**บทที่ 2**

**หลักการและทฤษฎีที่เกี่ยวข้อง**

**2.1 ชนิดของหมวกนิรภัย**

**2.1.1 หมวกนิรภัยแบบครึ่งศีรษะ** หมวกนิรภัยชนิดนี้เป็นหมวกที่นำเข้ามาใช้เป็นชนิดแรก มีลักษณะเป็นรูปครึ่งทรงกลม โดยด้านข้างและด้านหลังปิดเสมอระดับหู มีสายรัดคาง สามารถคลุมได้ครึ่งศีรษะ [1]



**รูปที่ 2.1** แสดงหมวกนิรภัยแบบครึ่งศีรษะ

ที่มา: สถาบันหมวกนิรภัย เว็บไซต์ http://wisdom7.sitthichoke.info/safety-helmet/

**2.1.2 หมวกนิรภัยแบบเต็มศีรษะ** หมวกนิรภัยชนิดนี้เป็นหมวกนิรภัยที่ตัดแปลงมาจากหมวกแบบครึ่งศีรษะ มีลักษณะเป็นทรงกลม ปิดด้านข้างและด้านหลังเสมอแนวขากรรไกรและต้นคอ ด้านหน้าเปิดเหนือคิ้วลงมาถึงปลายคาง มีสายรัดคาง [1]



**รูปที่ 2.2** แสดงหมวกนิรภัยแบบเต็มศีรษะ

ที่มา: สถาบันหมวกนิรภัย เว็บไซต์ http://wisdom7.sitthichoke.info/safety-helmet/

**2.1.3 หมวกนิรภัยแบบเต็มหน้า** หมวกนิรภัยชนิดนี้มีลักษณะเป็นทรงกลม ปิดด้านข้าง ด้านหลัง ขากรรไกร และคาง เปิดช่องหน้าตรงตำแหน่งตาเท่านั้น มีสายรัดคาง [2]



**รูปที่ 2.3** แสดงหมวกนิรภัยแบบเต็มหน้า

ที่มา: สถาบันหมวกนิรภัย เว็บไซต์ http://wisdom7.sitthichoke.info/safety-helmet/

**2.2 ความรู้พื้นฐานเกี่ยวกับรูปภาพ**

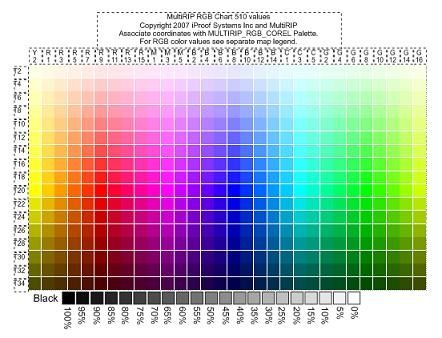
**2.2.1 รูปร่างของภาพ [3]**

วัตถุต่างๆ ที่มีอยู่ในโลกทั้งที่มนุษสร้างขึ้น และเกิดเองตามธรรมชาติ ย่อมมีรูปร่าง และขนาดแตกต่างกันออกไป ในศาสตร์ของการประมวลผลภาพ จึงมีการกำหนดขอบเขตของภาพทุกภาพให้อยู่ในรูปสี่เหลี่ยม (Rectangular image model) เพื่อทำให้การจัดเก็บข้อมูล และการอ่านภาพเป็นไปได้อย่างมีประสิทธิภาพ

โดยการจัดเก็บข้อมูลภาพลงหน่วยความจำของคอมพิวเตอร์จะทำได้โดยการจองพื้นที่ไว้ในรูปของตัวแปรอาร์เรย์ (Array) ขนาด M x N (M แถว และ N คอลัมน์) ใช้เก็บภาพขนาด M x N จุด ซึ่งแต่ละช่องตำแหน่งของอาร์เรย์จะแสดงคุณสมบัติของจุดภาพ (Pixel)

**2.2.2 ระบบสี RGB (Red/Green/Blue) [4]**

ระบบสี RGB เป็นระบบสีในคอมพิวเตอร์ที่มาจากสีพื้นฐานที่ทุกคนรู้จัก คือ สีแดง สีเขียว และสีน้ำเงิน โดย RGB เกิดจากการผสมสีที่มีอัตราส่วนของสีทั้งสามสีนั้นอยู่ในช่วง 0 ถึง 255 ซึ่งค่า 0 คือสีดำ และ 255 คือสีขาว ทั้งสามสีนั้นจะมีค่าสีทั้งหมด 256 สี คือ 0 ถึง 255 โดยมองเป็นค่าระดับเทา เมื่อเราจะใช้ก็ผสมทั้งสามสีเข้าด้วยกันให้ได้สีที่ต้องการ จึงสามารถผสมสีออกมาได้ถึง 16,777,216 สี



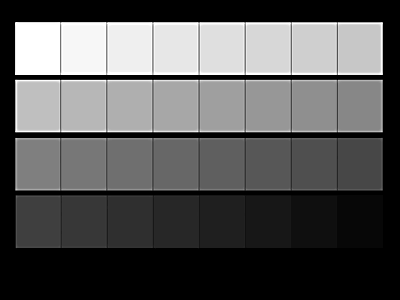
**รูปที่ 2.4** ระบบสี RGB

ที่มา: เว็บไซต์ http://www.qghservice.com/images/rgb.JPG

**2.2.3 ภาพระดับสีเทา (Grayscale Image) [5]**

ภาพระดับสีเทา เกิดจากค่าความเข้มแสงในแต่ละตำแหน่งของจุดภาพ โดยค่าความเข้มแสงหรือค่าระดับเทามีค่าเท่ากับ 2n เมื่อ n คือจำนวนบิตของภาพ เช่นภาพขนาด 8 บิต จะมีค่าความเข้มแสงหรือค่าระดับเทาทั้งหมด 28 = 256 ระดับ คือตั้งแต่ 0 ถึง 255 การแปลงภาพจาก RGB มาเป็นภาพระดับสีเทา สามารถทำได้ดังสมการนี้

Gray = (0.3× R) + (0.59×G) + (0.11× B) [4]



**รูปที่ 2.5** ค่าระดับเทาขนาด 4 บิต

ที่มา: http://www.blackandwhitedigital.com/images/gif/thumbnail/T\_GreyscaleOriginal.gif

**2.2.4 ภาพขาว-ดำ (Binary Image) [5]**

ภาพขาว-ดำ ในภาพจะประกอบไปด้วยสีขาว และสีดำเท่านั้น ซึ่งค่าที่เป็นไปได้จะมีเพียงสองค่า คือ 0 (สีดำ) และ 255 (สีขาว) จึงมีการเก็บค่าภาพเพียงแค่ 1 บิต



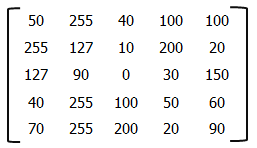
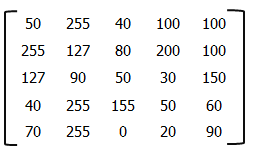
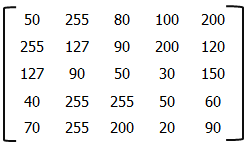
**รูปที่ 2.6** แสดงภาพขาว-ดำ

**2.3 Motion Detect**

เป็นกระบวนการที่ใช้ในการตรวจจับความเคลื่อนไหวของภาพ โดยอาศัยหลักการความแตกต่างของภาพทั้งสองภาพ (Background Subtraction) สามารถแทนด้วยสมการดังนี้

*g(x , y) = f( x , y) – h(x, y)*

โดยกำหนดให้ g(x, y) = ผลลัพธ์ของความต่างทั้งของภาพ f(x, y) = เฟรมภาพอดีต h(x, y) = เฟรมภาพปัจจุบัน



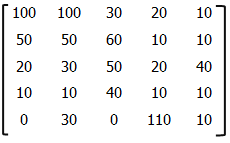
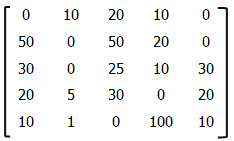
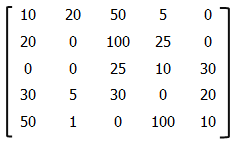
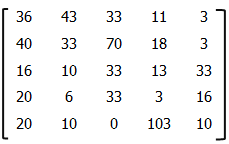
*g(x, y)* เฟรมภาพที่ n เฟรมภาพที่ n-1

**รูปที่ 2.7** กระบวนการ Motion Detect

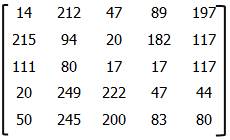
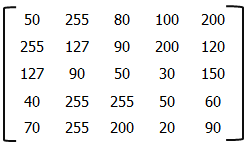
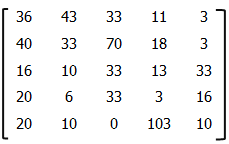
จากรูปที่ 2.7 แสดงตัวอย่างการหาความแตกต่างของภาพทั้งสองภาพได้ผลลัพธ์ออกมาเป็นภาพ g(x, y) ซึ่งภาพ g(x, y) นี้จะแสดงถึงวัตถุที่กำลังเคลื่อนไหว ข้อจำกัดของการหาความต่างของภาพคือ ภาพทั้งสองภาพต้องมีขนาดเท่ากันและต้องเป็นภาพประเภทเดียวกัน [5]

**2.4 Average Motion Detect**

เป็นกระบวนการที่นำการหาความแตกต่างของภาพมาประยุกต์ใช้เพื่อให้เกิดความแม่นยำมากขึ้น โดยนำภาพจำนวน n ภาพมาหาค่าเฉลี่ยของแต่ละพิกเซล แล้วนำไปสร้างเป็นภาพใหม่ เพื่อที่จะนำไปใช้ในการหาความแตกต่างของภาพที่กำลังเข้ามาในเฟรมปัจจุบัน ดังตัวอย่างรูปที่ 2.8



Average frame frame n-1 frame n-2 frame n-3



*g(x, y)* frame n averge frame

**รูปที่ 2.8** กระบวนการ Average Motion Detect

**2.5 การทำ Threshold [6]**

การทำ Threshold คือการแปลงภาพระดับสีเทา ให้กลายเป็นภาพขาว-ดำ (Binary Image) โดยจะให้จุดภาพ (Pixel) มีขนาด 1 บิตเท่านั้นคือ 0 (สีดำ) และ 1 (สีขาว) ซึ่งจะสามารถช่วยแยกวัตถุออกจากพื้นหลัง วิธีที่ใช้ในการแปลงภาพให้เป็นขาว-ดำ ทำได้โดยการกำหนดค่า Threshold ขึ้นมาค่าหนึ่ง หากจุดในภาพมีค่าความเข้มแสงมากกว่าค่า Threshold จุดนั้นจะถูกปรับให้เป็นสีดำ หรือถ้าจุดใดในภาพมีค่าความเข้มแสงน้อยกว่าหรือเท่ากับค่า Threshold จุดนั้นก็จะถูกปรับให้เป็นสีขาว



**รูปที่ 2.9** แสดงรูปที่เกิดจากการทำ Threshold

**2.5.1 Otsu Threshold [7]**

การทำ Otsu Threshold เป็นการทำ Threshold วิธีหนึ่งซึ่งมีหลักการคือ วัดระยะห่างในฮิสโตแกรมระหว่าง 2 บริเวณหรือ 2 ยอด โดยให้ระยะห่างของทั้ง 2 บริเวณมีค่ามากที่สุด เพราะบริเวณ 2 บริเวณจะได้แยกออกจากกันอย่างชัดเจน โดยจะต้องเลือกค่า Threshold ที่ทำให้ผลการคำนวณมีค่ามากที่สุด โดยสามารถคำนวนได้จากสมการ

เมื่อ

คือ ความแปรปรวนรวมของทั้งฮิสโตแกรม (ต้องให้มีค่ามากที่สุด)

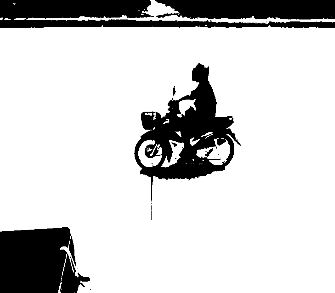
คือ ค่า Threshold ที่ต้องการหา

คือ จำนวนพิกเซลทั้งหมดของบริเวณด้านมืด ที่มีค่าความเข้มแสง ตั้งแต่ 0 จนถึงค่าความเข้มแสงเท่ากับ T - 1

คือ จำนวนพิกเซลทั้งหมดของด้านสว่าง ที่มีค่าความเข้มแสงตั้งแต่ T จนถึง 2B – 1 เมื่อ B คือจำนวนบิตของระบบภาพ

คือ ค่าเฉลี่ยของฮิสโตแกรมฝั่งด้านมืด

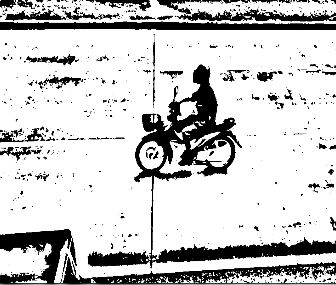
คือ ค่าเฉลี่ยของฮิสโตแกรมฝั่งด้านสว่าง



**รูปที่ 2.10** แสดงรูปที่เกิดจากการทำ Otsu Threshold

**2.5.2 Adaptive Threshold [6]**

Adaptive Threshold เป็นการทำ Threshold ที่เหมาะกับภาพที่มีความสว่างไม่สม่ำเสมอ เพราะเป็นการกำหนดค่า Threshold ขึ้นมาหลายค่า โดยสร้าง window ขนาด N x N ขึ้นมา (แนะนำให้ N เป็นเลขจำนวนคี่) แล้วนำไปวางครอบบนบริเวณใดบริเวณหนึ่งของภาพ จากนั้นนำค่า Grayscale ของแต่ละจุดภาพมาบวกกันแล้วหารด้วยจำนวนจุดทั้งหมด จะได้ ค่า Threshold มาเพื่อนำไปเปรียบเทียบแบบการทำ Threshold ปกติ โดยจะทำแบบนี้ทุกบริเวณที่ไม่ซ้ำกันของภาพ ก็จะได้ภาพขาวดำที่มีความละเอียดเพิ่มขึ้นอีก



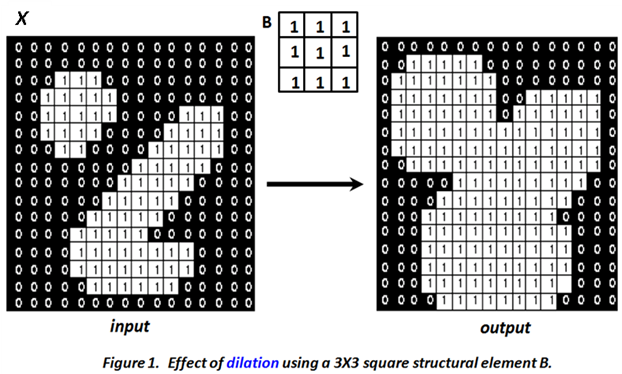
**รูปที่ 2.11** แสดงรูปที่เกิดจากการทำ Adaptive Threshold

**2.6 Morphological Image Processing [5]**

Morphological Image Processing เป็นทฤษฎีการประมวลผลภาพที่ใช้วิธีการจัดองค์ประกอบภาพขึ้นมาใหม่ โดยใช้การแทนที่หรือการตัดออก เพื่อให้ได้โครงสร้างภาพที่เหมาะสมสำหรับภาพขาว-ดำ (Binary Image)

**2.6.1 การขยายขนาดภาพ (Dilation) [5]**

เป็นการขยายขนาดของวัตถุในภาพเพื่อให้วัตถุที่ไม่มีสมาชิกร่วมกัน แต่อยู่ใกล้กันสามารถเชื่อมกันได้ โดยมีวิธีการดังนี้คือ ถ้าจุดศูนย์กลางมีค่าความเข้มแสงเป็น 0 ให้เลื่อนไปดูที่จุดถัดไป แต่ถ้าเป็น 1 ให้ทำการ OR กับสมาชิกที่อยู่รอบๆ แล้วเปลี่ยนค่าตามที่ออกมา

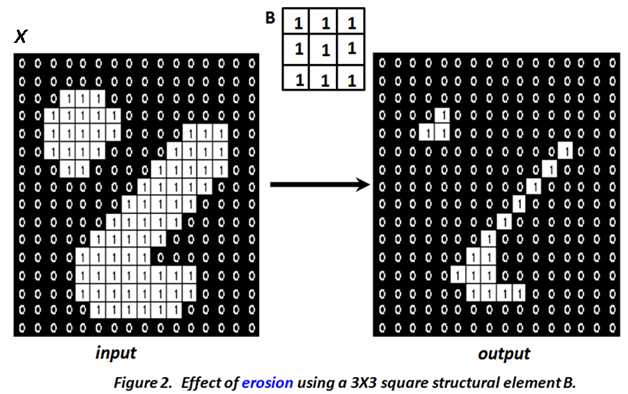


**รูปที่ 2.12** แสดงการเปรียบเทียบก่อน (ด้านซ้าย) – หลัง (ด้านขวา) การขยายขนาดภาพ

ที่มา: http://deploy.virtual-labs.ac.in/labs/cse19/theory\_html\_37e381f2.png

**2.6.2 การกร่อนข้อมูลภาพ (Erosion) [5]**

การกร่อนข้อมูลภาพเป็นการดำเนินการคล้ายๆ กับการขยายขนาดภาพ แต่เป็นการกระทำตรงกันข้าม คือลดขนาดของวัตถุในภาพให้เล็กลง โดยการเปลี่ยนแปลงค่าจุดที่เป็นขอบของวัตถุภายในภาพให้กลายเป็นค่า 0

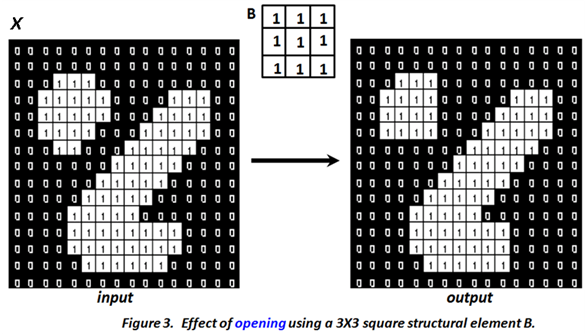


**รูปที่ 2.13** แสดงการเปรียบเทียบก่อน (ด้านซ้าย) – หลัง (ด้านขวา) การกร่อนข้อมูลภาพ

ที่มา: http://deploy.virtual-labs.ac.in/labs/cse19/theory\_html\_m23129e3e.png

**2.6.3 Opening and Closing Operation**

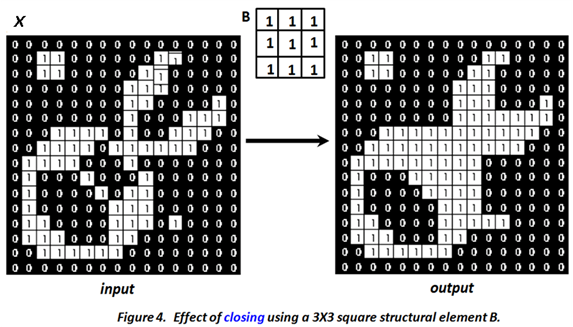
**2.6.3.1 Opening Operation** เป็นกระบวนการที่ช่วยในการกำจัดวัตถุขนาดเล็กออกไป เพื่อให้วัตถุขนาดใหญ่มีขอบที่เรียบและชัดเจนขึ้น โดยจะมีวิธีการทำงานคือ กร่อนข้อมูลภาพ (Erosion) ก่อน แล้วตามด้วยการขยายขนาดภาพ (Dilation)



**รูปที่ 2.14** แสดงการเปรียบเทียบก่อน (ด้านซ้าย) – หลัง (ด้านขวา) การทำ Opening Operation

ที่มา: http://deploy.virtual-labs.ac.in/labs/cse19/theory\_html\_m7e9b3438.png

**2.6.3.2 Closing Operation** เป็นกระบวนการที่ช่วยให้วัตถุในภาพที่เป็นช่องเล็กๆ ถูกเติมเต็ม ไม่ทำให้ภาพขนาดใหญ่ขึ้น แต่ขอบจะดูเรียบ โดยกระบวนการนี้ใช้วิธีขยายขนาดภาพ (Dilation) ก่อนแล้วจึงตามด้วยการกร่อนข้อมูลภาพ (Erosion)

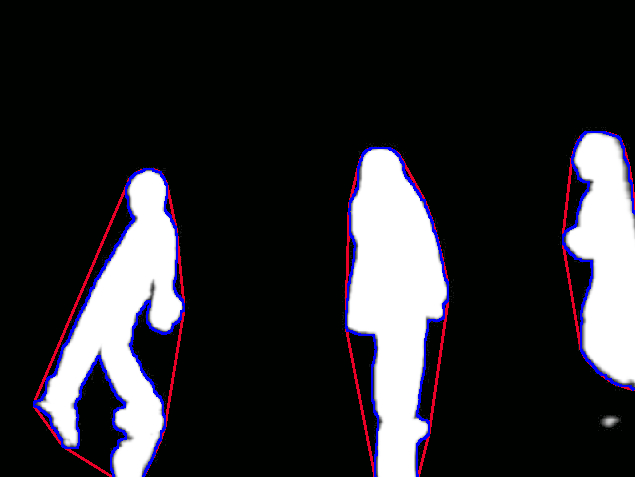


**รูปที่ 2.15** แสดงการเปรียบเทียบก่อน (ด้านซ้าย) – หลัง (ด้านขวา) การทำ Closing Operation

ที่มา: http://deploy.virtual-labs.ac.in/labs/cse19/theory\_html\_m594e600f.png

**2.7 Contours**

เป็นกระบวนการที่ใช้ในการหาขอบเขตของวัตถุในรูปภาพ เพื่อที่จะนำวัตถุนั้นมาใช้การในการวิเคราะห์ว่าวัตถุนั้นเป็นอะไร ดังตัวอย่าง



**รูปที่ 2.16** แสดงการทำ Contours

ที่มา: http://kineme.net/composition/benoitlahoz