**บทที่ 2**

**วรรณกรรมที่เกี่ยวข้อง**

การพัฒนาแอพลิเคชั่นแสดงข้อมูลสารสนเทศด้วย เทคโนโลยี AR Code บนสมาร์ตโฟน ที่ใช้ระบบปฎิบัติการแอนดรอยด์ ได้ทำการรวบรวมทฤษฎีเนื้อหาตลอดจนงานวิจัยที่เกี่ยวข้องไว้ดังนี้

1. ระบบปฎิบัติการแอนดรอย์
2. ระบบฐานข้อมูล mySQL
3. การทำงานของ www
4. ภาษาจาวา
5. การเขียนโปรแกรมเชิงวัตถุ
6. ยูเอ็มแอลไดอะแกรม
7. เว็บเซอร์วิส
8. การประมวลผลภาพดิจิตอล

**2.1.ระบบปฎิบัติการแอนดรอย์**

เป็นระบบปฎิบัติการที่มีพื้นฐานมาจาก ระบบปฎิบัติการลีนุกซ์ ซึ่งถูกออกแบบมาให้ใช้กับอุปกรณ์ที่ใช้ระบบสัมผัส เช่น สมาร์ตโฟน และ แท็บเล็ตคอมพิวเตอร์ แอยดรอย์ถูกคิดค้นและพัฒนาโดยบริษัท แอนดรอยด์ (Android, Inc.) เปิดตัวเมื่อปี พ.ศ. 2550 หลังจาก กูเกิลได้ซื้อมาพัฒนาต่อ พร้อมกับการก่อตั้งโอเพนเซตอัลไลแอนซ์ ซึ่งเป็นกลุ่มของบริษัทผลิตฮาร์ดแวร์, ซอฟต์แวร์ และการสื่อสารคมนาคม ที่ร่วมมือกันสร้างมาตรฐานเปิดสำหรับอุปกรณ์พกพา โดยสมารต์โฟนที่ใช้ระบบปฎิบัติการแอนดรอย์เครื่องแรกของโลกคือ เอซีที ดรีม วางจำหน่ายเมื่อ  
ปี 2551

แอนดรอยด์เป็นระบบปฏิบัติการโอเพนซอร์ซและกูเกิลได้เผยแพร่ภายใต้ลิขสิทธิ์อาปาเช ซึ่งโอเพนซอร์ซจะอนุญาตให้ผู้ผลิตปรับแต่งและวางจำหน่ายได้ รวมไปถึงนักพัฒนาและผู้ให้บริการเครือข่ายด้วย อีกทั้งแอนดรอยด์ยังเป็นระบบปฏิบัติการที่รวมนักพัฒนาที่เขียนโปรแกรมประยุกต์มากมาย ภายใต้ภาษาจาวา ในเดือนตุลาคม พ.ศ. 2555 มีโปรแกรมมากกว่า 700,000 โปรแกรมสำหรับแอนดรอยด์ และยอดดาวน์โหลดจากกูเกิลเพลย์ มากถึง 2.5 หมื่นล้านครั้ง จากการ

สำรวจในช่วงเดือน เมษายน ถึง พฤษภาคม ในปี พ.ศ. 2556 พบว่าแอนดรอยด์เป็น ระบบปฏิบัติการที่นักพัฒนาเลือกที่จะพัฒนาโปรแกรมมากที่สุด ถึง 71%

ปัจจัยเหล่านี้ทำให้แอนดรอยด์เป็นระบบปฏิบัติการที่ใช้กันอย่างแพร่หลายในปัจจุบันนำหน้าซิมเบียนในไตรมาสที่ 4 ของปี พ.ศ. 2553และยังเป็นทางเลือกของผู้ผลิตที่จะใช้ซอฟต์แวร์ที่มีราคาต่ำ, ตอบสนองความต้องการของผู้ใช้ได้ดี สำหรับอุปกรณ์ในสมัยใหม่แม้ว่าแอนดรอยด์จะดูเหมือนได้รับการพัฒนาเพื่อใช้กับสมาร์ตโฟนและแท็บเล็ต แต่มันยังสามารถใช้ได้กับโทรทัศน์,  
เครื่องเล่นวีดีโอเกมส์,กล้องดิจิตอล และอุปกรณ์อิเล็กทรอนิกส์อื่นๆ แอนดรอยด์เป็นระบบเปิด ทำให้นักพัฒนาสามารถพัฒนาคุณสมบัติใหม่ๆ ได้ตลอดเวลา

ส่วนแบ่งทางการตลาดของสมาร์ตโฟนแอนดรอยด์ นำโดยซัมซุง มากถึง 64% ในเดือนมีนาคม พ.ศ. 2556 เดือนกรกฎาคม พ.ศ. 2556 มีอุปกรณ์ที่ใช้ระบบปฏิบัติการแอนดรอยด์มากถึง 11,868 รุ่น จาก 8 เวอร์ชันของระบบปฏิบัติการแอนดรอยด์ความสำเร็จของระบบปฏิบัติการทำให้เกิดคดีด้านการละเมิดสิทธิบัตรที่เรียกกันว่า “สงครามสมาร์ตโฟน” (smartphone wars) ระหว่างบริษัทผู้ผลิตในเดือนพฤษภาคม พ.ศ. 2556 โปรแกรม 4.8 หมื่นล้านโปรแกรมได้รับการติดตั้งบนอุปกรณ์จากกูเกิลเพลย์และในวันที่ 3 กันยายน พ.ศ. 2556 มีอุปกรณ์ที่ใช้ระบบปฏิบัติการแอนดรอยด์ 1 พันล้านเครื่อง ได้ถูกเปิดใช้งาน

**2.2 ระบบฐานข้อมูล mySQL**

MySQL (มายเอสคิวแอล) เป็นระบบจัดการฐานจ้อมูลเชิงสัมพันธ์ (Relational Database Management System) โดยใช้

ภาษา SQLแม้ว่า MySQL เป็นซอฟต์แวร์โอเพนซอร์ส แต่แตกต่างจากซอฟต์แวร์  
โอเพนซอร์สทั่วไป โดยมีการพัฒนาภายใต้บริษัท MySQL ABในประเทศสวีเดน โดยจัดการ MySQL ทั้งในแบบที่ให้ใช้ฟรี และแบบที่ใช้ในเชิงธุรกิจ

MySQL สร้างขึ้นโดยชาวสวีเดน 2 คน และชาวฟินแลนด์ ชื่อ David Axmark, Allan Larsson และ Michael “Monty” Widenius.

ปัจจุบันบริษัทซันไมโครซิลเต็มส์ (Sun Microsystems, Inc.) เข้าซื้อกิจการของ MySQL AB เรียบร้อยแล้ว ฉะนั้นผลิตภัณฑ์ภายใต้ MySQL AB ทั้งหมดจะตกเป็นของซัน

ชื่อ “MySQL” อ่านออกเสียงว่า “มายเอสคิวเอล”หรือ “มายเอสคิวแอล” (ในการอ่านอักษร Lในภาษาไทย) ซึ่งทางซอฟต์แวร์ไม่ได้อ่าน มายซีเควล หรือ มายซีควล เหมือนกับซอฟต์แวร์จัดการฐานข้อมูลตัวอื่น

**2.3 การทำงานของ www**

คนส่วนใหญ่เข้าใจว่าอินเทอร์เน็ต กับ WWW คือสิ่งเดียวกัน แต่แท้ที่จริงแล้ว WWW เป้นเพียงบริการหนึ่งของอินเทอร์เน็ตเท่านั้น อินเทอร์เน็ตยังมีบริการอื่นๆ อีกด้วย

การทำงานของบริการ WWW นี้จะมีลักษณะเช่นเดียวกันกับบริการอื่นๆ ของอินเทอร์เน็ตคืออยู่ในรูปแบบไคลเอ็น-เซิร์ฟเวอร์ (Client-Server) โดยมีโปรแกรมเว็บไคลเอ็นต์ (web client)  
ทำหน้าที่เป็นผู้ร้องขอบริการ และมีโปรแกรมเว็บบราวเซอร์ (web browser) นั่นเอง สำหรับโปรแกรมเว็บเซิร์ฟเวอร์นั้นจะถูกติดตั้งไว้ในเครื่องของผู้บริการเว็บไซต์ การติดต่อระหว่างโปรแกรมเว็บบราวเซอร์กับโปรแกรมเว็บเซิร์ฟเวอร์จะกระทำผ่านโปรโตคอล HTTP (Hypertext Transfer Protocal)

**2.3.1 กลไกการทำงานของเว็บเพจ**

สำหรับเว็บเพจธรรมดาที่โดยปกติมีนามสกุลของไฟล์เป็น htm หรือ html นั้น เมื่อเราใช้เว็บบราวเซอร์เปิดดูเว็บเพจใด เว็บเซิร์ฟเวอร์ก็จะส่งเว็บเพ็จนั้นกลับมายังบราวเซอร์ จากนั้นบราวเซอร์จะแสดงผลไปตามคำสั่งภาษา HTML (Hypertext Markup Language) ที่อยู่ในไฟล์

จะเห็นได้ว่าเว็บเพจดังรูปเป็นเว็บเพจที่มีลักษณะ static กล่าวคือ ผู้ใช้จะพลกับเว็บเพจหน้าตาเดิมๆ ทุกครั้งจนกว่าผู้ดูแลเว็บจำ ทำการปรับปรุงเว็บเพจนั้น นี่คือข้อจำกัดอันมีต้นเหตุมาจากภาษา HTML ซึ่งเป็นภาษาที่ใช้อธิบายหน้าตาของเว็บเพจ หรือกล่าวอีกนัยหนึ่งคือ HTML สามารถกำหนดให้เว็บเพจมีหน้าตาอย่างที่เราต้องการได้ แต่ไม่ช่วยให้เว็บเพจมี “ความฉลาด” ได้

การสร้างเว็บเพจที่มีความฉลาดทำได้หลายวิธีด้วยกัน หนึ่งในนั้นก็คือ การฝังสคริปต์หรือชุดคำสั่งที่ทำงานทางฝั่งเซิร์ฟเวอร์ (Server-Side Script) ไว้ในเว็บเพจ

การทำงานของเว็บเพจที่ฝั่งสคริปต์ภาษา PHP ไว้ (ขอเรียกว่า ไฟล์ PHP) เมื่อเว็บบราวเซอร์ร้องขอไฟล์ PHP ไฟล์ใด เว็บเซิร์ฟเวอร์จะเรียก PHP engine ขึ้นมาแปล (interpret) และประมวลผลคำสั่งที่อยู่ในไฟล์ PHP นั้น โดยอาจมีการดึงข้อมูลจากฐานข้อมูลหรือเขียนข้อมูลลงไปยังฐานข้อมูลด้วย หลังจากนั้นผลลัพธ์ในรูปแบบ HTML จะถูกส่งกลับไปยังบราวเซอร์ บราวเซอร์ก็จะแสดงผลตามคำสั่ง HTML ที่ได้รับมา ซึ่งย่อมไม่มีคำสั่ง PHP ใดๆหลงเหลืออยู่ เนื่องจากถูกแปลและประมวลผลโดย PHP engine ที่ฝั่งเซิร์ฟเวอร์ไปหมดแล้ว

ให้สังเกตว่าการทำงานของบราวเซอร์ในกรณีนี้ไม่ต่างจากกรณีของเว็บเพจธรรมดาที่ได้อธิบายไปก่อนหน้านี้เลย เพราะสิ่งที่บราวเซอร์ต้องกระทำคือ การร้องขอไฟล์จากเว็บเซิร์ฟเวอร์จากนั้นก็รอรับผลลัพธ์กลับมาแล้วแสดงผล ความแตกต่างจริงๆ อยู่ที่การทำงานทางฝั่งเซิร์ฟเวอร์ ซึ่งกรณีหลังนี้ เว็บเพจที่เป็นไฟล์ PHP จะผ่านการประมวลผลก่อน แทนที่จะถูกส่งไปยังบราวเซอร์ เลยทันที

การฝังสคิปต์ PHP ไว้ในเว็บเพจ ช่วยให้เราสร้างเว็บเพจแบบ dynamic ได้ ซึ่งหมายถึงเว็บเพจที่มีเนื้อหาสาระและหน้าตาเปลี่ยนแปลงไปได้ในแต่ละครั้งที่ผู้ใช้เปิดดู โดยขึ้นอยู่กับเงื่อนไขต่างๆ เช่น ข้อมูลที่ผู้ใช้ส่งมาให้ หรือ ข้อมูลในฐานข้อมูล เป็นต้น

นอกจากนี้ยังมี จาวาสคริปต์ (JavaScript) ซึ่งเป็นภาษาสคริปต์ ที่มีลักษณะการเขียนแบบโปรโตไทพ์ (Prototyped-based Programming) ส่วนมากใช้ในหน้าเว็บเพื่อประมวลผลข้อมูลที่ฝั่งของผู้ใช้งาน แต่ก็ยังมีใช้เพื่อเพิ่มเติมความสามารถในการเขียนสคริปต์โดยฝังอยู่ในโปรแกรมอื่นๆ

ซัน ไมโครซิสเต็มส์เป็นเจ้าของเครื่องหมายการค้า “JavaScript” โดยมันถูกนำไปใช้ภายใต้สัญญาอนุญาตเพื่อการพัฒนาเทคโนโลยี เน็ตสเคป และมูลนิธิมอซิลลา

จาวาสคริปต์ เป็นภาษาในรูปแบบของภาษาโปรแกรมแบบโปรโตไทพ์ โดยมีใครสร้างของภาษาและไวยกรณ์อยู่บนพื้นฐานของภาษาซี

ปัจจุบันมีการใช้จาวาสคริปต์ที่ฝังอยู่ในเว็บบราวเซอร์ในหลายรูปแบบ เช่น ใช้เพื่อสร้างเนื้อหาที่เปลี่ยนแปลงเสมอภายในเว็บเพจ ใช้เพื่อตรวจสอบความถูกต้องของข้อมูลที่ผู้ใช้กรอกก่อนนำเข้าระบบ ใช้เพื่อเข้าถึงข้อมูลที่อยู่ภายใต้โครงสร้างแบบ Document Object Model (DOM)  
เป็นต้น

นอกจากนี้จาวาสคริปต์ยังถูกฝังอยู่ในแอปพลิเคชันต่างๆ นอกเหนือจากเว็บบราวเซอร์ได้อีกด้วย เช่น widget ของ Yahoo เป็นต้น โดยรวมแล้วจาวาสคริปต์ถูกใช้เพื่อให้นักพัฒนาโปรแกรม สามารถเขียนสคริปต์เพื่อสร้างคุณสมบัติพิเศษต่างๆ เพิ่มเติมจากที่มีอยู่บนแอปพลิเคชั่นดังเดิม

โปรแกรมใดๆ ที่สนับสนุนจาวาสคริปต์จะมีตัวขับเคลื่อนจาวาสคริปต์ (JavaScript Engine) ของตัวเอง เพื่อเรียกใช้งานโครงสร้างเซิงวัตถุของโปรแกรมหรือแอปพลิเคชั่นนั้น

**2.4 ภาษาจาวา**

ภาษาจาวา (Java Language) คือ ภาษาคอมพิวเตอร์ที่ถูกพัฒนาขึ้นโดยบริษัท ซันไมโครซิสเต็มส์ เป็นภาษาสำหรับเขียนโปรแกรมที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุ (OOP : Object-Oriented Programming) โปรแกรมที่เขียนขึ้นถูกสร้างภายในคลาส ดังนั้นคลาส (Class) คือที่เก็บเมทธอด (Method) หรือพฤติกรรม (Behavior) ซึ่งมีสถานะ (State) และรูปพรรณ (Identity) ประจำพฤติกรรม

ภาษาจาวา(Java Language) คือ ภาษาที่ถูกพัฒนาขึ้นโดยบริษัท ซันไมโครซิสเต็มส์   
(Sun Microsystems Inc.) เป็นภาษาสำหรับเขียนโปรแกรมภาษาหนึ่ง มีลักษณะสนับสนุนการเขียนโปรแกรมเชิงวัตถุ (OOP : Object-Oriented Programming) ที่ชัดเจน โปรแกรมต่าง ๆ ถูกสร้างภายใน class โปรแกรมเหล่านั้นถูกเรียกว่า method หรือ behavior โดยปกติจะเรียกแต่ละคลาสว่าวัตถุ โดยแต่ละวัตถุมีพฤติกรรมมากมาย โปรแกรมที่สมบูรณ์จะเกิดจากหลายวัตถุ หรือหลายคลาสมารวมกัน โดยแต่ละคลาสจะมีเมทธอด หรือพฤติกรรมแตกต่างกันไป

**2.5 การเขียนโปรแกรมเชิงวัตถุ**

OOP (Object Oriented Programming) เป็นวิธีการเขียนโปรแกรม โดยอาศัยแนวคิดของวัตถุชิ้นหนึ่ง มีความสามารถในการปกป้องข้อมูล และการสืบทอดคุณสมบัติ ซึ่งทำให้แนวโน้มของ OOP ได้รับการยอมรับและพัฒนามาใช้ในระบบต่าง ๆ มากมาย เช่น ระบบปฏิบัติการ วินโดวส์ เป็นต้น

ในที่นี้จะกล่าวถึงความเป็นมา และความหมายของแนวคิดการเขียนโปรแกรมเชิงวัตถุ ข้อควรรู้เกี่ยวกับหลักการเขียนโปรแกรมเชิงวัตถุ ตลอดจนความสำคัญของการเขียนและออกแบบระบบงานก่อนเขียนโปรแกรม รวมถึงประโยชน์ของการเขียนโปรแกรมเชิงวัตถุซึ่งจะทำให้ผู้เขียนได้เข้าใจถึงหลักการเบื้องต้นของแนวความคิดเชิงวัตถุนี้ได้

**ความเป็นมาของแนวคิดแบบ OOP**

แนวความคิดดั้งเดิมของการเขียนโปรแกรม ก็คือ การแก้ปัญหาโดยใช้คอมพิวเตอร์เป็นเครื่องมือ คล้ายกับการใช้เครื่องคิดเลขในการแก้ปัญหาทางคณิตศาสตร์ แนวความคิดแบบใหม่ที่ใช้ในการเขียนโปรแกรม ก็คือ การเน้นถึงปัญหาและองค์ประกอบของปัญหา (เพื่อแก้ปัญหา) การเน้นที่ปัญหาและองค์ประกอบของปัญหา (Problem Space) จะคล้ายกับแก้ไขปัญหาและชีวิตความเป็นอยู่ของมนุษย์ที่จะต้องมี คน สัตว์ สิ่งของ เพื่อแก้ปัญหา (มีหน้าที่แก้ปัญหา) มากกว่าจะมองที่วิธีการแก้ปัญหานั้น ๆ หรือขั้นตอนในการแก้ปัญหา (Solution Space) ซึ่งเป็นวิธีการเขียนโปรแกรมแบบเก่านั่นเอง

อาลัน เคย์ (Alan Kay) ได้เสนอกฎ 5 ข้อ ที่เป็นแนวทางของภาษาคอมพิวเตอร์เชิงวัตถุ หรือที่เรียกว่า Object-Oriented Programming (OOP) ไว้ดังนี้

ก. ทุก ๆ สิ่งเป็นวัตถุ (Everything is an Object)

ข. โปรแกรม ก็คือ กลุ่มของวัตถุที่ส่งข่าวสารบอกกันและกันให้ทำงาน (A Program is a Bunch of Object Telling Each Other What to do by Sending Messages)

ค. ในวัตถุแต่ละวัตถุจะต้องมีหน่วยความจำและประกอบไปด้วยวัตถุอื่น ๆ   
(Each Object has Its Own Memory Made Up of Other Objects)

ง. วัตถุทุกชนิดจะต้องจัดอยู่ในประเภทใดประเภทหนึ่ง (Every Object has a Type)

จ. วัตถุที่จัดอยู่ในประเภทเดียวกันย่อมได้รับข่าวสารเสมือนกัน (All Object of a Particular Type Can Receive the Same Messages)

**แนวคิดแบบOOP**

OOPก็คือ “ธรรมชาติของวัตถุ” หมายความว่า OOP จะมองสิ่งแต่ละสิ่งถือเป็น “วัตถุชิ้นหนึ่ง” (Object) มันจะมีสีแดงหรือสีเขียว ยาวหรือสั้น มันก็คือวัตถุชิ้นหนึ่งเหมือนกัน และเราสามารถกำหนดประเภทหรือคลาสให้กับวัตถุเหล่านั้นได้

นอกจากนี้ เมื่อ OOP มองทุกสิ่งถือเป็นวัตถุชิ้นหนึ่งแล้ว ยังสามารถคิดต่อไปอีกว่า “วัตถุแต่ละอย่างนั้น ต่างก็มีลักษณะและวิธีการใช้งานเป็นของตัวเอง” หมายความว่า วัตถุแต่ละชนิดหรือแต่ละชิ้นต่างก็มีรูปร่าง ลักษณะ และการใช้งาน (การกระทำ) ที่แตกต่างกันออกไป เราจะเรียกคุณลักษณะของวัตถุว่า แอตทริบิวต์ (Attribute) และจะเรียกวิธีการใช้งานวัตถุว่า เมธทอด (Method) ตัวอย่างเช่น “ดินสอเป็นวัตถุที่มีลักษณะเรียวยาว ภายในเป็นไส้ถ่านใช้สำหรับเขียน การใช้ดินสอทำได้โดยใช้มือจับและเขียนลงบนวัสดุรองรับ”

จากข้อความข้างต้น สามารถจับใจความได้ว่า คุณลักษณะของวัตถุ (Attribute) ก็คือ “ยาวเรียว ภายในเป็นไส้ถ่าน” ส่วนการใช้งาน (Method) ก็คือ “ใช้มือจับและเขียนลงบนวัสดุรองรับ”

จะเห็นได้ว่าแนวคิดของ OOP นั้นจะมีลักษณะที่คล้ายกับธรรมชาติของสิ่งหนึ่งซึ่งสามารถแบ่งแยกสิ่งต่าง ๆ ออกเป็นแต่ละประเภทได้ ถ้านำเอาแนวคิดของ OOP มาใช้ในการเขียนโปรแกรมและการจัดการข้อมูล จะพบว่าโปรแกรมหรือฟังก์ชันจะมีความเป็นอิสระแก่กันอย่างเห็นได้ชัด ก็คือ โปรแกรมหรือฟังก์ชันแต่ละตัวถึงแม้จะมาจากที่เดียวกันแต่สามารถทำงานในคนละหน้าที่ เก็บข้อมูลคนละค่าได้ โดยจะไม่มายุ่งเกี่ยวกันแต่อย่างใด

**ข้อความรู้เกี่ยวกับOOP**

เนื่องจากหลักการเขียนโปรแกรมเชิงวัตถุเป็นแนวคิดแบบใหม่ ดังนั้น การทำงานหลาย ๆ ส่วนของการเขียนโปรแกรมแบบนี้อาจจะยังไม่เป็นที่คุ้นเคยมากนัก จึงจำเป็นที่ผู้ศึกษาต้องทำความเข้าใจการทำงานของแนวคิดนี้ได้ดียิ่งขึ้น ข้อควรรู้เบื้องต้นเกี่ยวกับ OOP มีดังนี้

**การเชื่อมต่อ (Interface)**

อินเตอร์เฟส (Interface) หมายถึง การเชื่อมต่อ ถ้าเป็นการเชื่อมต่อระหว่างผู้ใช้กับคอมพิวเตอร์ จะเรียกการเชื่อมต่อนั้นว่า ยูสเซอร์อินเตอร์เฟส (User Interface) แต่ในการเขียนโปรแกรมเชิงวัตถุ การเชื่อมต่อยังรวมไปถึงวัตถุ (Object) เพราะในวัตถุจะต้องมีอินเตอร์เฟส  
อันเป็นส่วนที่วัตถุนั้น ๆ จะให้บริการหรือเป็นส่วนที่บอกว่าวัตถุนั้น ๆ สามารถทำอะไรได้บ้าง ซึ่งบางครั้งเรียกว่า เมธทอด (Method)

ข้อดีของการมีอินเตอร์เฟส ก็คือ การเปลี่ยนแปลงที่เกิดขึ้นภายในวัตถุจะไม่กระทบต่ออินเตอร์เฟส ดังนั้น ภายในวัตถุผู้เขียน คำสั่งสามารถดัดแปลง แก้ไข หรือเพิ่มเติมได้ตลอดเวลา นอกจากนี้ ภายในวัตถุยังสามารถเก็บค่าต่าง ๆได้อีกด้วย

**การซ่อนรายละเอียด (Encapsulation)**

ส่วนประกอบของวัตถุตามแนวความคิดการเขียนโปรแกรมเชิงวัตถุ จะต้องประกอบไปด้วยสองส่วนเป็นอย่างน้อย คือ ส่วนของคุณสมบัติใช้เก็บข้อมูลรายละเอียด สถานะ โอยใช้ตัวแปรเก็บค่าต่าง ๆ ไว้ และส่วนของเมธทอดทีเป็นตัวเชื่อมการทำงานของวัตถุนั้น ๆ โดยผู้ใช้จะไม่สามารถติดต่อใช้งานกับตัวแปรที่อยู่ข้างในได้ ในภาษา C++ จะใช้คำ Public,Private และ Protected เข้ามาช่วยกำหนดขอบเขตการใช้

**การนำวัตถุมาใช้ใหม่ (Reuse the Object)**

จุดประสงค์ใหญ่ของการเขียนโปรแกรมเชิงวัตถุ ก็คือ การนำส่วนต่าง ๆ ของวัตถุที่สร้างขึ้นกลับมาใช้ใหม่หรือที่เรียกในภาษาอังกฤษว่า “Reuse” เมื่อผู้เขียนโปรแกรมสร้างวัตถุมีจำนวนมากพอก็สามารถนำวัตถุที่สร้างขึ้นมาประกอบเป็นวัตถุใหม่ หรือที่เรียกว่าคอมโพสิตชั่น “Composition”

นอกจากวิธีการคอมโพสิตแล้ว ผู้ใช้ยังสามารถ Reuse ส่วนของวัตถุโดยการใช้การสืบทอดคุณสมบัติ (Inheritance) จากคลาส ลักษณะเช่นนี้ คือ เป็นการนำส่วนของวัตถุทั้งหมดมาใช้ ซึ่งปกติแล้ววัตถุที่นำมาใช้ในลักษณะนี้จะมีขนาดใหญ่ ถ้าเป็นการคอมโพสิตจะประกอบขึ้นจากส่วนของวัตถุที่มีขนาดเล็กกว่า อย่างไรก็ตาม ขนาดของวัตถุมิได้เป็นตัวกำหนดที่แน่นอนตายตัวเสมอไป

**การพ้องรูป (Polymorphism)**

รากฐานของการพ้องรูป (Polymorphism) ก็คือ การถ่ายทอดคุณสมบัติ เพราะถ้าไม่มีการถ่ายทอดคุณสมบัติก็จะไม่เกิดสภาวะการพ้องรูป การถ่ายทอดคุณสมบัติเป็นเครื่องมือยืนยันได้ว่าคลาสลูกที่เกิดจากคลาสแม่เดียวกันย่อมมีคุณสมบัติเหมือนกัน

**การเขียนโปรแกรมและการออกแบบระบบงาน**

ก่อนที่ผู้เขียนโปรแกรมจะสามารถเขียนคำสั่งได้ จะต้องมีการออกแบบระบบงานก่อนแล้วจึงเขียนโปรแกรมเป็นภาษาต่าง ๆ ตามชนิดของงานและความเหมาะสม การเขียนโปรแกรมเชิงวัตถุก็เช่นกัน ตะต้องมีการออกแบบระบบงานก่อน หลักสำคัญสำหรับการออกแบบเชิงวัตถุ ก็คือ การหาวัตถุให้พบ เมื่อพบแล้วจะต้องจำแนกวัตถุออกเป็นส่วนที่เปลี่ยนแปลงและส่วนที่อยู่คงที่ วัตถุที่ไม่เปลี่ยนแปลงสามารถนำไปใช้ได้เมื่อมีการปรับปรุงระบบงานใหม่ นั่นคือเหตุผลที่ทำให้ต้องมีการออกแบบระบบงาน วัตถุที่มีการเปลี่ยนแปลงบ่อยก็ได้แก่ วัตถุที่ทำหน้าที่เป็นอินเตอร์เฟส  
เป็นต้น

**ประโยชน์ของการโปรแกรมแบบ OOP**

เทคโนโลยีของออบเจ็กต์โอเรียนเท็ดได้นำประโยชนาของการพัฒนาซอฟต์แวร์มาให้ ดังนี้

1. ความสามารถในการเรียกใช้ได้หลายครั้ง ออบเจ็กต์ได้ถูกออกแบบตามหลักการที่ว่าสามารถเรียกใช้งานได้หลาย ๆ ครั้ง ในหลักการนี้ทำให้ Application ของ OOP ตัวแรกอาจจะทำได้ยาก แต่ว่าโปรแกรมแอปพลิเคชันที่เขียนภายหลังจะสร้างง่ายเพราะสามารถเรียกใช้ออบเจ็กต์ที่ถูกสร้างไว้ตั้งแต่โครงงานแรกได้
2. ความเชื่อถือได้ โปรแกรมแอปพลิเคชันของ OOP จะมีความเชื่อถือได้สูงเพราะจะรวมเอาส่วนย่อยที่ทดสอบจนได้มาตรฐานแล้วมารวมเข้าไว้ด้วยกัน รหัส (Code) ที่เขียนขึ้นมาใหม่ในแต่ละแอพลิเคชั่นจะมีไม่มากนัก เนื่องจากรหัสส่วนใหญ่จะถูกดึงมาจากไลบรารีที่มีความเชื่อถือได้สูงอยู่แล้ว และในการโปรแกรมภาษา C++ ยังมีคุณประโยชน์อื่นอีก
3. ความต่อเนื่องกัน การพัฒนาซอฟต์แวร์แบบ OOP ใ จะเปลี่ยนไปตามฝีมือและจำนวนนักเขียนโปรแกรมนักโปรแกรมที่ชำนาญสามารถเรียนรู้หลักการของ OOP ได้ภายในเวลาไม่นาน และสามารถเข้าใจเนื้อหาได้ไม่ยาก

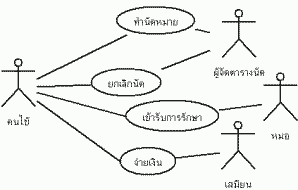
**2.6 ยูเอ็มแอลไดอะแกรม**

ยูเอ็มแอล (UML) ย่อมาจาก Unified Modeling Language เป็นภาษาที่ใช้อธิบายแบบจำลองต่าง ๆ หรือเป็นภาษาสัญลักษณ์รูปภาพมาตรฐานสำหรับใช้ในการสร้างแบบจำลองเชิงวัตถุ โดยยูเอ็มแอล เป็นภาษามาตรฐานสำหรับสร้างแบบพิมพ์เขียวให้แก่ระบบงานเราสามารถใช้ยูเอ็มแอลในการสร้างมุมมองกำหนดรายละเอียดสร้างระบบงานและจัดทำเอกสารอ้างอิงให้แก่ระบบงานได้ เนื่องจากยูเอ็มแอล เป็นภาษาที่มีการใช้สัญลักษณ์รูปภาพ จึงอาจมีผู้เข้าใจสับสนว่ายูเอ็มแอล เป็นการสร้างแผนภาพหรือเป็นเพียงการใช้สัญลักษณ์เพื่ออธิบายระบบงานเท่านั้น แต่แท้จริงแล้ว ยูเอ็มแอลมีลักษณะของแบบจำลองข้อมูล คือเป็นแบบจำลองที่เอาไว้อธิบายแบบจำลองอื่น ๆ อีกทีการใช้งานภาษายูเอ็มแอล นอกจากจะต้องเข้าใจในแนวความคิดเชิงวัตถุแล้ว ยังจำเป็นต้องมีพื้นฐานความเข้าใจเกี่ยวกับแบบจำลองภาพด้วยเช่นกัน

แบบจำลอง (Modeling) เป็นวิธีการวิเคราะห์ออกแบบ (Analysis and Design)อย่างหนึ่งที่เน้นการใช้งานแบบจำลองเป็นหลัก ซึ่งแบบจำลองที่สร้างขึ้นมาจะสามารถช่วยให้เข้าใจในปัญหาได้ง่ายขึ้น อีกทั้งยังสามารถนำแบบจำลองมาเป็นเครื่องมือในการสื่อสารถ่ายทอดความคิดกับบุคคลอื่น ๆ ที่เกี่ยวข้องในโครงการได้ เช่นลูกค้านักวิเคราะห์ระบบ นักออกแบบระบบ เป็นต้น ส่วนแบบจำลองภาพ คือการใช้สัญลักษณ์รูปภาพในการสร้างแบบจำลองของระบบที่จะพัฒนาเพื่อประโยชน์ที่คล้ายคลึงกันในการทำความเข้าใจกับความต้องการของลูกค้าการออกแบบระบบที่เป็นไปได้อย่างชัดเจนขึ้นและการบำรุงรักษาที่ง่ายยิ่งขึ้น แบบจำลองเกิดขึ้นโดยการนำเสนอส่วนต่าง ๆ ของระบบแต่เพียงส่วนที่สำคัญโดยไม่คำนึงถึงรายละเอียดปลีกย่อยต่าง ๆ ในการพัฒนาระบบซอฟต์แวร์ที่ซับซ้อน นักพัฒนาจำเป็นต้องทำความเข้าใจกับมุมมองด้านต่าง ๆ ของระบบก่อนทำการพัฒนาจริง โดยการสร้างแบบจำลองอันเปรียบเสมือนพิมพ์เขียวที่แสดงถึงภาพรวมทั้งหมดของระบบแบบจำลองที่สร้างขึ้นจะต้องมีความสอดคล้องกับความต้องการของผู้ใช้งานระบบเป็นสำคัญ ในส่วนของรายละเอียดต่าง ๆ จะค่อย ๆ ถูกเพิ่มเติมลงไปในตัวแบบจำลอง และในที่สุดแบบจำลองจะถูกนำไปพัฒนาขึ้นเป็นระบบจริง

ใน UML ประกอบด้วยแผนภาพดังนี้

1. Use case Diagram เป็น แผนภาพ เพื่อใช้อธิบายหน้าที่ของระบบใหม่ หรือระบบปัจจุบันว่ามีการกระทำอะไรกับระบบงานบ้าง และใครเป็นผู้กระทำ และมีความเกี่ยวข้องกับระบบงานอื่นๆ อย่างไร



ภาพที่ 1-1 Use case Diagram

1. Class Diagram ประกอยด้วย Class และความสัมพันธ์ระหว่าง Class เช่น Dependency, Generalization, Association เป็นต้น Class Diagram สามารถแสดงรายละเอียดว่ามี Method และ Attribute อย่างไร
2. Object Diagram ประกอบด้วย Object และ Relation ระหว่าง Object โดยแต่ละ Object จะแสดง Instance ของแต่ละ class ที่มีในระบบ และความสัมพันธ์ระหว่าง Class เช่น Dependency, Generalization หรือ Association ซึ่งมีลักษณะเช่นเดียวกับ Class Diagram
3. Sequence Diagram จะแสดงลำดับการทำงานของระบบ โดยมี Object และ เวลาเป็นตัวกำหนดลำดับของงาน และเน้นไปที่ instant ของ Object Sequence Diagram เป็น Diagram ซึ่งแสดงปฏิสัมพันธ์(Interaction) ระหว่าง Object ตามลำดับของเหตุการณ์ที่เกิดขึ้น ณ เวลาที่กำหนด message ที่เกิดขึ้นระหว่าง class จะสามารถนำไปสู่การสร้าง method ใน class ที่เกี่ยวข้องได้ จากตัวอย่าง Sequence ข้างต้นจะทำการ Design สำหรับการทำรายการบัญชีการเงินของธนาคาร โดยจะเริ่มจากหน้าจอ JSP ซึ่งถือเป็น User Interface(UI) ประเภทหนึ่ง จะทำหน้าที่ติดต่อกับผู้ใช้ เพื่อรับข้อมูลจากการใช้งานโดย User จากนั้น เมื่อกดปุ่มส่งคำสั่งใดๆ ผ่านหน้าจอ JSP ก็จะทำการเรียกใช้ Servlet โดยเรียกผ่าน Method doPost ของ Account Servlet และทำการส่งค่าและเรียกใช้ Class ต่างๆ ที่เกี่ยวข้องเพื่อทำการประมวลผลตาม business process ต่อไป Servlet และ Class ที่ทำการประมวลผลต่างๆ จึงเป็นตัวควบคุมการทำงาน หรือ Controller นั่นเอง ส่วน Class ที่ทำหน้าที่เก็บข้อมูลต่างๆ จะเรียกว่า Entity Class
4. Collaboration Diagram แสดงลำดับการทำงานของ วัตถุ ผู้เกี่ยวข้อง และกิจกรรม โดยลำดับการทำงานไม่ขึ้นกับเวลา เพราะการแสดงความสัมพันธ์ของ Object กับเวลาเป็นหน้าที่ของ Sequence Diagram
5. State Diagram ประกอบด้วย State ต่างๆ ของ Object และเหตุการณ์ต่างๆ ที่ทำให้สถานะของ Object เปลี่ยนและการกระทำที่เกิดขึ้นเมื่อสถานะของระบบเปลี่ยนไป สามารถบอกสถานะของ Object ได้ โดยจะให้ความสนใจว่า ณ เวลาใดๆ Object นั้นมี status เป็นแบบใด
6. Activity Diagram Activity Diagram แสดงลำดับ กิจกรรมของการทำงาน(Work Flow) สามารถแสดงทางเลือกที่เกิดขึ้นได้ Activity Diagram จะแสดงขั้นตอนการทำงานในการปฏิบัติการ โดยประกอบไปด้วยสถานะต่างๆ ที่เกิดขึ้นระหว่างการทำงาน และผลจากการทำงานในขั้นตอนต่าง ๆ
7. Component Diagram เป็น Diagram ซึ่งแสดงโครงสร้างทางกายภาพของ Software โดยจะประกอบด้วยองค์ประกอบซึ่งอยู่ในรูปต่างๆ เช่น Binary, text และ executable ภายใน Component Diagram ก็จะมีความสัมพันธ์แสดงอยู่เช่นเดียวกับ Class Diagram, Object Diagram
8. Deployment Diagram เป็นสิ่งที่สามารถทำการแสดงระบบสถาปัตยกรรมของ Hardware/Software ตลอดจนความสัมพันธ์ระหว่าง hardware/software

**2.7 เว็บเซอร์วิส**

เว็บเซอร์วิส (Web service) คือระบบซอฟต์แวร์ที่ออกแบบมา เพื่อสนับสนุนการแลกเปลี่ยนข้อมูลกัน ระหว่างเครื่องคอมพิวเตอร์ผ่านระบบเครือข่าย โดยที่ภาษาที่ใช้ในการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ คือเอกซ์เอ็มแอล เว็บเซอร์วิสมีอินเทอร์เฟส ที่ใช้อธิบายรูปแบบข้อมูลที่เครื่องคอมพิวเตอร์ประมวลผลได้ เช่น WSDL ระบบคอมพิวเตอร์ใช้งานสื่อสารโต้ตอบกับเว็บเซอร์วิสตามรูปแบบที่ได้กำหนดไว้แล้ว โดยการส่งสาสน์ตามอินเตอร์เฟสของเว็บเซอร์วิสนั้น โดยที่สาสน์ดังกล่าวอาจแนบไว้ในซอง SOAP หรือส่งตามอินเตอร์เฟสในแนวทางของREST สาสน์เหล่านี้ปกติแล้วถูกส่งโดยอาศัย HTTP และใช้ XML ร่วมกับมาตรฐานเกี่ยวกับเว็บอื่นๆ โปรแกรมประยุกต์ที่เขียนโดยภาษาต่างๆ และทำงานบนแพลตฟอร์มต่างๆกันสามารถใช้เว็บเซอร์วิสเพื่อแลกเปลี่ยนข้อมูลผ่านทางเครือข่ายคอมพิวเตอร์ เช่น อินเทอร์เน็ต ในลักษณะเดียวกับการสื่อสารระหว่างโปรเซส (Inter-process communication) บนเครื่องเดียวกัน ความสามารถในการแลกเปลี่ยนข้อมูลระหว่างระบบที่ต่างกันนี้ (เช่น การแลกเปลี่ยนข้อมูลระหว่าง โปรแกรมที่เขียนโดยภาษาจาวา และโปรแกรมที่เขียนโดยภาษาไพทอน หรือการแลกเปลี่ยนข้อมูลระหว่างโปรแกรมประยุกต์ที่ทำงานบนไมโครซอฟท์วินโดวส์และโปรแกรมประยุกต์ที่ทำงานบนลินุกซ์) เกิดขึ้นได้เนื่องจากการใช้มาตรฐานเปิด โดย OASIS และ W3Cเป็นคณะกรรมการหลักในการรับผิดชอบมาตรฐานและสถาปัตยกรรมของเว็บเซอร์วิส

**ข้อกำหนดซอฟต์แวร์และมาตรฐานอื่นที่เชื่อมโยงกัน**

**SOAP**

เว็บเซอร์วิสเป็นลักษณะในรูปแบบของการออกแบบโมเดลสื่อสาร ในลักษณะของการกระจาย,การติดต่อสื่อสารที่เป็นตัวกลาง โดยโพรโทคอลที่ใช้ในการสื่อสารคือ SOAP (Simple Object Access Protocol) เป็นโพรโทคอลในการติดต่อแลกเปลี่ยนข้อมูลระหว่างกันเป็นโพรโทคอลการสื่อสารในระดับ Application Layer หรือในระดับ แอปพลิเคชันโดยอาศัยผ่านอินเทอร์เน็ตโพรโทคอล อย่างเช่น HTTP, SMTP โพรโทคอลพัฒนารากฐานมาจาก XML โดยมาตรฐานของ SOAP ปัจจุบันอยู่เวอร์ชัน 1.2 เอกสารสามารถดูได้ที่ W3C

**UDDI (Universal Description, Discovery and Integration)**

เป็นเหมือนตัวไดเรกทอรี่ที่เก็บหรือลงทะเบียนเว็บเซอร์วิส UDDI หรือ Universal Description,Discovery and Integration (UDDI) เป็นข้อกำหนดอันเกี่ยวกับระบบบริการลงทะเบียน (registry service) สำหรับเว็บเซอร์วิส และสำหรับบริการอื่นๆ ทั้งที่ไม่ใช่แบบอิเล็กทรอนิกส์และแบบอิเล็กทรอนิกส์ ตัว บริการลงทะเบียน UDDI คือเวบเซอร์วิสซึ่งจัดการข้อมูลเกี่ยวกับผู้ให้บริการด้านต่างๆ หรือแม้แต่กระทั่งให้บริการข้อมูล บรรดาผู้ให้บริการสามารถใช้ UDDI ในการประกาศว่า บริการใดๆบ้างที่ให้บริการ และ client สามารถใช้บริการของ UDDI ในการค้นหาบริการที่ตนต้องการได้ตรงตามความต้องการของตนได้

ข้อกำหนด UDDI ได้มีการนิยามตามนี้

1. SOAP API (Simple Object Access Protocol) Application Programming Interface) ซึ่งตัวแอปพลิเคชันจะใช้ในการสอบถามและประกาศข้อมูลไปยังระบบลงทะเบียน UDDI
2. XML Schema คือโครงสร้างรูปแบบของระบบลงทะเบียนและ รูปแบบของข้อความ SOAP (SOAP Message format)
3. WSDL คือข้อกำหนดของ SOAP APIs
4. ข้อกำหนด UDDI รีจิสทรี (รูปแบบทางเทคนิค t-models) ของข้อกำหนดหลายๆ อย่างและหมวดหมู่ของระบบซึ่งอาจใช้ในการกำหนดและจัดแบ่งหมวดหมู่การลงทะเบียน UDDI

**WSDL**

ย่อมาจาก Web Services Description Language เป็นภาษาที่อธิบายเว็บเซอร์วิส ซึ่งทำให้ผู้เรียกใช้เว็บเซอร์วิสหรือโปรแกรมที่ต้องการเรียกใช้เว็บเซอร์วิสทราบว่ามีโอเปอเรชั่นอะไรให้บริการ และในแต่ละโอเปอเรชั่นจะต้องส่งข้อมูลอะไรไปและจะได้รับข้อมูลแบบใดกลับมา ตลอดจนทราบอินเทอร์เน็ตโพรโทคอลที่จะต้องใช้ในการติดต่อเว็บเซอร์วิสและที่อยู่ของเว็บเซอร์วิส

**XLANG**

สนับสนุนการติดตามข้อมูล (transaction) เหมาะสำหรับการสื่อสารที่สามารถติดตามข้อมูลใน web ขนาดใหญ่ รวมไปถึง การสื่อสารคล้ายๆ ลักษณะชุมทางหรือ webservice หลายๆ เวบเซอร์วิส

**XKMS (XML Key Management Specification)**

เป็นการทำงานที่เกิดจากความร่วมมือของ Microsoft และ Verisign ซึ่งสำหรับการตรวจสอบสิทธิและการลงทะเบียน

**2.8 การประมวลผลภาพดิจิตอล**

การประมวลผลภาพดิจิตอล (Digital image processing) เป็นขั้นตอนที่สำคัญในการทำงานกับข้อมูลรูปภาพ ซึ่งใช้อย่างแพร่หลายในงานหลายประเภท เช่น การปรับปรุงภาพ การค้นหาวัตถุลักษณะที่ต้องการ การนับจำนวนวัตถุ เป็นต้น

งานวิจัยนี้จะศึกษาขั้นตอนวิธีที่เป็นเครื่องมือพื้นฐานการประมวลผลภาพดิจิตอล ซึ่ง สามารถนำไปประยุกต์ใช้กับงานที่ซับซ้อนยิ่งขึ้น ได้แก่ การแปลงแบบฮิตออร์มิส การจับคู่โดยใช้ สหสัมพันธ์ และการหาขอบโดยวิธีแคนนี เพื่อนำไปปรับปรุงประสิทธิภาพด้วยหน่วยประมวลผลกราฟิกส์

**2.8.1 การแปลงแบบฮิตออร์มิส (Hit-or-miss transformation: HMT) [15]**

การแปลงแบบฮิตออร์มิส เป็นวิธีการประมวลผลภาพด้วยรูปร่างลักษณะ (Morphological image processing) วิธีการหนึ่ง โดยมีจุดประสงค์เพื่อระบุตำแหน่งของวัตถุที่ต้องการในรูปภาพ

การแปลงแบบฮิตออร์มิสในงานวิจัยนี้จะใช้กับภาพขาวดำ (Binary image) ซึ่งแสดงในพื้นที่ระบบพิกัดเลขจำนวนเต็มสองมิติที่เขียนแทนด้วย Z2 และสามารถอธิบายการทำงานได้ดว้ยทฤษฎีเซต เหมือนกับวิธีการประมวลผลภาพด้วยรูปร่างลักษณะอื่นๆ

สำหรับการแปลงแบบฮิตออร์มิสของ A โดย B หรือ (A⊗ B) มีตัวอย่างการทำงานดัง

ภาพที่2-1 และการทำงานแสดงดังสมการที่ (2.1)

A⊗B = (AΘX ) ∩ [ACΘ(W − X )]

สมการที่ (2.1)

โดยที่ A คือ เซตของทุกจุดบนรูปภาพขาวดำตั้งต้น

B คือ ส่วนประกอบโครงสร้าง (Structuring element) ของวัตถุที่ต้องการ

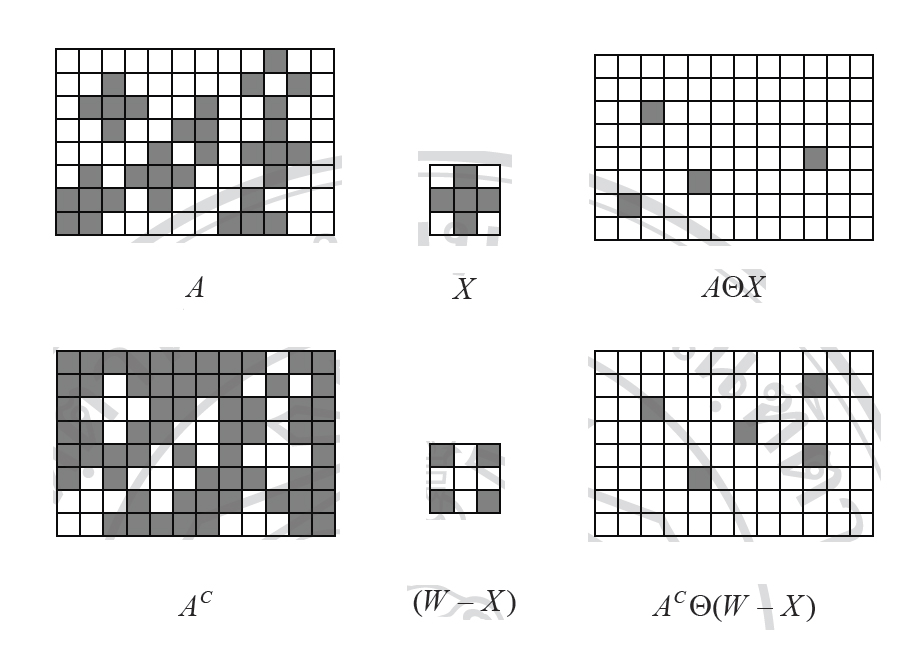
AC คือ ส่วนเติมเต็ม (Complement) ของเซต A

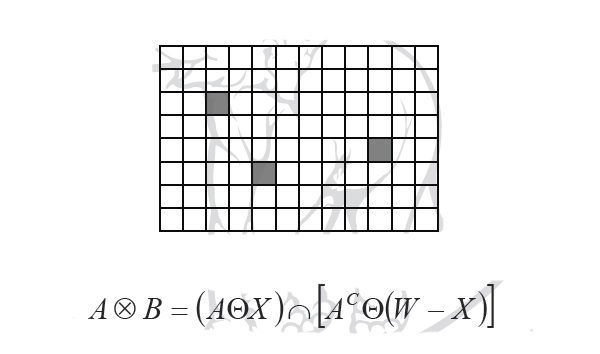
X คือ ส่วนที่เป็นวัตถุในเซต B

(W - X ) คือ ส่วนที่เปน็ ฉากหลังในเซต B

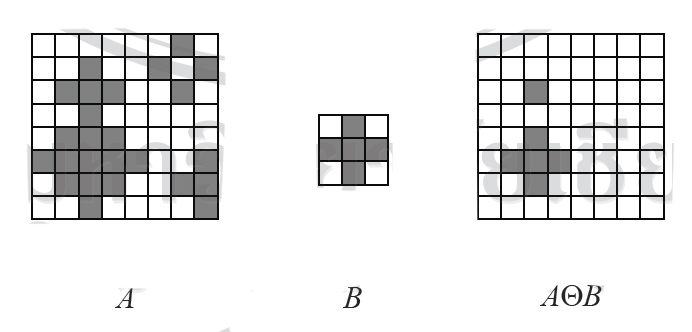
AΘB คือ การดำเนินการอีโรชัน (Erosion) ของ A โดย B โดยที่

AΘB = {z : B + z ⊆A} ตัวอย่างของการดำเนินการอีโรชัน แสดงใน ภาพที่ 1-2





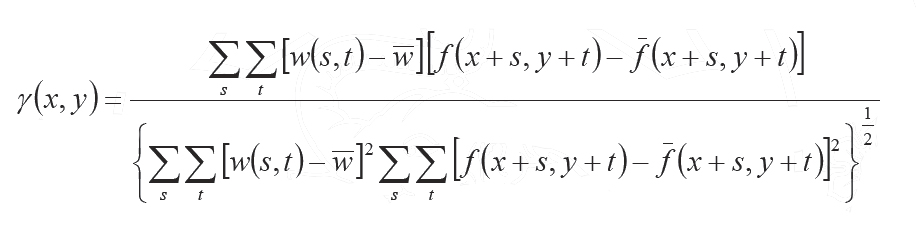
ภาพที่ 1-2 ตัวอย่างการทำงานของการแปลงแบบฮิตออร์มิส



ภาพที่ 1-3 ตัวอย่างการทำงานของการดำเนินการอีโรชัน

**2.8.2 การจับคู่โดยใช้สหสัมพันธ์ (Matching by correlation) [15]**

การต่อภาพพาโนรามา (Panorama image) คือการนำภาพมาต่อกันให้เป็นภาพที่ใหญ่ขึ้น ซึ่งเป็นวิธีการที่นิยมใช้สำหรับการเพิ่มความละเอียดของรูปภาพ ขั้นตอนหนึ่งที่สำคัญในการต่อภาพพาโนรามาคือ การจับคู่รูปแบบของภาพ (Template matching) หรือสหสัมพันธ์ (Correlation) เป็นกระบวนการที่ใช้ในการหาความสัมพันธ์ระหว่างภาพกับรูปแบบที่ต้องการด้วย ค่าระดับสีเทาของภาพและรูปแบบ ตำแหน่งพิกัดที่รูปแบบสามารถจับคู่กับภาพได้พอดีจะมีค่า สัมประสิทธิ์สหสัมพันธ์ (Correlation coefficient) มากที่สุด ซึ่งค่าสัมประสิทธิ์สหสัมพันธ์ สามารถคำนวณได้ดังสมการที่ (2.2)



สมการที่ (2.2)

โดยที่ y(x, y) คือ ค่าสัมประสิทธิ์สหสัมพันธ์ที่ตำแหนง่ พิกัด (x, y)

w คือ หน้าต่างรูปแบบที่ต้องการจับคู่ขนาด m× n

f คือ ภาพตั้งต้น

(s,t) คือ พิกัดในการเลื่อนตำแหนง่ ในหน้าต่างและภาพตั้งตน้

w คือ ค่าเฉลี่ยระดับสีเทาในหน้าต่าง w

f คือ ค่าเฉลี่ยระดับสีเทาใน f ที่กำลังเปรียบเทียบกับหน้าต่าง w

**2.8.3 การหาขอบโดยวิธีแคนนี (Canny edge detection) [15]**

การหาขอบโดยวิธีแคนนี เป็นขั้นตอนวิธีที่ใช้สำหรับหาขอบของวัตถุ ซึ่งได้ประสิทธิภาพ ที่ดีได้ผลลัพธ์เป็นเส้นขอบที่มีความต่อเนื่อง ชัดเจน เกิดความผิดพลาดน้อย ทำงานโดยอาศัย สมมติฐานว่าขอบของวัตถุคือบริเวณที่ค่าระดับสีเทามีการเปลี่ยนแปลงในทันที มีขั้นตอนการ

ทำงาน 4 ขั้นตอน ได้แก่

2.8.3.1 การทำให้ภาพเรียบ (Image smoothing) โดยใช้ตัวกรองเกาส์เซียน (Gaussian filter)

2.8.3.2 การคำนวณตัวดำเนินการเกรเดียนต์ (Gradient operator)

2.8.3.3 การกำจัดพิกเซลที่เป็นขอบที่ไม่ใช่ค่าสูงสุด  
(Nonmaxima suppression)

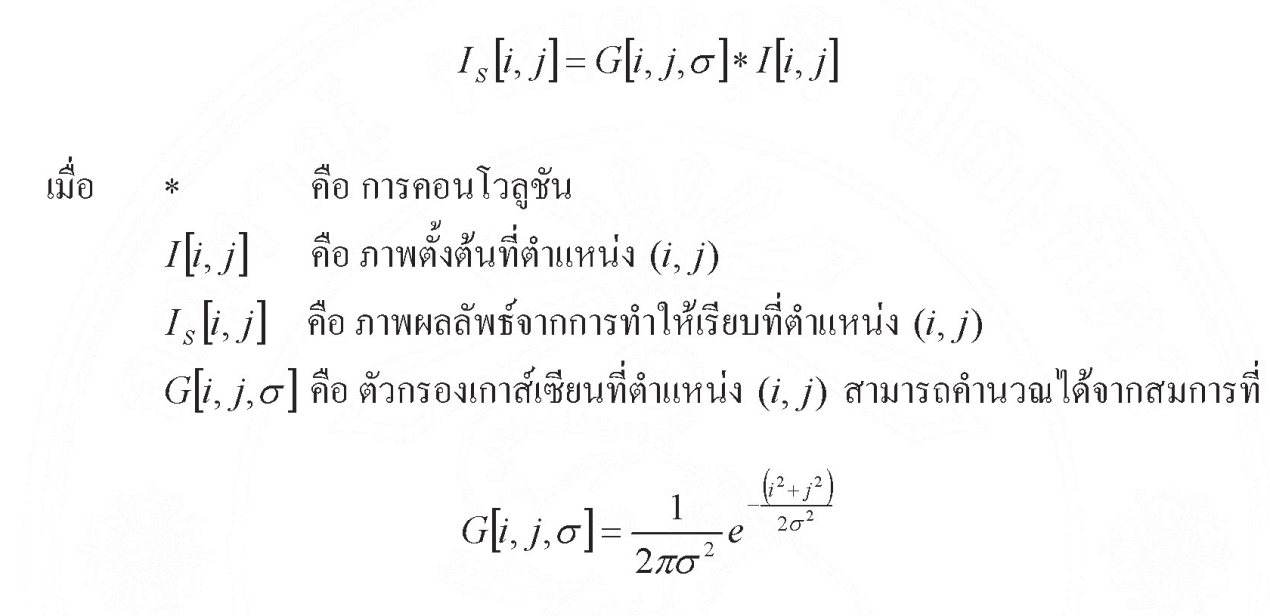
2.8.3.4 การแบ่งข้อมูลโดยใช้ขีดแบ่งสองระดับ (Double thresholding)

ในขั้นตอนแรกคือการทำให้ภาพเรียบ เพื่อลดสัญญาณรบกวนโดยใช้ตัวกรองเกาส์เซียน วิธีการนี้สามารถคำนวณได้จากผลของการคอนโวลูชัน (Convolution) ระหว่างรูปภาพหลักกับ หน้าต่างของตัวกรองเกาส์เซียน ดังสมการที่ (2.3)

สมการที่ (2.3)

สมการที่ (2.4)

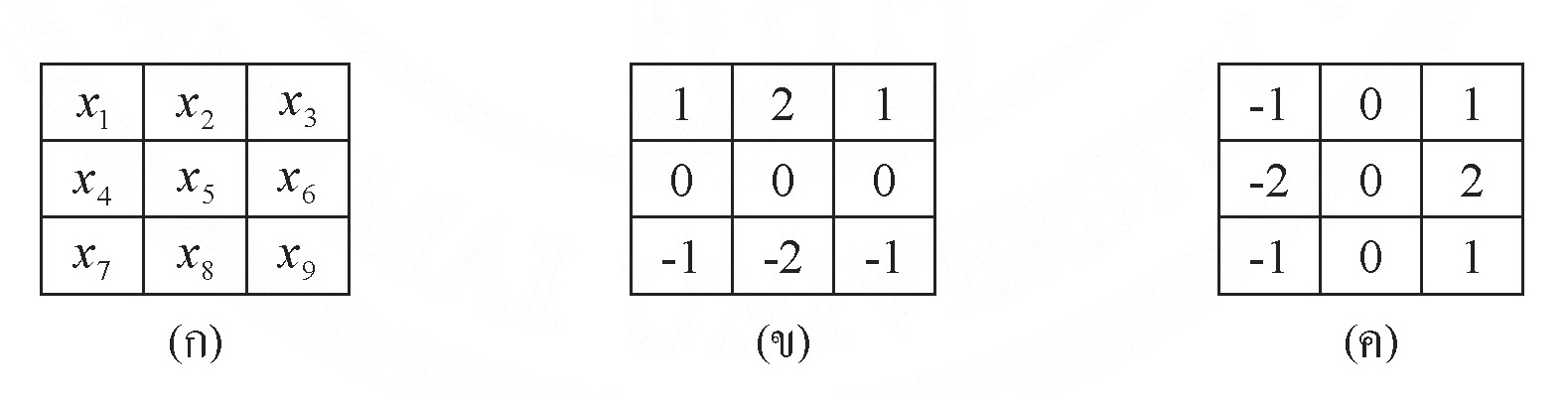
(2.4)



โดยที่ σ คือ ค่าการกระจายของเกาส์เซียน

ขั้นตอนต่อมาเป็นการหาเกรเดียนต์ของค่าระดับสีเทา ซึ่งอาศัยการหาขอบโดยวิธีโซเบล

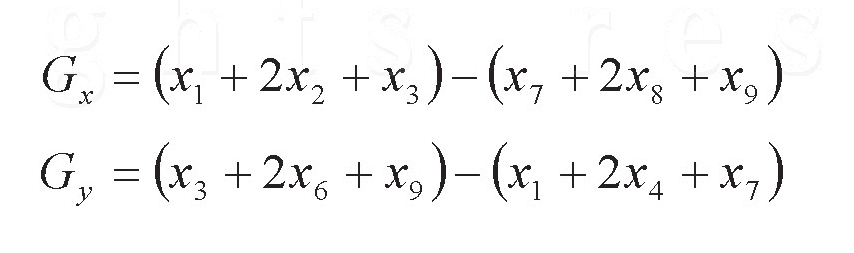
(Sobel edge detection) ที่มีลักษณะของหน้าต่างย่อย ดังภาพที่ 2-3



ภาพที่ 2.3 หน้าต่างย่อยในการหาขอบโดยวิธีโซเบล

(ก) พิกัดในหน้าต่างย่อย (ข) แนวนอน (ค) แนวตั้ง

จากรภาพที่ 2.3 สามารถคำนวณเกรเดียนต์เวกเตอร์ตามแนวนอน ( x G ) และแนวตั้ง ( y G )   
ดังสมการที่ (2.5) และ (2.6) ได้ตามลำดับ

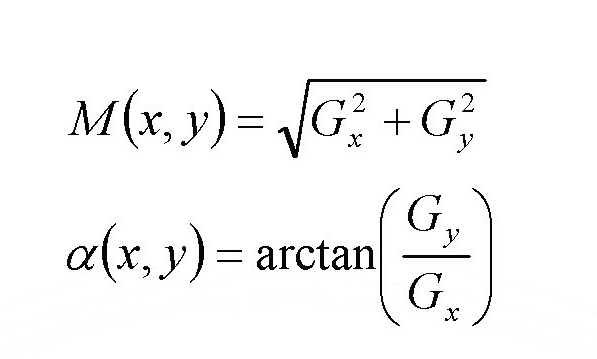


สมการที่ (2.6)

สมการที่ (2.5)

แล้วจึงนำเกรเดียนต์เวกเตอร์ทั้งแนวนอนและแนวตั้งที่ได้ ไปหาขนาดและทิศทางของเกรเดียนต์

เวกเตอร์ซึ่งคำนวณตามสมการที่ (2.7) และ (2.8) ตามลำดับ



สมการที่ (2.8)

สมการที่ (2.7)

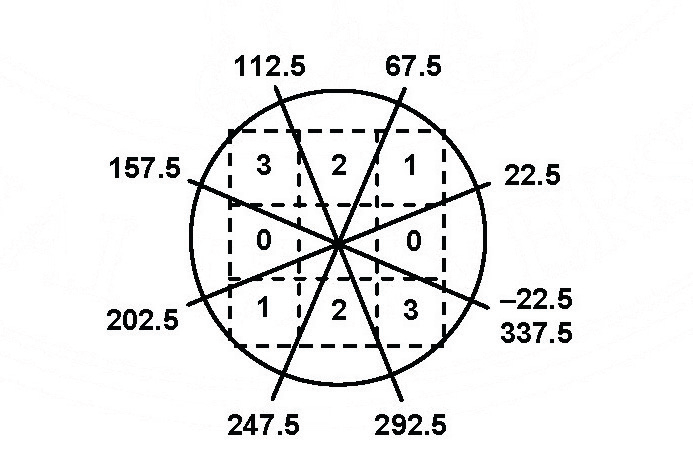
เมื่อ M(x, y) คือขนาดของเกรเดียนต์เวกเตอร์และ a(x, y) คือทิศทางของเกรเดียนต์เวกเตอร์ขั้นตอนต่อมาคือ การกำจัดพิกเซลที่เป็นขอบที่ไม่ใช่ค่าสูงสุด เป็นขั้นตอนที่เลือกเฉพาะ พิกเซลที่มีค่าขนาดของเกรเดียนต์เวกเตอร์สูงสุด โดยต้องมีทิศทางเดียวกันด้วยเท่านั้น สำหรับ ทิศทางที่พิจารณา จะถูกปรับให้ตรงกับตำแหน่งของพิกเซลในทิศทางนั้นๆ ก่อน ในกรณีหน้าต่าง ย่อยขนาด 3× 3 พิกเซลค่ามุมทิศทางที่ได้จะถูกปรับไปที่ทิศทางของพิกเซล 4 ทิศทาง ดังแสดงใน ภาพที่ 1-4 ซึ่งจะถูกพิจารณาดังนี้

ค่ามุมระหว่าง - 22.5 ถึง 22.5 และ 157.5 ถึง 202.5 จะถูกปรับเป็นทิศทาง 0

ค่ามุมระหว่าง 22.5 ถึง 67.5 และ 202.5 ถึง 247.5 จะถูกปรับเป็นทิศทาง 1

ค่ามุมระหว่าง 67.5 ถึง 112.5 และ 247.5 ถึง 292.5 จะถูกปรับเป็นทิศทาง 2

ค่ามุมระหว่าง 112.5 ถึง 157.5 และ 292.5 ถึง 337.5 จะถูกปรับเป็นทิศทาง 3

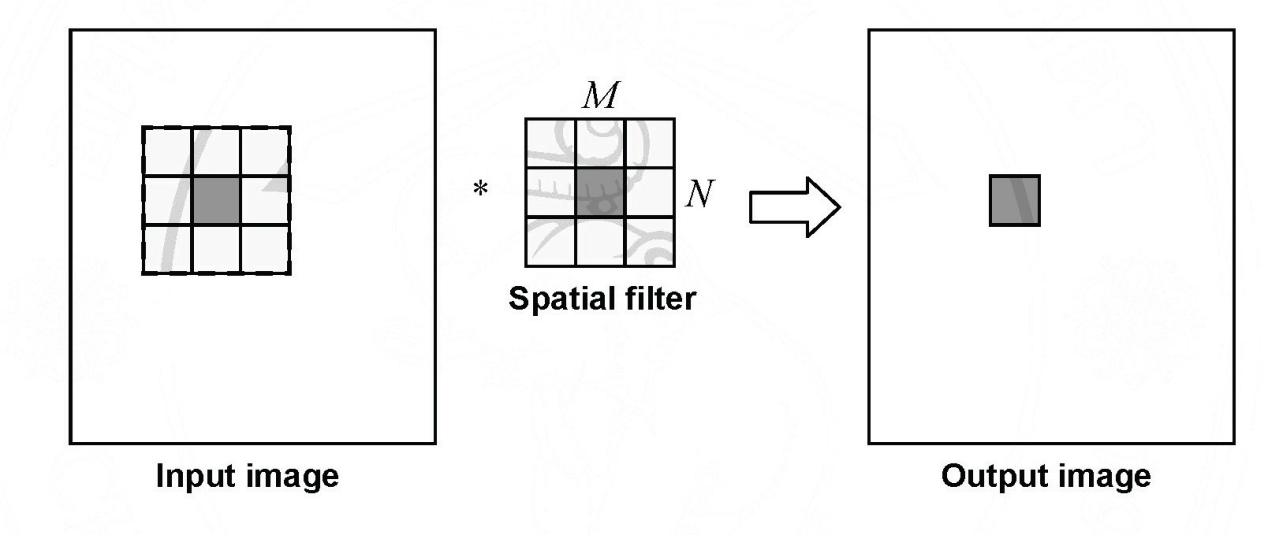
ขั้นตอนนี้เป็นขั้นตอนที่จะช่วยกำจัดส่วนที่เป็นสัญญาณรบกวนที่ยังคงเหลือ และส่วนที่เป็นพื้นผิวออกไป

ภาพที่ 1-4 ทิศทางของเกรเดียนต์เวกเตอร์

หลังจากที่ได้จุดของภาพขอบแล้ว ก็จะมาทำขั้นตอนสุดท้าย คือการแบ่งข้อมูลเพื่อเป็นการ ตัดสินใจว่า ส่วนใดบ้างคือขอบที่แท้จริงของวัตถุ โดยการแบ่งข้อมูลจะใช้ค่าขีดแบ่งสองค่า เนื่องจากภาพผลลัพธ์ก่อนที่จะมาถึงขั้นตอนนี้อาจยังมีเส้นขอบอื่นๆ ที่เกิดจากลวดลายภายในภาพ ซึ่งไม่ใช่ขอบที่แท้จริง โดยกำหนดให้ค่าขีดแบ่งทั้งสองคือ ขีดแบ่งระดับบน ( T1 ) และ ขีดแบ่ง ระดับล่าง ( T2 ) จากนั้นพิจารณาค่าของแต่ละจุด ถ้ามีค่ามากกว่า T1 จะถูกปรับเป็น 1 ซึ่งหมายความ ว่าพิกเซลนั้นเป็นขอบ ถ้ามีค่าน้อยกว่า T2 จะถูกปรับค่าเป็น 0 หมายความว่าพิกเซลนั้นไม่ใช่ขอบ ส่วนพิกเซลที่มีค่ามากกว่า T2 แต่น้อยกว่า T1 นั้นจะถูกนำมาพิจารณาพิกเซลใกล้เคียงว่าเป็นขอบหรือไม่ ถ้าเป็นก็จะปรับค่าเป็น 1 โดยภาพผลลัพธ์สุดท้าย จะสามารถบ่งบอกได้ว่า จุดพิกเซลที่เป็นขอบจะมีค่า 1 ส่วนจุดอื่นๆ จะมีค่า 0

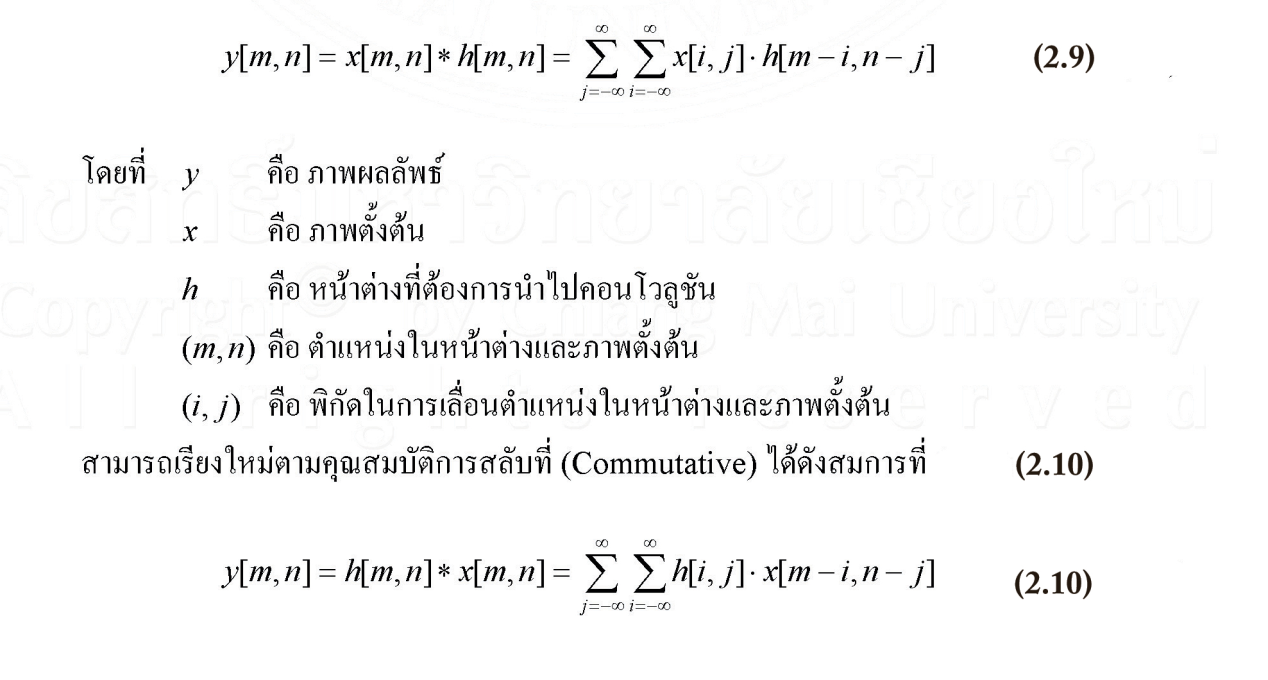
**2.8.4 การคอนโวลูชันแบบแยก (Separable convolution) [16]**

การคอนโวลูชันสำหรับการประมวลผลภาพดิจิตอล เป็นการคอนโวลูชัน 2 มิติ ซึ่ง ประกอบด้วยแนวนอนและแนวตั้ง วิธีการหนึ่งที่นิยมใช้ในการเพิ่มความเร็วในการทำงานของการ คอนโวลูชัน 2 มิติ คือ การคอนโวลูชันแบบแยก



ภาพที่ 1-5 การคอนโวลูชัน 2 มิติ

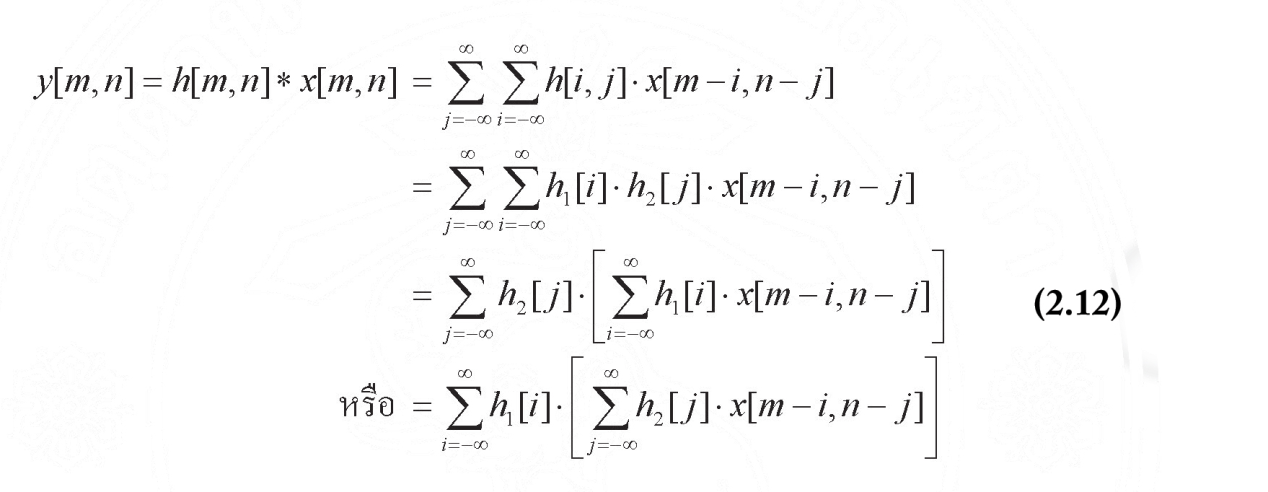
จากตัวอย่างใน ภาพที่ 1-5 แสดงการคอนโวลูชัน 2 มิติของภาพตั้งต้นกับหน้าต่างขนาด  
M × N แต่ละพิกเซลในรูปภาพหลักจะมีการคำนวณ M × N ครั้ง ดังสมการที่ (2.9)



เมื่อ h แยกไปเป็นหน้าต่างขนาด M ×1 และ 1× N ตามแกนแนวนอน (X-Pass) และแนวตั้ง (Y-Pass) ดังสมการที่ (2.11)

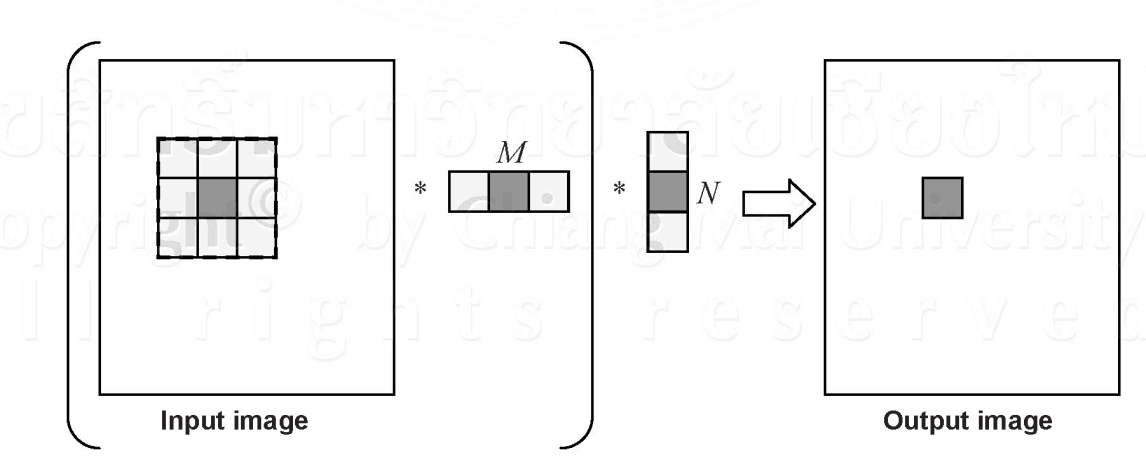
h[ m ,n] = h1[m] . h2[n] สมการที่ (2.11)

นำกลับไปแทนที่ในสมการที่ (2.10) จะได้

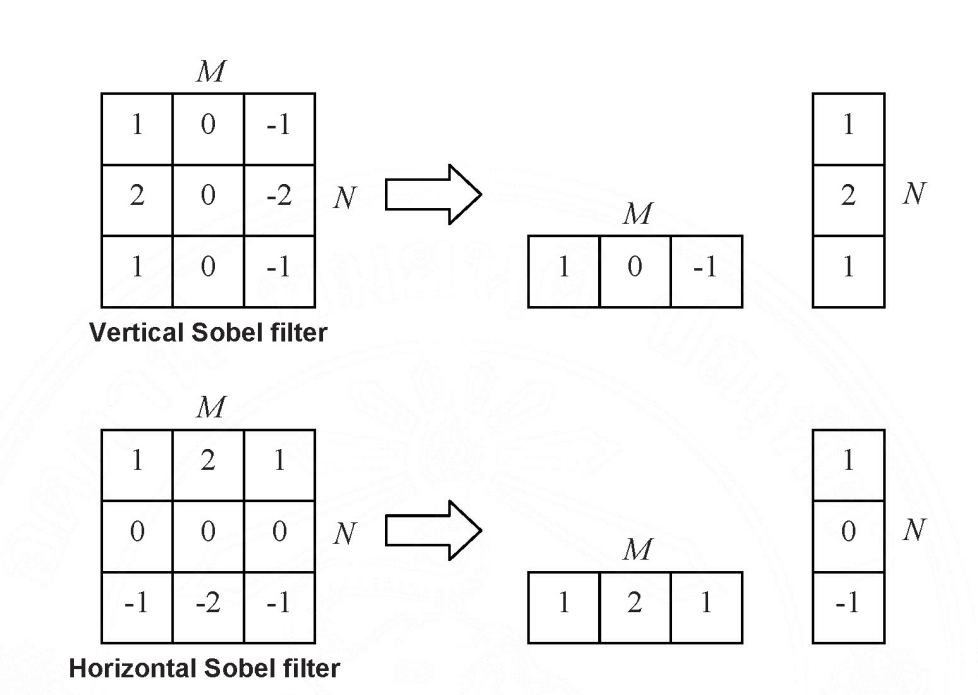


เมื่อพิจารณาสมการที่ (2.12) จะพบว่า เป็นการคอนโวลูชัน 1 มิติ จำนวน 2 ครั้ง แสดงว่า การคอนโวลูชันแบบแยกให้ผลลัพธ์สุดท้ายของการคอนโวลูชันเหมือนกัน อีกทั้งยังสามารถลดการ คำนวณในแต่ละพิกเซลลงไปเป็น M และ N ครั้งตามแกน จึงสรุปได้ว่าเมื่อเปลี่ยนการคอนโวลูชัน 2 มิติ เป็นการคอนโวลูชันแบบแยก แต่ละพิกเซลมีการคำนวณลดลงจาก M × N ครั้งเป็น

M + N ครั้ง ดังตัวอย่างใน ภาพที่ 1-6 ส่วนลำดับของแกนในการคอนโวลูชันไม่มีผลต่อจำนวนครั้ง และผลลัพธ์สุดท้าย ทำให้สามารถเริ่มคอนโวลูชันด้วยแนวแกนใดก่อนก็ได้ สำหรับ ภาพที่ 2.7 เป็น ตัวกรองโซเบลแบบแยก



ภาพที่ 1-6 การคอนโวลูชัน 2 มิติ แบบแยก



ภาพที่ 1-7 หน้าต่างของตัวกรองโซเบล สำหรับการคอนโวลูชันแบบแยก