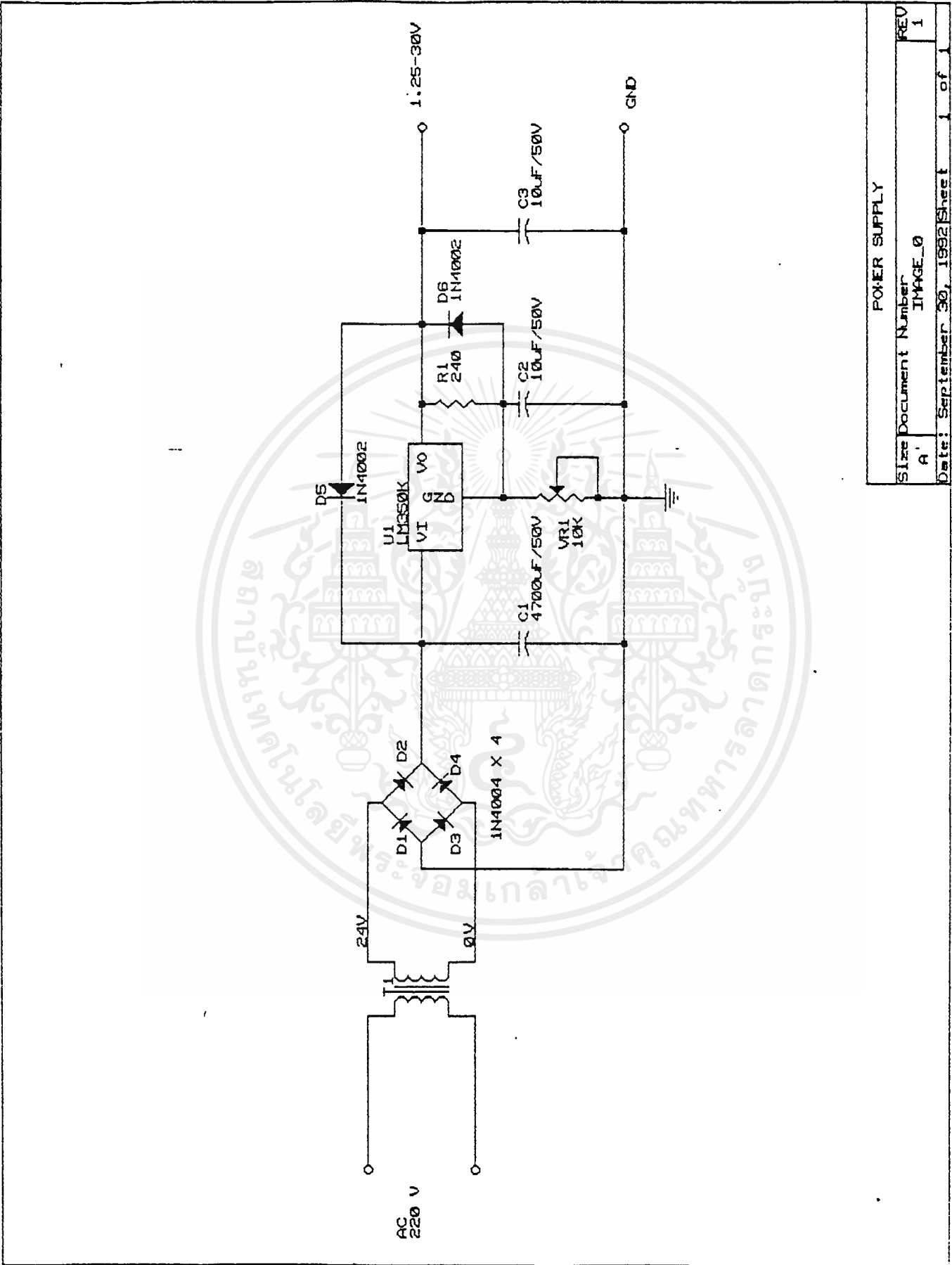


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

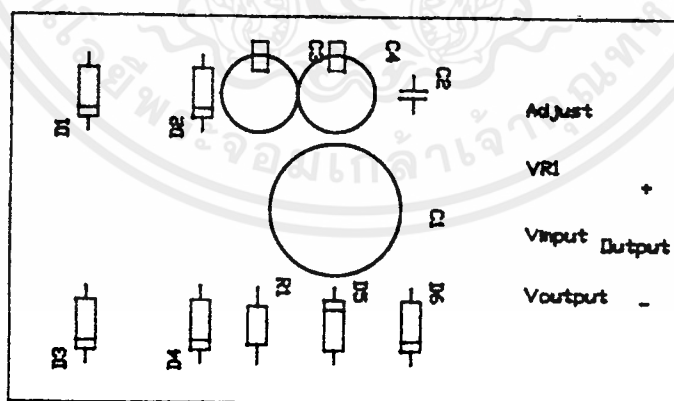
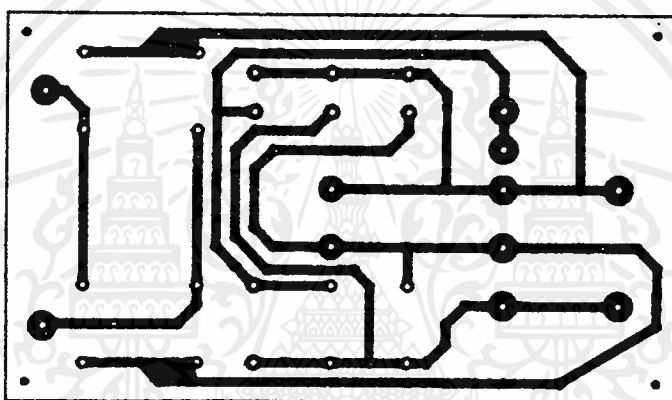
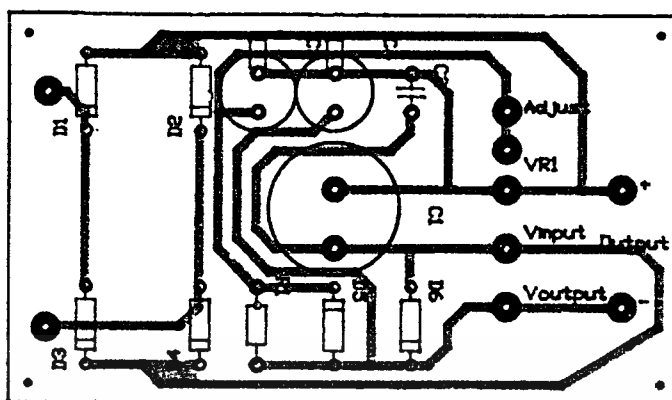


PCB DRIVER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



PCB POWER SUPPLY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8

สรุปผลงานวิจัย

ปัญหาทาง Soft ware

ในการจัดเก็บภาพนั้น จะต้องทำให้ถูกวิธี และเหมาะสม ในเบื้องต้นได้ทดลองเปรียบเทียบข้อมูลโดยใช้ขอบภาพ แต่ลักษณะการทำงานที่ได้จะช้ามาก ไม่สามารถที่จะติดตามวัตถุที่เคลื่อนที่เร็วมากๆ ได้ ดังนั้นจึงได้ใช้การเปรียบเทียบสีแค่ 2 ระดับ คือกำหนดสีของวัตถุให้เป็นสีระดับหนึ่ง และ สีพื้นอีกระดับหนึ่ง และทำการการตรวจสอบ ผลที่ได้ คือ ค่อนข้างจะถูกต้องและเร็วขึ้น แต่ก็ยังมีปัญหาอยู่คือ ถ้าแสงภายในห้องไม่พอ จะมีระดับสีของวัตถุและสีพื้นมีค่าใกล้เคียงกันทำให้โปรแกรมทำงานผิดพลาด มีลักษณะเป็นสัญญาณรบกวน การแก้ไขคือต้องเขียนโปรแกรมเพิ่มเติม เพื่อแก้ไขสัญญาณรบกวนนั้น

ปัญหาทาง Hard ware

ในการออกแบบครั้งแรก เราใช้ DC Motor ซึ่งเจอปัญหามากมาย เช่น เราไม่สามารถควบคุมตำแหน่งได้ และ ไม่สามารถรับน้ำหนักของกล่อง VDO ได้ จึงเปลี่ยนมาใช้ Stepping Motor ซึ่งสามารถควบคุมตำแหน่งได้ แต่ไม่สามารถรับน้ำหนักของกล่อง VDO ได้ จึงออกแบบโดยใช้การทดเฟือง จึงสามารถรับน้ำหนักของกล่อง VDO ได้ ซึ่งเราใช้ทดรอบ 1:32

กิตติกรรมประกาศ
(ACKNOWLEDGMENT)

ความสำเร็จของโครงการงานชิ้นนี้ ได้ประสบความสำเร็จล่วง ไปได้ด้วยดีแต่ใน กระบวนการค้นคว้าย่อมมีอุปสรรคนั้น เป็นเรื่องธรรมดากับการทำงานทุกอย่าง แต่คติพจน์ ที่ว่า "ความพยายามอยู่ที่ไหน ความสำเร็จย่อมมาถึง" ยังใช้ได้อยู่กำลังใจเป็นสิ่ง สำคัญในการทำงาน มีพลังอยู่ในตัวการ ทำงานโครงการงานชิ้นนี้จะต้องอาศัยระยะเวลา มั่นสมอง และความรู้ที่ได้สะสมมาจาก ครู, อาจารย์ ที่ได้ประสิทธิ์ประสาทวิชาความรู้ให้ พร้อมทั้งประสบการณ์จากการทำงาน ด้วยความสำเร็จของโครงการงานชิ้นนี้ มิใช่จะมีแต่ผู้ กระทำโครงการงานอย่างเดียว แต่มีสิ่งรอบข้างและบุคคลรอบข้างให้กำลังใจกับการทำงาน ในส่วนลึกของจิตใต้สำนึกจะลืมบุคคลที่ให้กำลังใจมิได้เลย แต่จะกล่าวขอบคุณอยู่ตลอด เวลา ด้วยความรู้สึกที่อยู่ในใจจะขอกล่าวประกาศกิตติกรรม ลงเป็นตัวอักษรในกิตติกรรม ประกาศนี้ การก่อกำเนิดของมนุษย์ที่มีสติปัญญา และสมองเติบโตมา และให้การศึกษามา จนได้มีโอกาสได้ทำโครงการงานนี้ ขอกล่าวขอบคุณ บิดา และมารดา ที่ให้โอกาสกับลูก ขอขอบคุณ ภาควิชาวิศวกรรม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ให้โอกาสกับ ข้าพเจ้าให้สถานที่ศึกษาเล่าเรียน ขอขอบคุณอาจารย์ ที่ปรึกษา อาจารย์ทรงชัย วีระวิมาศ ที่ให้โอกาสกับ ข้าพเจ้าได้เลือกโครงการงานนี้ และ ให้คำปรึกษากับปัญหาอย่าใกล้ชิด ตลอดระยะเวลาการทำงาน ขอขอบคุณ คุณสมศักดิ์ ดวงรัตน์ ซึ่งเป็นผู้จัดการโรงงาน ฝ่ายอิเล็กทรอนิกส์ และ คุณอัศกริน มุทธาประนัง ซึ่งเป็นหัวหน้าแผนก RD ที่ บริษัทชนิโยนิเวอร์แซลจำกัด และขอขอบคุณ อาจารย์ภาค วิชา เทคโนโลยีวิศวกรรมทั้งภาค ที่ให้ความรู้กับข้าพเจ้า ถึงจะมีใช้โดยตรงกับโครงการงาน แต่ ก็มีเกล็ดเล็กๆ น้อยๆ ที่สามารถนำมาใช้ประโยชน์ในโครงการงานนี้ ได้ ขอขอบคุณ คุณนุสร , คุณภัทรินทร์ และเพื่อน ๆ ห้อง 2Q ที่ให้กำลังใจ ถามข่าวคราวความคืบหน้าของโครงการงาน ซึ่งแสดงว่าโครงการงานนี้มีผู้ที่สนใจอยู่ ขอกล่าวขอบคุณมา ณ ที่นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. ยืน ภู่วรวรรณ, ดร.ชัยยงค์ วงศ์ชัยสุวัฒน์ "เทคโนโลยีไมโครคอมพิวเตอร์ 16 บิต"
สำนักพิมพ์เอ็ดยุคเคชั่น, พ.ศ. 2531
2. ธาณินทร์ ถาวรศาสนวงศ์ม ทินกร ตึก "การอินเตอร์เฟส IBM PC" สำนักพิมพ์
PHYSICS CENTER
3. หนังสือวารสารเคมีคอนดักเตอร์ ฉบับที่ 104, 111, 112, 113, 114 พ.ศ.2535
4. Louis J.Galbiati,Jr., "Machine vision and digital Image
Processing fundamentals",Prentice-hall,1990
5. Rafael C. Gonzalez,"Digital Image Processing", Second Edition,
Addison-Wesley,1987
6. Michael C.Fairhurst, "Computer Vision for Robotic System an
introduction", Prentice-Hall, 1988



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include <conio.h>
#include <fcntl.h>
#include <stdio.h>
#include <stdlib.h>
#include <alloc.h>
#include <dos.h>
#include <mem.h>
#include <string.h>
#include <math.h>
#include <graphics.h>

#define PutPixel(X,Y,C) pokeb(0xA000,Y*320+X,C)
#define mem_add 0xcc00
#define mem_vrm 0xa000
#define nrm_cont 32
#define nrm_brig 50
#define Error 10
#define PortA 0x6F0
#define PortB 0x6F1
#define PortCtrl 0x6F3
#define Times 10
#define Step1 1
#define Step2 5
#define Step3 50
#define Limit 1480
#define old_x 0
#define old_y 0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

unsigned      ctrlport,cardseg;

unsigned char far      *prt;

int           AbsX,AbsY;

unsigned char far      *original;

int           Teshold,his_value[64];

int Row_X=0,Col_X= -1,Abs_X=0;

int Row_Y=0,Col_Y= -1,Abs_Y=0;

int           Right_X=0,Right_Y=0;

int           Left_X=1,Left_Y=1;

char far      *FONT;

char           OldMode;

unsigned char      **VGAMAP;

/* Haft Step */

/*int Table[8][4]={ {1,0,0,0},{1,1,0,0},{0,1,0,0},{0,1,1,0},
    {0,0,1,0},{0,0,1,1},{0,0,0,1},{1,0,0,1} };*/

/* Full Step */

int Table[8][4]={ {1,0,0,1},{1,1,0,0},{0,1,1,0},{0,0,1,1},
    {1,0,0,1},{1,1,0,0},{0,1,1,0},{0,0,1,1} };

void           SetMode(unsigned char);

void           PutChar(int,int,int,int,char);

void           OutText(int,int,int,int,char*);

void           Box(int,int,int,int);

void           Bar(int,int,int,int);

void           SetColor(unsigned char**);

void           SetGray(void);

void           Open256(void);

void           Close256(void);

void           Display(void);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void                Display1(void);
void SetCon_Brig(unsigned*,unsigned*);
void                chk_ID_card(void);
void digitizeBW(unsigned,unsigned);
void                read_addr(void);
void                vdo2buff(int);
void                vdo2buff1(void);
void                vdo2buff2(void);
void                cal_img(void);
void                Mov_x(int);
void                Mov_y(int);
void                RotateX(int,int);
void                RotateY(int,int);
void                HomePosition(void);
void MoveManual(unsigned,unsigned);
void                Man_Ctrl(void);
void                cursor_off(void);

/*****/
/*    Set Mode Display    */
/*****/
void SetMode(unsigned char mode)
{
    _AH = 0;
    _AL = mode;
    geninterrupt(0x10);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/******
```

```
/* Set Font vga card */
```

```
*****
```

```
void Open256(void)
```

```
{
```

```
int i;
```

```
_AH = 0x11;
```

```
_AL = 0x30;
```

```
_BH = 0x03;
```

```
geninterrupt(0x10);
```

```
FONT = MK_FP(_ES,_BP);
```

```
OldMode = peekb(0,0x449);
```

```
VGAMAP = (unsigned char**)malloc(sizeof(unsigned char*)*256);
```

```
for(i=0;i<256;i++)
```

```
VGAMAP[i]=(unsigned char*)malloc(sizeof(unsigned char)*3);
```

```
SetMode(0x13);
```

```
}
```

```
/******
```

```
/* Set Gray Level */
```

```
*****
```

```
void SetGray(void)
```

```
{
```

```
int i;
```

```
for(i=0;i<256;i++)
```

```
{
```

```
VGAMAP[i][0]=(unsigned char)i*0.25;
```

```
VGAMAP[i][1]=(unsigned char)i*0.25;
```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        VGAMAP[i][2]=(unsigned char)i*0.25;

    }

    SetColor(VGAMAP);

}

/*****/

/*  Close Font vga card  */

/*****/

void Close256(void)
{
    int i,j;

    for(j=0;j<64;j++)
    {
        for(i=0;i<256;i++)
        {
            VGAMAP[i][0]--;
            VGAMAP[i][1]--;
            VGAMAP[i][2]--;
        }

        SetColor(VGAMAP);

        delay(25);
    }

    free(VGAMAP);

    SetMode(0x03);

}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*****/

/*  Set Color  */

/*****/

void SetColor(unsigned char **Vgamap)
{
    int i;
    for(i=0;i<255;i++)
    {
        outportb(0x3c8,(unsigned char) i);
        outportb(0x3c9,(unsigned char)Vgamap[i][0]);
        outportb(0x3c9,(unsigned char)Vgamap[i][1]);
        outportb(0x3c9,(unsigned char)Vgamap[i][2]);
    }
    outportb(0x3c8,(unsigned char) i);
    outportb(0x3c9,(unsigned char)63);
    outportb(0x3c9,(unsigned char)63);
    outportb(0x3c9,(unsigned char)63);
}

/*****/

/*  Display Font to Monitor  */

/*****/

void PutChar(int X,int Y,int Color,int S,char ch)
{
    char far *Fptr;
    int i,j;

    Fptr = FONT +(unsigned int)ch*0x0008; X += S*S;
    for(i=0;i<8;i++)
        for(j=0;j<8;j++)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if(((*(Fptr+i))>>j)&0x01) != 0 )

            Box(X-(j*S),Y+(i*S),S,Color);

    }

    /*****
    /* Display Font is Text to Monitor */
    *****/

void OutText(int X,int Y,int C,int S,char *str)

{
    char i=0;

    for(i=0;i<strlen(str);i++)
        PutChar(X+(i*8*S),Y,C,S,str[i]);
}

/*****
/* Set size of Font */
*****/

void Box(int X,int Y,int S,int Color)

{
    int i,j;

    for(i=0;i<S;i++)
        for(j=0;j<S;j++) PutPixel((X+j),(Y+i),Color);
}

```

```

/*****/
/*  Set Display Position */
/*****/
void Bar(int X1,int Y1,int X2,int Y2)
{
    int i,j;
    for(i=Y1;i<Y2;i++)
        for(j=X1;j<X2;j++) PutPixel(j,i,i);
}

```

```

/*****/
/*  Set Display Main      */
/*****/
void Display(void)
{
    int x=1,y=135;
    OutText(x,y,128,1,"  Object Following Program      By 2Q");
    OutText(x,y+15,128,1,"                Key Control System ");
    OutText(x,y+30,128,1," F1=>Set   Color      F2=>Set. Original");
    OutText(x,y+40,128,1," F3=>Home Position  F4=>Run");
    OutText(x,y+50,128,1," F5=>Move Manual   Esc=>Exit");
}

```

```

/*****/

/*  Set Diplay Subroutine  */

/*****/

void Display1(void)
{
    int x=1,y=135;

    OutText(x,y,128,1,"  Set Contrast and Brightness ");
    OutText(x,y+15,128,1,"          Key Control System  ");
    OutText(x,y+30,128,1," F1=>Contrast   Left Arrow=>DEC");
    OutText(x,y+40,128,1," F2=>Brightness Right Arrow=>INC");
    OutText(x,y+50,128,1," N  => Normal  Esc  => Exit    ");
}

```

```

/*****/

/*  Set Contrast and Brightness  */

/*****/

void SetCon_Brig(unsigned *contrast, unsigned *brightness)
{
    char ch;
    int cnt,set_cont=0;

    SetMode(0x03);
    SetMode(0x13);
    SetGray();
    Display1();

    do
    {
        digitizeBW(*contrast,*brightness);

        vdo2buff1();
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(kbhit())
{
    ch = getch();
    switch(ch)
    {
        case 00 : ch=getch();
                switch(ch)
                {
                    case 59 : set_cont = 0; /*Set contrast*/
                                break;
                    case 60 : set_cont = 1;
                                break;
                    case 77 : if(set_cont == 0)
                                {
                                    if(*contrast<63)
                                        (*contrast)++;
                                    else
                                        (*contrast) = 63;
                                }
                                else
                                {
                                    if(*brightness<63)
                                        (*brightness)++;
                                    else
                                        (*brightness) = 63;
                                }
                                break;
                    case 75 : if(set_cont == 0)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if(*contrast>0)

            (*contrast)--;

        else

            (*contrast) = 0;

    }

    else

    {

        if(*brightness>0)

            (*brightness)--;

        else

            (*brightness) = 0;

    }

    break;

}

break;

case 'n' : *contrast = nrm_cont;

           *brightness = nrm_brig;

           break;

case 'N' : *contrast = nrm_cont;

           *brightness = nrm_brig;

           break;

}

}

}

} while(ch != 27);

Teshold = Histogram();

}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*****/
/*          check ID card          */
/*****/

void chk_ID_card(void)
{
    int port[]={0x210,0x2a0,0x310,0x3a0};
    int index=0,cnt;
    unsigned swport,time,chk1,chk2,chkport;

    for(cnt=1;cnt!=4;cnt++)
    {
        ctrlport=port[index];
        swport=ctrlport+1;
        chkport=ctrlport | 0x8001;
        chk1=inportb(swport);
        chk1=inportb(swport);
        chk1=(chk1 ^ 0xf) & 0xf;
        chk2=chk1;

        for(time=0;time<200;time++)
        {
            chk1=inportb(chkport);
            chk1=(chk1 ^ 0xf) & 0xf;
            if(chk1!=chk2) goto nextbase;
            chk2=(chk2 ^ 0xf) & 0xf;
        }

        return;
    nextbase:
        index++;
    }
}

```

3
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SetMode(0x03);

printf("Card DZ-II Not Found !!");

,   exit(1);
}

/*****

/*      check digit state      */

*****/

void chk_digi_stat()
{
    int pulse,time;

    for(pulse=1;pulse!=0;pulse--)
    {
        time=0;
        while((inportb(ctrlport)>>5)&1!=pulse)
        {
            time++;
            if(time>4000)
            {
                SetMode(0x03);
                printf("video signal not found");
                exit(1);
            }
        }
    }
}
}
}

```

```

/*****/
/*          Digitize B&W          */
/*****/

void digitizeBW(unsigned contrast,unsigned brightness)
{
    unsigned digicom_y=0x80;
    unsigned digicomstop=0;
    unsigned ref_neg;

    ref_neg=63-brightness;
    outportb(ctrlport+2,ref_neg);
    outportb(ctrlport+1,ref_neg + ((63-ref_neg)*(63-contrast))/63);

    outportb(ctrlport,digicom_y);
    delay(40);
    chk_digi_stat(ctrlport);
    outportb(ctrlport,digicomstop);
}

/*****/
/*          READ ADDRESS          */
/*****/

void read_addr(void)
{
    unsigned chksw,swport;

    swport=ctrlport+1;
    chksw=inportb(swport);
    chksw=inportb(swport);

    chksw=chksw & 0xe;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

switch(chksw)
{
    case 0:cardseg=0xc400;break;
    case 2:cardseg=0xc800;break;
    case 4:cardseg=0xcc00;break;
    case 6:cardseg=0xd000;break;
    case 8:cardseg=0xd800;break;
    case 10:cardseg=0xdc00;break;
    case 12:cardseg=0xe000;break;
    case 14:cardseg=0x9000; /*NULL*/;break;
}

```

```

}

```

```

/*****

```

```

/* MOVE data form VIDEO RAM to obj */

```

```

*****/

```

```

void vdo2buff(int temp)

```

```

{

```

```

    int cnt,cnt1;

```

```

    unsigned char tuk;

```

```

    outportb(ctrlport,0);

```

```

    for(cnt=0;cnt<=63;cnt++)

```

```

    {
        for(cnt1=0;cnt1<=127;cnt1++)

```

```

        {

```

```

            tuk = (peekb(mem_add,cnt*256+cnt1) >> 2)&0x3F;

```

```

            tuk = tuk >= temp ? 250 : 15;

```

```

            pokeb(mem_vrm,cnt*320+cnt1,tuk);

```

```

        }
    }
}

```

outportb(ctrlport,1);การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(cnt=0;cnt<=63;cnt++)
    for(cnt1=0;cnt1<=127;cnt1++)
    {
        tuk = (peekb(mem_add,cnt*256+cnt1) >> 2)&0x3F;
        tuk = tuk >= temp ? 250 : 15;
        pokeb(mem_vrm,(cnt+64)*320+cnt1,tuk);
    }
}

/*****
/* MOVE data form VIDEO RAM to obj */
*****/
void vdo2buff1(void)
{
    unsigned cnt,cnt1;

    outportb(ctrlport,0);
    for(cnt=0;cnt<=63;cnt++)
        for(cnt1=0;cnt1<=127;cnt1++)
        {
            pokeb(mem_vrm,cnt*320+cnt1+172,peekb(mem_add,
                cnt*256+cnt1));
            original[cnt*128+cnt1] = (peekb(mem_add,cnt*256+cnt1)>>2)
                &0x3F;
        }

    outportb(ctrlport,1);
    for(cnt=0;cnt<=63;cnt++)
        for(cnt1=0;cnt1<=127;cnt1++)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        pokeb(mem_vrm, (cnt+64)*320+cnt1+172, peekb(mem_add,
            cnt*256+cnt1));

    original[(cnt+64)*128+cnt1] = (peekb(mem_add, cnt*256+cnt1)
        >>2)&0x3F;

}

}

; /*****
;
;  /*  MOVE data form VIDEO RAM to obj  */
;
; *****/
void vdo2buff2(void)
{
    unsigned cnt, cnt1;

    outportb(ctrlport, 0);
    for(cnt=0; cnt<=63; cnt++)
        memcpy(MK_FP(mem_vrm, cnt*320+172), MK_FP(mem_add, cnt*256),
            127);

    outportb(ctrlport, 1);
    for(cnt=0; cnt<=63; cnt++)
        memcpy(MK_FP(mem_vrm, (cnt+64)*320+172), MK_FP(mem_add, cnt*256),
            127);

}

```

```

/*****
/*          Histogram          */
*****/

int Histogram(void)
{
    int row,column;

    int loop,cnt;

    int temp=0,Level=0;

    for (column=0;column<64;column++)
        his_value[column]=0;

    for (row=0;row<128;row++)
        for (column=0;column<128;column++)
        {
            cnt = (int)original[128*row+column];
            his_value[cnt]++;
        }

    for(loop=0;loop<64;loop++)
        if(temp<=his_value[loop])
        {
            temp = his_value[loop];
            Level = loop;
        }

    return Level;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*****/

/*  Check Object  */

/*****/

int Check_Obj(void)
{
    int scan_x,scan_y,cnt;

    int Done=1;

    scan_x=0;scan_y=0; cnt=0;

    while(Done)
    {
        if(peekb(mem_vrm,scan_y*320+scan_x)==15)
            cnt++;
        else
            cnt = 0;

        if(scan_x==127)
        {
            scan_y++;
            scan_x=0;
        }
        else
            scan_x++;

        if(cnt==Error || scan_y==127)
            Done = 0;
    }

    return scan_y;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*****/

/* Calculate imgae */

/*****/

void cal_img(void)
{
    int scan_x,scan_y,cnt;
    int current_x,current_y;
    int post_x,post_y;
    int negt_x,negt_y;
    int Done=1;

    scan_y=0; scan_x=0; cnt=0;
    while(Done)
    {
        if(peekb(mem_vrm,scan_y*320+scan_x)==15)
            cnt++;
        else
            cnt = 0;

        if(scan_x==127)
        {
            scan_y++;
            scan_x=0;
        }
        else
            scan_x++;

        if(cnt==Error || scan_y == 127)
            Done = 0;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
post_y=scan_y;          /* post_y = position +Y axis */
```

```
scan_x=0; scan_y=127; cnt=0; Done=1;
```

```
while(Done)
```

```
{
```

```
    if(peekb(mem_vrm,scan_y*320+scan_x)==15)
```

```
        cnt++;
```

```
    else
```

```
        cnt = 0;
```

```
    if(scan_x==127)
```

```
    {
```

```
        scan_y--;
```

```
        scan_x=0;
```

```
    }
```

```
    else
```

```
        scan_x++;
```

```
    if(cnt==Error || scan_y==0)
```

```
        Done = 0;
```

```
}
```

```
negt_y=scan_y;          /* negt_p = position -Y axis */
```

```
scan_y=0; scan_x=127; cnt=0; Done=1;
```

```
while(Done)
```

```
{
```

```
    if(peekb(mem_vrm,scan_y*320+scan_x)==15)
```

```
        cnt++;
```

```
    else
```

```
        cnt = 0;
```

```
    if(scan_y==127)
```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    scan_x--;
    scan_y=0;
}
else
    scan_y++;

if(cnt==Error || scan_x==0)
    Done = 0;
}

post_x=scan_x;      /* post_x = position +X axis */

scan_x=0; scan_y=0; cnt=0; Done=1;
while(Done)
{
    if(peekb(mem_vrm,scan_y*320+scan_x)==15)
        cnt++;
    else
        cnt = 0;

    if(scan_y==127)
    {
        scan_x++;
        scan_y=0;
    }
    else
        scan_y++;

    if(cnt==Error || scan_x==127)
        Done=0;
}

```

```

/*****
/* Rotate X axis */
*****/

void RotateX(int Dir,int Time)
{
    unsigned char Value=0;

    if(Dir==Right_X)
    {
        if(Col_X<7)
            Col_X++;
        else
            Col_X=0;
        Abs_X++;
    }
    else if(Dir==Left_X)
    {
        if(Col_X>0)
            Col_X--;
        else
            Col_X=7;
        Abs_X--;
    }

    if(Table[Col_X][0]==1) Value |= 0x01;
    if(Table[Col_X][1]==1) Value |= 0x02;
    if(Table[Col_X][2]==1) Value |= 0x04;
    if(Table[Col_X][3]==1) Value |= 0x08;

    outp(PortA,Value);

    delay(Time);
}

```

```

/*****
/* Manual control motor */
*****/

void Man_Ctrl(void)
{
    char ch;

    int a,b, cnt1;

    clrscr();

    outp(PortCtrl,0x80);    /* set control_port of 8255 */

    outp(PortA,0xf0);
    outp(PortB,0xf0);
    printf("\n");
    printf("                                \n");
    printf("          SET POINT ORIGINAL OF X,Y AXIS          \n");
    printf("  Cursor right = +X aixs   Cursor left = -X aixs   \n");
    printf("  Cursor up    = +Y aixs   Cursor down = -Y aixs   \n");
    printf("  Esc  to Quit                                \n");
    printf("                                \n");
    cursor_off();

    do
    {
        if(kbhit())
        {
            ch = getch();

            if(ch==0)
            {
                ch = getch();

                switch(ch)
                {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/**/
```

```
/* Set Cursor Off */
```

```
/**/
```

```
void cursor_off()
```

```
{
```

```
    union REGS reg;
```

```
    reg.h.ch = 0x20;
```

```
    reg.h.cl = 0;
```

```
    reg.h.ah = 1;
```

```
    int86(0x10,&reg,&reg);
```

```
}
```

```
/**/
```

```
/*          MAIN          */
```

```
/**/
```

```
void main(void)
```

```
{
```

```
    unsigned contrast=32,brightness=50;
```

```
    char ch,key;
```

```
    int x=1,y=135;
```

```
    int Run;
```

```
    prt=(unsigned char far*)farmalloc(16384);
```

```
    if(prt==NULL)
```

```
    {
```

```
        printf("Allocate failed memory\n");
```

```
        exit(1);
```

```
    }
```

```
    original = (unsigned char far*)farmalloc(16384);
```

```
    if(original==NULL)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{

    printf("Allocate failed memory\n");

    exit(1);

}

Man_Ctrl();

Open256();

SetGray();

chk_ID_card();

read_addr();


OutText(x,y,128,1,"      Object Following Program");
OutText(x,y+15,128,1,"      Key Control System  ");
OutText(x,y+30,128,1," F1=>Set Color");
OutText(x,y+40,128,1," F2=>Manual Control  Esc=>Exit");
OutText(x,y+55,128,1,"      By 2Q <KMITL>");


do
{
    digitizeBW(contrast,brightness);
    vdo2buff1();
    if(kbhit())
    {
        key = getch();
        if(key==0)
        {
            key = getch();
            switch(key)
            {

```

```

        case 59 : SetCon_Brig(&contrast,&brightness);

                key = 27;

                break;

        case 60 : SetMode(0x03);

                Man_Ctrl();

                break;

    }

}

}

} while(key != 27);

key = 30;

SetMode(0x03);
SetMode(0x13);
SetGray();
Display();

Run=0;
do
{
    digitizeBW(contrast,brightness);
    vdo2buff(Teshold);
    vdo2buff2();
    if(Run==1)
    {
        if(Check_Obj()<127)

            cal_img();

        else

            HomePosition();
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(kbhit())
{
    key = getch();

    if(key==0)
    {
        key = getch();

        switch(key)
        {
            case 59 : SetCon_Brig(&contrast,&brightness);
                      SetMode(0x03);
                      SetMode(0x13);
                      SetGray();
                      Display();
                      break;
            case 61 : HomePosition(); break;
            case 60 : SetMode(0x03);
                      Man_Ctrl();
                      SetMode(0x13);
                      SetGray();
                      Display();
                      break;

            case 62 : Run = (Run==1) ? 0 : 1; break;
            case 63 : MoveManual(contrast,brightness);
                      SetMode(0x03);
                      SetMode(0x13);
                      SetGray();
                      Display();
                      break;
        }
    }
}

```

}

}

```
} while(key != 27);
```

```
outp(PortA,0x00);
```

```
outp(PortB,0x00);
```

```
farfree(prt);
```

```
farfree(original);
```

```
SetMode(0x03);
```

}



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้