ขั้นตอนวิธีทางคอมพิวเตอร์เพื่อรับรู้วัตถุจากภาพร่างด้วยมือ: การสร้างรูปทรง 3 มิติเบื้องต้น

นายพชระ จาฏุพจน์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาสถาปัตยกรรมศาสตรมหาบัณฑิต
สาขาวิชาสถาปัตยกรรม ภาควิชาสถาปัตยกรรมศาสตร์
คณะสถาปัตยกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2546

ISBN 974-17-4471-4

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

COMPUTER ALGORITHM FOR OBJECT RECOGNITION FROM A FREEHAND SKETCH: BASIC 3D MODELING

Mr. Pojchara Jatupoj

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Architecture in Architecture

Department of Architecture
Faculty of Architecture
Chulalongkorn University

Academic Year 2003
ISBN 974-17-4471-4

หัวข้อวิทยานิพนธ์	ขั้นตอนวิธีทางคอมพิวเตอร์เพื่อการรับรู้วัตถุจากภาพร่างด้วยมือ: การสร้างรูปทรง 3 มิติเบื้องต้น
โดย	นายพชระ จาฏุพจน์
สาขาวิชา	สถาปัตยกรรม
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ กวีไกร ศรีหิรัญ
อาจารย์ที่ปรึกษาร่วม	อาจารย์ ธิดาสิริ ภัทรากาญจน์

คณะสถาปัตยกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะสถาปัตยกรรมศาสตร์
(รองศาสตราจารย์ ดร. วีระ สัจกุล)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(รองศาสตราจารย์ เลอสม สถาปิตานนท์)

..... อาจารย์ที่ปรึกษา
(ผู้ช่วยศาสตราจารย์ กวีไกร ศรีหิรัญ)

..... อาจารย์ที่ปรึกษาร่วม
(อาจารย์ ธิดาสิริ ภัทรากาญจน์)

..... กรรมการ
(อาจารย์ สุรพล พฤกษ์ไพบูลย์)

..... กรรมการ
(อาจารย์ ภิญโญ จินันทุยา)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

พชระ จาฏุพจน์ : ขั้นตอนวิธีทางคอมพิวเตอร์เพื่อการรับรู้วัตถุจากภาพร่างด้วยมือ: การสร้างรูปทรง 3 มิติเบื้องต้น. (COMPUTER ALGORITHM FOR OBJECT RECOGNITION FROM A FREEHAND SKETCH : BASIC 3D MODELING) อ. ที่ปรึกษา : ผู้ช่วยศาสตราจารย์ กวี ไกร ศรีหิรัญ, อ.ที่ปรึกษาร่วม : อาจารย์ ธิดาสิริ ภัทรากาญจน์ จำนวนหน้า 106 หน้า. ISBN 974-17-4471-4.

วัตถุประสงค์ของงานวิจัยชิ้นนี้คือ การศึกษาการร่างภาพทางสถาปัตยกรรมและการออกแบบด้วยมือเพื่อพัฒนาเป็นขั้นตอนวิธีทางคอมพิวเตอร์ให้สามารถรับรู้ และสร้างภาพวัตถุขึ้นได้ โดยการประยุกต์ใช้กับการสร้างรูปทรง 3 มิติเบื้องต้น การวิจัยนี้เป็นการรวบรวมข้อมูลที่เกี่ยวข้องกับการร่างภาพด้วยมือและทฤษฎีทางคอมพิวเตอร์ที่เกี่ยวข้อง นอกจากนี้ยังเป็นการวิเคราะห์ข้อมูลที่ได้มาเพื่อใช้เป็นแนวทางในการพัฒนาโปรแกรมคอมพิวเตอร์เพื่อช่วยในการออกแบบของสถาปนิก ให้สามารถใช้การร่างภาพด้วยมือบนคอมพิวเตอร์เพื่อแสดงแนวความคิด และกระบวนการออกแบบของตนได้อย่างมีประสิทธิภาพ

จากการศึกษาลักษณะของการร่างภาพด้วยมือทางสถาปัตยกรรมพบว่าพื้นฐานของการร่างภาพทางสถาปัตยกรรมแบ่งออกเป็น 2 ลักษณะได้แก่การร่างภาพด้วยการวาดเส้นแบบเรขาคณิต และการร่างภาพด้วยการวาดเส้นแบบอิสระซึ่งมีรูปแบบและลักษณะการร่างภาพที่แตกต่างกัน งานวิจัยชิ้นนี้ได้ศึกษาขั้นตอนและวิธีทางคอมพิวเตอร์และทฤษฎีการคำนวณทางคณิตศาสตร์หลายวิธี ได้แก่ วิธีรับรู้รูปแบบที่เหมือนกันของภาพวาดลายเส้น วิธีรับรู้จุดมุมของภาพด้วยการหาค่าเวลาจากการลากเส้น วิธีรับรู้การขนานกันและความยาวของเส้น มาพัฒนาเป็นโปรแกรมคอมพิวเตอร์ที่สามารถรับรู้วัตถุจากการร่างภาพด้วยมือที่ปรากฏบนจอภาพ และยังสามารถนำไปปรับปรุงให้เป็นรูปทรง 3 มิติเบื้องต้นในขั้นสุดท้ายได้

ผลจากการวิจัยพบว่าโปรแกรมช่วยให้สถาปนิกสามารถใช้การร่างภาพด้วยมือ เพื่อช่วยงานออกแบบทางสถาปัตยกรรม เช่นการสร้างหุ่นจำลอง 3 มิติเบื้องต้นด้วยคอมพิวเตอร์ได้ง่ายขึ้น และยังสามารถนำผลของการวิจัยไปประยุกต์ใช้กับการพัฒนาโปรแกรมสร้างรูปทรง 3 มิติจากการร่างภาพด้วยมือที่ซับซ้อนด้วยคอมพิวเตอร์ต่อไปได้

ภาควิชา.....สถาปัตยกรรมศาสตร์..... ลายมือชื่อนิสิต.....
สาขาวิชา.....สถาปัตยกรรม..... ลายมือชื่ออาจารย์ที่ปรึกษา.....
ปีการศึกษา 2546 ลายมือชื่ออาจารย์ที่ปรึกษาร่วม.....

457 41590 25 : MAJOR ARCHITECTURE

KEY WORD: ARCHITECTURAL SKETCHES / RECOGNITION / SKETCH / 3D MODELING

POJCHARA JATUPOJ : THESIS TITLE. (COMPUTER ALGORITHM FOR OBJECT RECOGNITION FROM A FREEHAND SKETCH : BASIC 3D MODELING) THESIS ADVISOR : ASST. PROF. KRAWEEKRAI SRIHIRUN, THESIS COADVISOR : THIDASIRI BHATRAKARN, 106 pp. ISBN 974-17-4471-4.

The main objective of this research was to study freehand architectural sketches and architectural design that would develop the computer algorithm for object recognition and to create an image of the object by applying with the basic 3D modeling. This research was conducted by collecting the data of sketch recognition and related software to analyst relavant information and to develop new software that would be practical for architects. This software would be helpful for architects to communicate their ideas and design process more efficiently.

From the studies of architectural sketches, there were two basic types of line sketching, geometric and freehand line sketching. Because both types were varied by forms and styles of strokes many algorithms of sketch recognition were studies by using differences mathematical equations such as, pattern recognition, corner detection by checking the time differences, parallel and length test of line recognition. This research was used those algorithms to develope the specific sketch recognition software. The software was able to convert a sketch into a computer geometry and create a basic 3D modeling from a sketch at the end.

The result of this research revealed that the freehand sketch recognition software can help architects using a freehand sketch in their architectural designs as creating a basic 3D model more easily. Furthermore, this algorithm can be applied to develop a freehand sketch recognition software for other complicated architectural 3D modelings in the future.

Department.....ARCHITECTURE……... Student's signature.....

Field of study...ARCHITECTURE…….. Advisor's signature.....

Academic year 2003…… Co-advisor's signature.....

กิตติกรรมประกาศ

ขอขอบพระคุณ บิดา มารดาและครอบครัวที่ช่วยสนับสนุนและให้กำลังใจในการศึกษาจนมาถึงจุดนี้

ขอขอบคุณจิระภา ศรีคำ สำหรับกำลังใจที่ดีและมุมมองใหม่ๆในการวิจัย

ขอขอบคุณผ.ศ. กวีไกร ศรีหิรัญ และอาจารย์ธิดาสิริ ภัทรากาญจน์ ที่ได้ให้คำแนะนำแนวทางในการทำงานและข้อมูลที่เป็นประโยชน์อย่างยิ่งในการทำวิจัย

และขอขอบคุณอาจารย์ ภาควิชาสถาปัตยกรรมศาสตร์ สาขาวิชาสถาปัตยกรรมกลุ่มวิชาคอมพิวเตอร์ในการออกแบบสถาปัตยกรรม คณะสถาปัตยกรรมศาสตร์ ทุกท่านที่ได้ให้พื้นฐานและความรู้ใหม่ๆ ในการวิจัย

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง

บทคัดย่อภาษาอังกฤษ ..... จ

กิตติกรรมประกาศ ..... ฉ

สารบัญ ..... ช

สารบัญตาราง ..... ฎ

สารบัญภาพ ..... ฎ

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหาของการศึกษา 1

1.2 วัตถุประสงค์ของการวิจัย 2

1.3 ขอบเขตของการวิจัย 3

1.4 แผนผังแสดงวิธีดำเนินการวิจัย 3

1.5 วิธีดำเนินงานวิจัย 4

1.6 ระยะเวลาในการวิจัย 5

1.7 ประโยชน์ที่คาดว่าจะได้รับ 5

บทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้อง 6

2.1 ทฤษฎีที่เกี่ยวข้องกับการวาดภาพลายเส้น 6

2.1.1 ความหมายของการวาดภาพลายเส้น 6

2.1.2 พื้นฐานการวาดภาพลายเส้นด้วยมือ 7

2.2 ทฤษฎีการสร้างรูปทรง 3 มิติ 8

2.3 ขั้นตอนวิธีทางคอมพิวเตอร์ในการรับรู้ภาพวาดลายเส้น 12

2.3.1 การรับรู้ภาพกราฟิก 12

2.3.2 การวิเคราะห์มุมจากการวาดภาพลายเส้น 14

2.3.3 การวิเคราะห์รูปแบบที่เหมือนกันของภาพวาดลายเส้น 16

2.3.4 การทดสอบความขนานกันของเส้น 18

2.3.5 การทดสอบความยาวของเส้น 19

สารบัญ (ต่อ)

หน้า

2.3.6 การวาดเส้นตรงแบบดีดีเอ 21
2.3.7 การหาระยะห่างระหว่างจุดภาพ 22
2.2.8 การหาเส้นขอบบนวัตถุ 22
2.3.9 การค้นหามุมของวัตถุในภาพ 23
2.3.10 การสร้างรูปวงกลม 25
2.3.11 ผลกระทบของการใช้คอมพิวเตอร์ในการออกแบบขั้นต้น 27
2.3.12 การเก็บข้อมูลแบบเว็กเตอร์และแรสเตอร์ 27
2.4 โปรแกรมที่เกี่ยวข้องกับการวาดภาพลายเส้น 27
2.4.1 โปรแกรม Smooth Teddy 28
2.4.2 โปรแกรม Chateau 29
2.4.3 โปรแกรม Quick Sketch 31
2.4.4 โปรแกรม Sketching Interfaces Like Krazy (SILK) 32
2.4.5 โปรแกรม PerSketch (A Perceptually-Supported Sketch Editor) 33
2.4.6 โปรแกรม AutoDesk Architectural Studio 35
2.4.7 งานวิจัย The Computer's Role in Sketch Design 36
2.4.8 โปรแกรม Electronic Cocktail Napkin 37
2.5 สรุปลักษณะสำคัญร่วมกันของโปรแกรมที่เกี่ยวข้องกับการวาดภาพลายเส้น 39
2.5.1 เปรียบเทียบความถี่ของคุณลักษณะที่สำคัญของโปรแกรม 39
บทที่ 3 แนวทางการออกแบบโปรแกรม 41
3.1 การเลือกเครื่องมือเพื่อใช้ในการวิจัย 41
3.1.1 Hardware 41
3.1.2 Software ได้เลือกโปรแกรมดังต่อไปนี้ 42
3.2 การวิเคราะห์ข้อมูลจากเอกสารและงานวิจัยที่เกี่ยวข้อง 42
3.2.1 การพัฒนาเครื่องมือวาดเส้น 43
3.2.2 การพัฒนาระบบเลเยอร์ 43
3.2.3 การพัฒนาระบบวาดเส้นเรขาคณิตและระบบวาดเส้นอิสระ 44

สารบัญ (ต่อ)

หน้า

3.2.4 การพัฒนาระบบแสดงทัศนียภาพ 45
3.2.5 การพัฒนาระบบรับรู้วัตถุ 46
3.2.6 การนำเอาปากกาอิเล็กทรอนิกส์มาใช้ในงานวิจัย 47
3.2.7 การพัฒนาระบบสร้างหุ่นจำลอง 3 มิติ 49
3.3 การกำหนดตัวแปรเพื่อนำมาใช้ในการพัฒนาโปรแกรม 50
3.3.1 ตัวแปรในการพัฒนาระบบวาดเส้นเรขาคณิตและวาดเส้นอิสระ 50
3.3.2 ตัวแปรในการพัฒนาระบบรับรู้วัตถุ 51
3.4 การออกแบบและพัฒนาโปรแกรม 58
3.4.1 การพัฒนาส่วนรับข้อมูล input จากการวาดภาพลายเส้น 59
3.4.2 การพัฒนาโปรแกรมด้วยวิธีรับรู้ข้อมูลจากภาพวาดลายเส้น 60
3.4.3 การพัฒนาส่วนโต้ตอบผู้ใช้ 68
บทที่ 4 ผลการออกแบบโปรแกรม 70
4.1 เตรียมอุปกรณ์ 70
4.1.1 การปรับแรงกด 70
4.1.2 ระบบปฏิบัติการที่ใช้ 71
4.1.3 อุปกรณ์เร่งความเร็วกราฟิก 71
4.2 การติดตั้งโปรแกรม 72
4.3 ปุ่มและคำสั่งในการทำงาน 73
4.3.1 ส่วนเมนูหลัก 73
4.3.2 ส่วนปุ่มในการทำงาน 74
4.3.3 ส่วนพื้นที่วาดภาพ 75
4.3.4 ส่วนแสดงข้อมูลประกอบการทำงาน 76
4.3.5 ส่วนรายงานผล 76
4.4 การใช้งานโปรแกรมโดยทั่วไป 77
4.4.1 การวาดภาพ 77
4.4.2 การรับรูปภาพวาดลายเส้น 79

สารบัญ (ต่อ)

หน้า

4.4.3 การเปลี่ยนมุมมองและการสร้างภาพ 3 มิติ 81
4.5 การปรับค่าในการใช้งาน 84
4.6 การจัดทำรายงาน 85
บทที่ 5 การประเมินผลการวิจัย 87
5.1 การรับรู้ภาพวาดลายเส้น 87
5.2 การสร้างรูปทรง 3 มิติเบื้องต้น 90
บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ 92
6.1 สรุปผลการวิจัย 92
6.2 ปัญหาและอุปสรรค 93
6.2.1 ปัญหาด้าน Hardware 93
6.2.2 ปัญหาด้าน Software 94
6.3 ข้อเสนอแนะ 95
6.3.1 การเลือกอุปกรณ์ Hardware 95
6.3.2 การพัฒนาโปรแกรมในอนาคต 96

รายการอ้างอิง 98
ภาคผนวก 101
ประวัติผู้เขียนวิทยานิพนธ์ 106

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญตาราง

หน้า

ตาราง 1.1 กำหนดระยะเวลาในการทำวิจัย 5
ตาราง 2.1 สรุปลักษณะร่วมกันของโปรแกรมที่เกี่ยวข้อง 39
ตาราง 2.2 ตารางเปรียบเทียบความถี่ของคุณลักษณะที่สำคัญของโปรแกรมที่เกี่ยวข้อง 39
ตาราง 3.1 ตัวอย่างโปรแกรมที่ใช้วิธีรับรู้เส้นตรง 61
ตาราง 3.2 ตัวอย่างโปรแกรมรับรู้การตัดกันของเส้น 63
ตาราง 3.3 ตัวอย่างโปรแกรมรับรู้ระยะห่างของเส้น 64
ตาราง 3.4 ตัวอย่างโปรแกรมรับรู้ระยะยื่นของเส้น 65
ตาราง 3.5 ตารางแสดงโปรแกรมหาความแตกต่างของเวลาในหน่วย 1/1000 วินาที 66
ตาราง 5.1 แสดงผลการทดสอบการรับรู้ภาพวาดลายเส้น a1-b3 88
ตาราง 5.2 แสดงผลการทดสอบการรับรู้ภาพวาดลายเส้น c1-c3 88

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

หน้า

รูปภาพ 2.1 การวาดภาพลายเส้น (ซ้าย) ซึ่งนำไปสู่การพัฒนาให้เป็นงานออกแบบ (ขวา) 7
รูปภาพ 2.2 แสดงภาพที่ผู้ใช้งานวาดขึ้น a กับภาพที่คอมพิวเตอร์เก็บข้อมูล b 14
รูปภาพ 2.3 กราฟแสดงความเร็วและเวลาของการวาดภาพสี่เหลี่ยม 15
รูปภาพ 2.4 กราฟแสดงความแตกต่างของเวลาในการลากเส้นแต่ละจุด 15
รูปภาพ 2.5 แสดงการวาดภาพอย่างต่อเนื่อง 16
รูปภาพ 2.6 แสดงการวาดภาพลายเส้นทางสถาปัตยกรรมอย่างอิสระ 17
รูปภาพ 2.7 ภาพแสดงลักษณะการวาดภาพลายเส้น ที่คอมพิวเตอร์ต้องตีความหมายให้ถูกต้อง 17
รูปภาพ 2.8 การเปรียบเทียบความขนานกันของเส้น 18
รูปภาพ 2.9 การแปลงภาพวาดลายเส้น 20
รูปภาพ 2.10 แสดงการเลือกจุดภาพเพื่อวาดเส้นตรง 22
รูปภาพ 2.11 แสดงการตัดสินจุดขอบ โดย P1 ไม่เป็นจุดภาพบนเส้นขอบ 23
รูปภาพ 2.12 แสดงวัตถุและเส้นขอบวัตถุ ภาพซ้าย ก่อนหาเส้นขอบ ภาพขวา หลังหาเส้นขอบ 23
รูปภาพ 2.13 ตัวอย่างลำดับขั้นตอนการทำงานของโปรแกรมในการสร้างหุ่นจำลอง 3 มิติ 28
รูปภาพ 2.15 ภาพตัวอย่างของโปรแกรม Chateau 30
รูปภาพ 2.17 ภาพตัวอย่างขั้นตอนการทำงานของโปรแกรม 31
รูปภาพ 2.19 ภาพลักษณะการใช้งานโปรแกรม SILK ในขั้นต่างๆ 33
รูปภาพ 2.20 ตัวอย่างการใช้งานโปรแกรม PerSketch 34
รูปภาพ 2.21 ตัวอย่างวิธีการทางคอมพิวเตอร์ที่ใช้ในการแปลงภาพวาดลายเส้น 34
รูปภาพ 2.22 ตัวอย่าง User Interface ของโปรแกรม Architectural Studio 35
รูปภาพ 2.23 ตัวอย่างการใช้โปรแกรม Architectural Studio 36
รูปภาพ 2.24 ตัวอย่างโปรแกรมทดลองประกอบงานวิจัย 37
รูปภาพ 2.25 ภาพตัวอย่างการใช้งาน Electronic Cocktail Napkin 38
รูปภาพ 2.26 ภาพแสดงตัวอย่างการนำเอาโปรแกรมมาประยุกต์ใช้งาน 38
รูปภาพ 2.27 คุณลักษณะที่พบมากที่สุดคือ การวาดเส้นเรขาคณิต 40
รูปภาพ 3.1 ระบบเลเยอร์มีความเหมาะสมกับการทำงานเขียนแบบ 44

สารบัญภาพ (ต่อ)

หน้า

รูปภาพ 3.2 โปรแกรม Quick Sketch จะแปลงภาพวาดลายเส้น 45
รูปภาพ 3.3 การทำงานของโปรแกรม Chateau 46
รูปภาพ 3.4 ลำดับการทำงานอย่างง่ายของโปรแกรม Persketch 46
รูปภาพ 3.5 โปรแกรม Electronic Cocktail Napkin 47
รูปภาพ 3.6 ตัวอย่างอุปกรณ์รับข้อมูลด้วยปากกาอิเล็กทรอนิกส์ 48
รูปภาพ 3.7 ตัวอย่างการใช้อุปกรณ์รับข้อมูลด้วยปากกาอิเล็กทรอนิกส์ 48
รูปภาพ 3.8 อุปกรณ์รับข้อมูลด้วยปากกาอิเล็กทรอนิกส์สำหรับ งานออกแบบและงานกราฟิก 49
รูปภาพ 3.9 งานวิจัย Teddy โดย Takeo Igarashi เน้นการสร้างหุ่นจำลอง 3 มิติ 49
รูปภาพ 3.10 การพัฒนาระบบวาดวาดเส้นเพื่อรองรับการวาดเส้นที่นิยมใช้ในงานสถาปัตยกรรม 51
รูปภาพ 3.11 ภาพแสดงตัวอย่างการเปรียบเทียบเส้นสองเส้นโดยมีค่าความชัน M1 และ M2 53
รูปภาพ 3.12 ภาพแสดงจุดตัดของเส้นสองเส้นโดยที่จุด x, y อยู่บนส่วนของเส้นตรงทั้งสองเส้น 55
รูปภาพ 3.13 ภาพแสดงจุดตัดของเส้นสองเส้นโดยที่จุด x, y ไม่ได้อยู่บนส่วนของเส้นตรงใดเลย 55
รูปภาพ 3.14 ภาพ (a) แสดงการวาดเส้นที่มีการเว้นระยะห่างน้อยกว่าค่าที่กำหนด (Li%) ภาพ (b) แสดงระยะที่มากกว่าค่าที่กำหนด (Li%) 56
รูปภาพ 3.15 ภาพแสดงรูปแบบของการวาดเส้นที่ผู้วาดจะลากเส้นยื่นออกไปจากจุดตัดของเส้น 57
รูปภาพ 3.16 ภาพแสดงระยะเวลาในการวาดภาพรูปหลายเหลี่ยมจะเห็นความแตกต่าง ของเวลาที่เพิ่มขึ้นเมื่อถึงจุดมุมของภาพ 58
รูปภาพ 3.17 ภาพ (a) แสดงการวาดภาพรูปหลายเหลี่ยมโดยผู้วาดต้องการให้จุดที่มีวงกลม คือจุดมุมของภาพและภาพ (b) คือการกำหนดจุดมุมด้วยวิธีการหาความแตกต่างของเวลา 58
รูปภาพ 3.18 ภาพแสดงการตั้งค่าแรงกดของปากกาในระดับ 70% โดยใช้ปากกาอิเล็กทรอนิกส์ ของ Genius รุ่น Wizard Pen 4x3 59
รูปภาพ 3.19 ภาพแสดงการวาดเส้นตรงที่มีความชันใกล้เคียงกัน (ซ้าย) โปรแกรมจะรับรู้ว่าเป็นเส้นตรงเดียวกัน และการวาดเส้นตรงที่มีความชันระหว่างจุดต่างกัน (ขวา) มากเกินกว่าตัวแปรที่กำหนดโปรแกรมจะรับรู้ว่าไม่เป็นเส้นตรงเดียวกัน 61

สารบัญภาพ (ต่อ)

หน้า

รูปภาพ 3.20 ภาพแสดงการทำงานของโปรแกรมที่สามารถรับรู้เส้นที่ตัดกันได้อย่างถูกต้อง (ซ้าย) และยังสามารถยกเลิกการจุดตัดในกรณีที่จุดตัดไม่ได้อยู่บนส่วนของเส้นตรงทั้งสองเส้น 63
รูปภาพ 3.21 ภาพแสดงการรับรู้ระยะห่างของจุดปลายของเส้นที่มีค่าเกินกว่าที่กำหนด (ซ้าย)..... 64
รูปภาพ 3.22 ภาพแสดงการทำงานของโปรแกรมโดยจะรับรู้ระยะยื่นเฉพาะเส้นที่ยื่นออกไปไม่เกินค่าที่กำหนด..... 65
รูปภาพ 3.23 ภาพแสดงการทำงานของโปรแกรมรับรู้จุดมุมของภาพ 66
รูปภาพ 3.24 ภาพแสดงการเก็บค่าความแตกต่างของเวลาของจุดในการวาดภาพลายเส้น..... 67
รูปภาพ 3.25 ภาพแสดงระบบตารางขนาด 3x3 (a) และวิธีการรับรู้ลำดับการวาดเส้น (b) 67
รูปภาพ 3.26 ภาพแสดงตัวอย่างการวาดเส้นตามลำดับ 0,2,8,6 (a) และผลจากการรับรู้ตรงกันกับฐานข้อมูลเป็นหุ่นจำลอง Bed (b) 68
รูปภาพ 3.27 ภาพแสดงการออกแบบส่วนโต้ตอบผู้ใช้โดยออกแบบให้มีปุ่มน้อยที่สุดและเน้นพื้นที่การวาดภาพ..... 69
รูปภาพ 3.28 ภาพแสดงส่วนแสดงรายงานข้อมูลที่สำคัญจากการวาดภาพลายเส้น โดยสามารถบันทึกเป็นแฟ้มเพื่อนำไปวิเคราะห์ต่อด้วยโปรแกรม MS Excel ได้..... 69
รูปภาพ 4.1 อุปกรณ์รับข้อมูลด้วยปากกาอิเล็กทรอนิกส์ 71
รูปภาพ 4.2 การตั้งค่าแรงกดของปากกาเท่ากับ 70% 71
รูปภาพ 4.3 ภาพแสดงแฟ้มสำหรับติดตั้งโปรแกรม..... 72
รูปภาพ 4.4 ภาพแสดงหน้าจอการติดตั้งโปรแกรม..... 72
รูปภาพ 4.5 ภาพแสดง icon สำหรับเปิดโปรแกรม..... 72
รูปภาพ 4.6 ภาพแสดงหน้าจอการทำงานของโปรแกรมโดยตั้งชื่อโปรแกรมว่า SketchBoard 73
รูปภาพ 4.7 ภาพแสดงเมนูหลักของโปรแกรม..... 74
รูปภาพ 4.8 ภาพแสดงปุ่มหลักในการใช้โปรแกรม 75
รูปภาพ 4.9 ภาพแสดงการวาดภาพลายเส้นบนพื้นที่สำหรับวาดภาพ 75
รูปภาพ 4.10 ภาพแสดงข้อมูลประกอบการทำงาน..... 76
รูปภาพ 4.11 ภาพแสดงส่วนรายงานผล..... 76

สารบัญภาพ (ต่อ)

หน้า

รูปภาพ 4.12 ภาพแสดงการนำรายงานไปวิเคราะห์ด้วยโปรแกรม Microsoft Excel 77
รูปภาพ 4.13 ภาพแสดงการวาดภาพแบบเรขาคณิต 78
รูปภาพ 4.14 ภาพแสดงการวาดภาพลายเส้นอิสระแบบต่อเนื่อง (a) และผลจากการรับรู้เส้น (b)..... 78
รูปภาพ 4.15 ภาพแสดงการวาดเส้นแบบเรขาคณิต..... 79
รูปภาพ 4.16 การรับรู้ภาพวาดลายเส้นอิสระแบบตัดมุม..... 79
รูปภาพ 4.17 การรับรู้ภาพวาดลายเส้นอิสระแบบเส้นต่อเนื่อง 79
รูปภาพ 4.18 การรับรู้ภาพวาดลายเส้นอิสระโดยรับรู้ระยะเว้นว่าง 80
รูปภาพ 4.19 การรับรู้ภาพวาดลายเส้นอิสระโดยรับรู้ระยะยื่น..... 80
รูปภาพ 4.20 รับรู้ภาพวาดลายเส้นแบบเรขาคณิต..... 80
รูปภาพ 4.21 รับรู้รูปแบบที่เหมือนกันของภาพวาดลายเส้น (Sketches Pattern Recognition)..... 80
รูปภาพ 4.22 ภาพแสดงการเปลี่ยนมุมมอง 3 มิติ 81
รูปภาพ 4.23 ภาพแสดงการสร้างความสูงในแกน z ให้กับวัตถุ 81
รูปภาพ 4.24 ภาพแสดงการปรับมุมมองภาพโดยการลากปากกาในพื้นที่ทำงานเพื่อหมุนกล้อง..... 82
รูปภาพ 4.25 ภาพแสดงการบอกระยะของวัตถุที่สร้างขึ้น..... 82
รูปภาพ 4.26 ภาพแสดงการสร้างวัตถุที่ระดับความสูงในแกน z ที่ผู้ใช้เป็นผู้กำหนด 82
รูปภาพ 4.27 ภาพแสดงการใส่เงาให้กับวัตถุทั้งหมดในภาพเพื่อสร้างความสมจริงให้กับวัตถุ..... 83
รูปภาพ 4.28 ภาพแสดงการสร้างเส้นโค้ง (a) และแผ่นระนาบโค้ง 3 มิติ (b) 83
รูปภาพ 4.29 ภาพแสดงการจัดหุ่นจำลองชุดเก้าอี้กับผนังโค้งจากการรับรู้ภาพวาดลายเส้น..... 84
รูปภาพ 4.30 ภาพแสดงส่วนปรับแต่งการทำงานของโปรแกรม 85
รูปภาพ 4.31 ภาพแสดงระบบรายงานผลการทำงานของโปรแกรม..... 85
รูปภาพ 4.32 ภาพแสดงการบันทึกข้อมูลจากรายงานเป็นแฟ้มข้อมูล *.skb..... 86
รูปภาพ 4.33 ภาพแสดงการเปิดข้อมูลที่บันทึกไว้ด้วยโปรแกรม Microsoft Excel 86
รูปภาพ 4.34 ภาพแสดงการวิเคราะห์ข้อมูลวาดภาพลายเส้นด้วยกราฟจากโปรแกรม Microsoft Excel จะเห็นการใช้เวลาสูงกว่าปกติ 4 จุด ซึ่งเป็นจุดมุมของการวาดภาพสี่เหลี่ยม 86
รูปภาพ 5.1 ต้นแบบที่ใช้ในการทดสอบการรับรู้วาดภาพลายเส้น..... 87

สารบัญภาพ (ต่อ)

หน้า

รูปภาพ 5.2 แสดงการทดสอบการรับรู้ภาพวาดลายเส้นด้วยตัวแปร Time Difference 89
รูปภาพ 5.3 แสดงความถูกต้องในการรับรู้ ซึ่งโปรแกรมสามารถแปลงภาพวาดลายเส้นให้เป็นภาพ3 มิติได้ทันที 89
รูปภาพ 5.4 แสดงความแตกต่างของเวลาในการวาดภาพลายเส้น 90
รูปภาพ 5.5 ภาพแสดงการประยุกต์ใช้โปรแกรมเพื่อช่วยในการสร้างรูปทรง 3 มิติเบื้องต้น 90
รูปภาพ 5.6 ภาพแสดงตัวอย่างการสร้างรูปทรง 3 มิติเบื้องต้นเพื่อช่วยในการออกแบบ 91
รูปภาพ 5.7 ภาพแสดงตัวอย่างการสร้างหุ่นจำลองจากระบบรับรู้รูปแบบที่เหมือนกันของภาพวาดลายเส้น (Sketches Pattern Recognition) 91
รูปภาพ 5.8 ภาพแสดงตัวอย่างการประยุกต์ใช้ในการออกแบบ 91
รูปภาพ 6.1 ภาพแสดงตัวอย่างอุปกรณ์ Tablet PC (ภาพจาก http://www.toshiba.com) 96

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหาของการศึกษา

การวาดภาพลายเส้นเป็นวิธีการที่สถาปนิกมักจะใช้ในการทดสอบความคิด และสื่อสารความคิดในการออกแบบอยู่เสมอ สถาปนิกจะเริ่มกระบวนการออกแบบด้วยการออกแบบร่าง (ธิดาสิริ ภัทรากาญจน์, 2546: 181) โดยใช้การวาดรูป หรือการวาดภาพลายเส้นด้วยกระดาษและปากกาเพื่อร่างงานออกแบบขั้นต้น แล้วจึงนำงานออกแบบร่างไปพัฒนาเป็นงานออกแบบที่สมบูรณ์ในขั้นต่อไป ซึ่งสาเหตุที่มีการใช้ภาพวาดลายเส้นในการออกแบบร่าง เพราะเป็นขั้นตอนวิธีที่สามารถทำได้อย่างสะดวกรวดเร็วและให้ความรู้สึกอิสระแก่สถาปนิก ด้วยคุณสมบัติของเส้นที่ไม่ชัดเจนและไม่มีระยะที่แน่นอนทำให้การวาดภาพลายเส้นเป็นวิธีการที่มีความเหมาะ สมกับงานออกแบบขั้นต้นของสถาปนิกเป็นอย่างยิ่ง

จากการสังเกตการทำงานของสถาปนิกในปัจจุบันที่มีการใช้คอมพิวเตอร์มาช่วยในการทำงานเขียนแบบเป็นหลักและนำมาช่วยสร้างภาพจำลอง 3 มิติกันอย่างแพร่หลาย แต่กลับไม่มีการนำเอาคอมพิวเตอร์มาช่วยในการวาดภาพลายเส้นเพื่อการออกแบบเท่าที่ควร ทั้งที่เป็นวิธีการพื้นฐานที่สถาปนิกใช้ในการออกแบบอยู่เสมอ โดยสังเกตได้จากการทำงานของสถาปนิกเมื่อเริ่มออกแบบสถาปนิกจะวาดภาพลายเส้นลงบนกระดาษเพื่อพัฒนางานออกแบบในขั้นต้นจนได้ภาพร่างของงานออกแบบทั้งหมด แล้วจึงนำมาวางไว้เป็นแบบหรือนำมาสแกนเป็นภาพเพื่อพัฒนาต่อบนคอมพิวเตอร์หรือเพื่อเขียนเป็นงานออกแบบขั้นสุดท้าย โดยจะใช้โปรแกรมที่แพร่หลายในสำนักงานสถาปนิก เช่น AutoCAD[1], FormZ[2], ArchiCAD[3] เป็นเครื่องมือในการเขียนแบบ ซึ่งโปรแกรมเหล่านี้ส่วนใหญ่มีขั้นตอนการทำงานแบบ Window Icon Menu และ Pointing Device (Takeo Igarashi, 1990) ทำให้เวลาที่สถาปนิกต้องการวาดภาพจะต้องแยกภาพในความคิดออกเป็นส่วนๆ ก่อน เช่น จุด เส้น กล่องสี่เหลี่ยม เพื่อจะไปกดปุ่มและลากเมาส์ เพื่อวาดทีละคำสั่งจนได้ภาพที่ต้องการ ซึ่งขั้นตอนเหล่านี้แตกต่างไปจากการวาดภาพลายเส้นมาก เพราะการวาดภาพลายเส้นจะให้ความรู้สึกอิสระและไม่ต้องการเครื่องมือในการทำงานมาก สามารถเริ่มจากจุดไหนก่อนก็ได้โดยใช้สายตากะระยะ (John

[1] Autodesk http://www.autodesk.com

[2] auto-des-sys http://www.formz.com

[3] Graphisoft http://www.graphisoft.com

Rieman, 1999) จึงทำให้เกิดข้อแตกต่างกันอย่างมากระหว่างการใช้มือวาดภาพลายเส้นกับการใช้คอมพิวเตอร์ช่วยในงานสถาปัตยกรรม หากมีโปรแกรมคอมพิวเตอร์ที่ช่วยให้สถาปนิกสามารถวาดภาพลายเส้นได้อย่างอิสระโดยไม่ต้องใช้เครื่องมือในการทำงานมาก น่าจะช่วยให้สถาปนิกสามารถใช้การวาดภาพลายเส้นมาช่วยออกแบบงานสถาปัตยกรรมขั้นต้นบนคอมพิวเตอร์ได้ และช่วยให้สถาปนิกสามารถพัฒนางานออกแบบได้อย่างต่อเนื่องและสะดวกมากขึ้น โดยเฉพาะหากนำมาช่วยในการออกแบบองค์ประกอบทางสถาปัตยกรรมที่ไม่มีความซับซ้อนมาก เช่น การสร้างรูปทรง 3 มิติ เบื้องต้น เป็นต้น

จากประเด็นดังกล่าวทำให้เกิดความสนใจ ที่จะพัฒนาขั้นตอนวิธีทางคอมพิวเตอร์เพื่อรับรู้วัตถุจากภาพลายเส้น โดยจะประยุกต์ใช้กับการสร้างรูปทรง 3 มิติเบื้องต้น ซึ่งการพัฒนาโปรแกรมนี้จะช่วยให้สถาปนิกสามารถวาดภาพลายเส้นด้วยมือลงบนคอมพิวเตอร์เลียนแบบการทำงานด้วยปากกาและกระดาษ จะทำให้เกิดความต่อเนื่องของกระบวนการออกแบบไปสู่การสร้างรูปทรง 3 มิติเพื่อให้การทำงานมีความสะดวกในการทำงานมากขึ้น ทั้งนี้การวิจัยชิ้นนี้ไม่ได้มีจุดมุ่งหมายให้สถาปนิกหันมาใช้คอมพิวเตอร์วาดภาพลายเส้นแทนการใช้กระดาษ แต่เป็นการสนับสนุนให้เกิดทางเลือกในการทำงานจากการวาดภาพลายเส้นไปสู่การพัฒนางานออกแบบขั้นต่อไปบนคอมพิวเตอร์ได้ทันที และสนับสนุนให้สถาปนิกสามารถใช้คอมพิวเตอร์ช่วยในการออกแบบได้สะดวกมากขึ้น อีกด้วย

1.2 วัตถุประสงค์ของการวิจัย

ในการวิจัยมีวัตถุประสงค์ที่สำคัญดังต่อไปนี้

1. พัฒนาโปรแกรมคอมพิวเตอร์โดยใช้ขั้นตอนวิธีทางคอมพิวเตอร์ (algorithm) เพื่อรับรู้วัตถุจากภาพลายเส้น
2. เพื่อพัฒนาโปรแกรมคอมพิวเตอร์ช่วยในการสร้างรูปทรง 3 มิติเบื้องต้นให้สถาปนิกสามารถทำงานได้สะดวกมากขึ้น

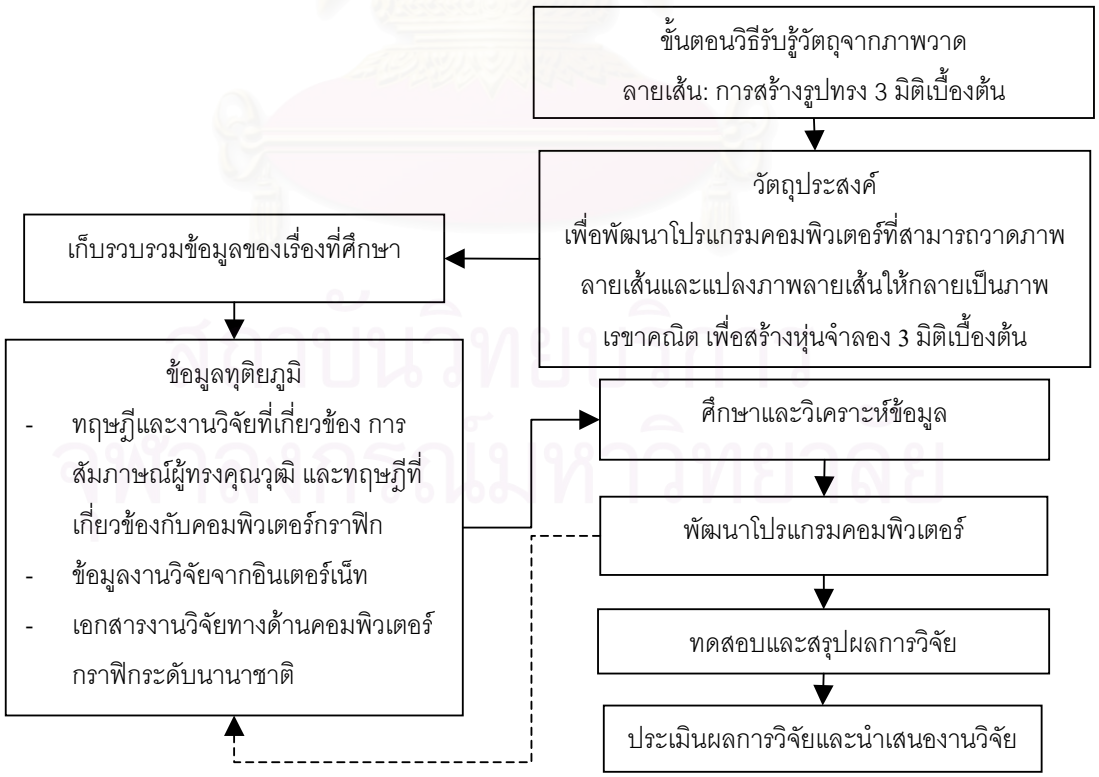
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

1.3 ขอบเขตของการวิจัย

ในการวิจัยมีขอบเขตของการศึกษาดังต่อไปนี้

1. ลักษณะของการวาดภาพลายเส้นที่นำมาใช้ในการวิจัยจะเน้นที่การวาดภาพลายเส้นทางสถาปัตยกรรม โดยจะศึกษาเฉพาะการวาดเส้นแบบเรขาคณิตด้วยมือในลักษณะต่างๆ
2. ศึกษาการโปรแกรมคอมพิวเตอร์และงานวิจัยที่เกี่ยวข้องกับการวาดภาพลายเส้น โดยศึกษาเฉพาะส่วนที่เกี่ยวข้องกับการวาดเส้นเรขาคณิตและเทคนิควิธีในการวาดภาพลายเส้นด้วยคอมพิวเตอร์ เช่น วิธีการกำหนดจุดตัดของเส้น การกำหนดจุดกึ่งกลางของเส้นโค้ง การกำหนดจุดตั้งฉาก และวิธีการรับข้อมูลจากอุปกรณ์อินพุตแบบต่างๆ เป็นต้น
3. ศึกษางานวิจัยที่เกี่ยวข้องกับการสร้างภาพ 3 มิติด้วยคอมพิวเตอร์
4. โปรแกรมจะรับรู้ข้อมูลของวัตถุในคอมพิวเตอร์ แล้วแปลงด้วยวิธีทางคอมพิวเตอร์ให้เป็นรูปทรง 3 มิติเบื้องต้นได้
5. โปรแกรมนี้สามารถสร้างรูปทรง 3 มิติเบื้องต้นและสามารถแสดงผลงานออกแบบด้วยคอมพิวเตอร์ได้

1.4 แผนผังแสดงวิธีดำเนินการวิจัย



1.5 วิธีดำเนินงานวิจัย

การวิจัยจะเน้นที่การวิจัยเชิงคุณภาพ (Qualitative Research) โดยใช้ระเบียบวิธีการศึกษาข้อมูลเชิงเอกสาร (Documentary Research) และการวิจัยเชิงทดลอง (Experimental Research) โดยเรียงลำดับกระบวนการวิจัยได้ดังนี้

1.5.1 ทำการศึกษาข้อมูลเชิงเอกสาร ได้แก่ การทบทวนวรรณกรรม ประกอบด้วยทฤษฎีและงานวิจัยที่เกี่ยวข้อง และการสัมภาษณ์ผู้ทรงคุณวุฒิ โดยแบ่งเป็น

- ตำราวิชาการทั้งภาษาไทยและภาษาอังกฤษทางด้านการพื้นฐานออกแบบสถาปัตยกรรมหรือหลักการวาดเส้นทางสถาปัตยกรรม
- วิทยานิพนธ์ (Thesis) ได้แก่ การศึกษาผลงานวิจัยที่เกี่ยวข้องทั้งในและต่างประเทศ
- เอกสารอื่นๆ เช่น บทความทางด้านการสร้างภาพด้วยคอมพิวเตอร์จากวารสารทางด้านคอมพิวเตอร์ วารสารทางด้านการพัฒนาคอมพิวเตอร์เพื่อการออกแบบสถาปัตยกรรมในต่างประเทศ เป็นต้น
- ข้อมูลจากอินเตอร์เน็ต
- เอกสารงานวิจัยทางด้านคอมพิวเตอร์กราฟิกระดับนานาชาติ ACM CHI SIGGRAPH

แหล่งข้อมูลเชิงเอกสารได้แก่

- ห้องสมุดของมหาวิทยาลัยต่างๆ เช่น สำนักวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย ห้องสมุดคณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ห้องสมุดคณะสถาปัตยกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ห้องสมุดคณะสถาปัตยกรรมศาสตร์ มหาวิทยาลัยรังสิต

แหล่งข้อมูลของผู้ทรงคุณวุฒิได้แก่

- อาจารย์ภาควิชาสถาปัตยกรรมศาสตร์ คณะสถาปัตยกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
- อาจารย์ภาควิชาสถาปัตยกรรมศาสตร์ สาขาวิชาสถาปัตยกรรม กลุ่มวิชาคอมพิวเตอร์ในการออกแบบสถาปัตยกรรม คณะสถาปัตยกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

1.5.2 ศึกษาและวิเคราะห์เทคนิคในการสร้างภาพลายเส้นบนคอมพิวเตอร์อย่างละเอียด โดยเน้นศึกษาทฤษฎีทางคอมพิวเตอร์ที่เกี่ยวข้องโดยแบ่งเป็นหัวข้อดังนี้

- การวาดเส้นเรขาคณิตด้วยคอมพิวเตอร์
- การเปลี่ยนมุมมองและแก้ไขข้อมูลวัตถุในภาพ
- เทคนิคการเขียนโปรแกรมเชิงวัตถุ
- การสร้างภาพ 3 มิติ ด้วย OpenGL version 1.2 [4]
- การสร้างภาพกราฟิกด้วย Microsoft Visual C++ version 6.0 [5]

1.5.3 พัฒนาโปรแกรมจากข้อมูลและวิธีการทางคอมพิวเตอร์ที่ได้มา

1.5.4 ทดสอบความถูกต้องของโปรแกรมและสรุปผลรายงานการวิจัย

1.5.5 ประเมินผลของการวิจัยและนำเสนองานวิจัย

1.6 ระยะเวลาในการวิจัย

	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.
เก็บรวบรวมข้อมูลเบื้องต้น	■	■							
ทบทวนวรรณกรรม		■	■	■	■	■			
พัฒนาโปรแกรม			■	■	■	■	■		
ทดสอบและสรุปผล							■	■	■
ประเมินผลและนำเสนอ									■

ตาราง 1.1 กำหนดระยะเวลาในการทำวิจัย

1.7 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถพัฒนาขั้นตอนวิธีการทางคอมพิวเตอร์เพื่อรับรู้ภาพวาดลายเส้นได้
2. โปรแกรมนี้สามารถสร้างภาพ 3 มิติ จากภาพวาดลายเส้นเพื่อแสดงผลงานออกแบบได้
3. โปรแกรมนี้สามารถช่วยให้สถาปนิกสามารถสร้างรูปทรง 3 มิติได้สะดวกขึ้น

[4] Silicon Graphic Inc. http://www.opengl.org

[5] Microsoft Corporation

บทที่ 2

เอกสารและงานวิจัยที่เกี่ยวข้อง

จากการรวบรวมและศึกษาเอกสารและงานวิจัยที่เกี่ยวข้อง ทำให้พบแง่มุมที่น่าสนใจเกี่ยวกับการวาดภาพลายเส้นและพบปัญหาหรือข้อจำกัดของการวิจัยที่เกี่ยวข้องทั้งทางด้านการพัฒนาโปรแกรมและทางด้านทฤษฎี โดยสามารถแบ่งข้อมูลจากการศึกษาทั้งหมดได้ 4 ประเภทดังนี้

2.1 ทฤษฎีที่เกี่ยวข้องกับการวาดภาพลายเส้น

2.2 ทฤษฎีการสร้างรูปทรง 3 มิติ

2.3 ขั้นตอนวิธีทางคอมพิวเตอร์ในการรับรู้ภาพวาดลายเส้น

2.4 โปรแกรมที่เกี่ยวข้องกับการวาดภาพลายเส้น

2.5 สรุปลักษณะสำคัญร่วมกันของโปรแกรมที่เกี่ยวข้องกับการวาดภาพลายเส้น

2.1 ทฤษฎีที่เกี่ยวข้องกับการวาดภาพลายเส้น

ทฤษฎีการวาดภาพลายเส้นคือความหมายของการวาดภาพลายเส้นและแนวความคิดพื้นฐานที่เกี่ยวข้องกับการวาดภาพลายเส้น ซึ่งจะนำไปสู่การพัฒนาโปรแกรมให้เหมาะสมกับการวาดภาพลายเส้นต่อไป โดยมีหัวข้อที่เกี่ยวข้องดังนี้

2.1.1 ความหมายของการวาดภาพลายเส้น

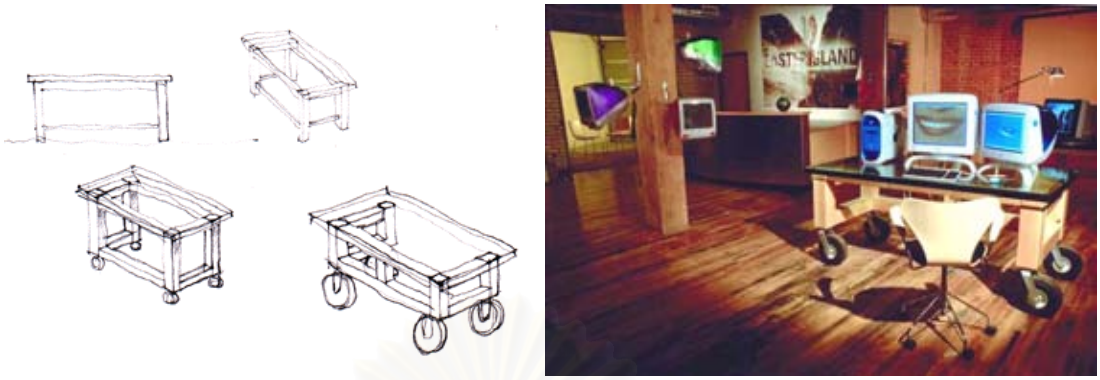
2.1.2 พื้นฐานการวาดภาพลายเส้นด้วยมือ

2.1.1 ความหมายของการวาดภาพลายเส้น

เมื่อศึกษาถึงความหมายของการวาดภาพลายเส้นหรือที่เรียกว่า Sketch โดยพิจารณาจากความหมายในพจนานุกรม (Webster, 1995) ได้ความหมายดังนี้

"**Sketch** (n.) a *simple, rough drawing or design, done rapidly and without much detail*."

การวาดภาพลายเส้นนั้นคือการวาดเส้นอย่างง่ายโดยมีการลดทอนรายละเอียดออกไป และในทางสถาปัตยกรรมนั้นการวาดภาพลักษณะนี้จะมีความสัมพันธ์กับกระบวนการออกแบบด้วย กล่าวคือ การวาดภาพลายเส้นจะถูกนำไปใช้เป็นอันดับแรกของการออกแบบ และนับเป็นทักษะขั้นพื้นฐานของการออกแบบทางด้านสถาปัตยกรรมที่จำเป็นต้องได้รับการฝึกฝน เช่น ทักษะการลากเส้นสายและการให้น้ำหนักแสงเงา เป็นต้น



รูปภาพ 2.1 การวาดภาพลายเส้น (ซ้าย) ซึ่งนำไปสู่การพัฒนาให้เป็นงานออกแบบ (ขวา)

2.1.2 **พื้นฐานการวาดภาพลายเส้นด้วยมือ**

ในการศึกษาลักษณะการวาดภาพลายเส้นของสถาปนิกว่ามีลักษณะอย่างไรบ้าง เพื่อให้สามารถพัฒนาโปรแกรมได้เหมาะสมกับการทำงานของสถาปนิกมากที่สุด ซึ่งพบว่าสถาปนิกจะใช้การวาดภาพลายเส้นในการออกแบบขั้นพื้นฐานมากที่สุดและจากหลักการออกแบบขั้นพื้นฐาน (เลอสม สถาปิตานนท์, 2537) สรุปได้ว่า การวาดภาพลายเส้นมีพื้นฐานมาจาก

จุด ซึ่งเป็นองค์ประกอบที่มองเห็นได้เมื่ออยู่บนกระดาษ รูปร่างของสิ่งที่เราเรียกว่าจุดนั้นจะมาจากความเล็กแต่ว่าความเล็กหรือใหญ่ของจุดขึ้นอยู่กับขนาดของกรอบที่ล้อมรอบจุดอยู่

เส้น เมื่อนำจุดมาประกอบกันโดยนำมาเรียงเป็นแถว จะทำให้มองเห็นติดกันเป็นเส้นโดยลักษณะของเส้นจะเป็นได้ทั้งเส้นตรง เส้นโค้งงอ คดไปมา โดยจะให้ความรู้สึกแตกต่างไปตามรูปร่างและขนาดของเส้น

ระนาบ เกิดจากแนวเส้นต่อเนื่องกัน ปิดล้อมพื้นที่ใดพื้นที่หนึ่งทำให้เกิดรูปร่าง หรือกลุ่มของจุดและเส้นซึ่งเรามองผ่านไปแล้วเกิดลักษณะของระนาบ

จากการศึกษาการวาดภาพลายเส้นยังอีกพบว่าการวาดภาพลายเส้นของสถาปนิกนั้นจะมีลักษณะเด่นกว่าการวาดภาพลายเส้นของจิตกรหรือนักวิชาชีพอื่นที่เกี่ยวข้อง คือจะเน้นที่การวาดภาพลายเส้นแบบเรขาคณิตเป็นหลัก (Frank Ching, 1985: 27) โดยในการวาดภาพลายเส้นขั้นพื้นฐานของสถาปนิกจะแบ่งเป็นลักษณะต่างๆ เช่น

การวาดเส้นแบบ Parallel Lines เป็นการวาดเส้นในแนวตั้งและแนวนอนโดยวาดเว้นระยะห่างประมาณหนึ่ง วาดเป็นเส้นให้ขนานกัน

การวาดเส้นแบบ Random Lines เป็นการวาดเส้นเชื่อมจุดสองจุด เข้าด้วยกัน

การวาดเส้นแบบ Circular Spirals เป็นการวาดเส้นจากจุดศูนย์กลางออกมาเป็นวงกลม

จากตัวอย่างประเภทของการวาดเส้นข้างต้นจะทำให้เราสามารถมองเห็นกรอบของการพัฒนาโปรแกรมว่าจะ ต้องมีการพัฒนาโปรแกรมในส่วนของการวาดเส้นให้รองรับการวาดเส้นลักษณะใดบ้าง อีกทั้งทำให้เราสามารถเลือกโครงสร้างการเขียนโปรแกรมให้เหมาะสมกับการวาดเส้นบางประเภทได้

2.2 **ทฤษฎีการสร้างรูปทรง 3 มิติ** (ปุณณรัตน์ พิชญไพบูลย์, 2542)

โดยทั่วไปแล้วการสร้างสรรค์วัตถุ 3 มิติ ด้วยคอมพิวเตอร์กราฟิก อาจเปรียบได้กับ ดินเหนียวและทรายเมื่อมารวมกันเข้าอย่างมีสัดส่วนก็เกิดเป็นเนื้อดินที่ใช้ปั้น การที่ประติมากรหยิบก้อนดินชิ้นหนึ่งขึ้นมาปั้นเป็นรูปประติมากรรม ก็เช่นเดียวกับการนำจุดมาวางเรียงต่อกันอย่างมีระเบียบจนเกิดเป็นเส้น และเมื่อเส้นถูกต่อเข้าตามเงื่อนไขในช่องว่างก็เกิดเป็นรูประนาบแบนขึ้นมา ซึ่งเป็นพื้นฐานไปสู่วัตถุ 3 มิติต่อไป จึงอาจกล่าวได้ว่าการสร้างรูปทรง 3 มิติเป็นกระบวนการที่ผู้สร้างใช้ประสบการณ์ตรงของการสัมผัสหยิบจับองค์ประกอบย่อยมารวมเข้าด้วยกันจนเกิดเป็นวัตถุชิ้นใหม่ที่มีตัวตนและเคลื่อนย้ายได้ สำหรับในระบบคอมพิวเตอร์กราฟิก 3 มิติ กระบวนการดังกล่าวเป็นการสร้างวัตถุ 3 มิติที่เลียนแบบสภาวะจริง โดยจำกัดการแสดงผลอยู่แค่ช่องมองที่เป็นกรอบของจอภาพ ผู้สร้างสรรค์จำเป็นต้องอาศัยประสบการณ์จริงเข้ามามีส่วนช่วยอย่างยิ่งในการคิดพิจารณาเงื่อนไข เพื่อการจำลองสภาวะต่างๆ ให้เหมือนจริง

ซึ่งกระบวนการสร้างรูปทรง 3 มิติ จะมีรายละเอียดดังต่อไปนี้

2.2.1 ธรรมชาติของการขึ้นรูปวัตถุ 3 มิติ

2.2.2 การสร้างภาพ 3 มิติ

2.2.3 วิธีการสร้างรูปทรง 3 มิติ

2.2.1 **ธรรมชาติของการขึ้นรูปวัตถุ 3 มิติ**

รูปทรงที่ปรากฏในคอมพิวเตอร์กราฟิก ไม่ว่าจะเป็นรูปทรงที่เราคุ้นเคยในชีวิตประจำวัน เช่น รถยนต์ เครื่องบิน อาคารสถาปัตยกรรม ฯลฯ หรือรูปทรงที่เป็นนามธรรม

เช่น อะตอม โมเลกุล และยังรวมถึงภาพจากจินตนาการ รูปที่กล่าวมานี้เป็นข้อมูลที่ถูกเก็บบันทึกไว้ในข้อมูลเชิงตัวเลข ซึ่งเมื่อข้อมูลเหล่านั้นถูกนำมาประมวลผลโดยซอฟท์แวร์และฮาร์ดแวร์จะปรากฏเป็นรูปทรงขึ้นมา การสร้างรูปวัตถุบางอย่าง เช่นการสร้างจุดดาวจำนวนมากๆ ซึ่งเป็นผลจากการใช้ค่าตัวแปรที่ได้จากการสุ่มเพื่อนำมาคำนวณวางตำแหน่งให้ปรากฏไปบนพื้นที่ว่างก็ได้

วัตถุหรือรูปทรงที่ปรากฏในคอมพิวเตอร์กราฟิก 3 มิติมีวิธีสร้างขึ้นมาได้หลายวิธี นับตั้งแต่การป้อนค่าพิกัดทีละสองระนาบ (XY, XZ, ZY) หรือการใช้อุปกรณ์รอบข้าง 3 มิติที่สามารถอ่านตำแหน่งพิกัดบนระนาบทั้งสาม (X,Y,Z)

ในสิ่งแวดล้อมรอบตัวเราล้วนเป็นวัตถุ 3 มิติ เช่น ห้องเรียนห้องหนึ่ง ประกอบด้วยความยาว ความกว้าง และความสูง สิ่งที่อยู่ภายในห้องประกอบด้วย โต๊ะ เก้าอี้ กระดานดำ สิ่งหนึ่งที่เป็นคุณสมบัติในตัววัตถุ 3 มิติเหล่านี้คือ การมีคุณลักษณะที่เป็นเอกลักษณ์ของตัวเอง เช่น สี พื้นผิว ความกว้าง ความสูง วัตถุแต่ละตัวจึงต้องมีการแบ่งจำแนกประเภทรวมทั้งการตั้งชื่อกำกับ ซึ่งอาจเป็นการกำหนดหมายเลขรหัสตัวอักษร ในคอมพิวเตอร์กรฟิกเราจะพบว่า ค่าพิกัดนอกจากจะใช้เป็นเครื่องแสดงตำแหน่งจุด (X,Y) ในระบบคาร์ทีเซียนแล้ว ยังเป็นชื่อเรียกจุดได้อีกด้วย ดังนั้นวัตถุ 3 มิติทุกชิ้นจำเป็นต้องมีชื่อเรียกเพื่อใช้จำแนกสิ่งต่างๆ ที่ปรากฏบนที่ว่างได้อย่างถูกต้อง ชื่อจึงเป็นส่วนหนึ่งของ ลักษณะประจำ (Attribute) ซึ่งเป็นสิ่งที่แสดงเอกลักษณ์ของวัตถุแต่ละตัว ลักษณะประจำนี้ได้แก่ ชื่อ สี พื้นผิว ขนาด น้ำหนัก กลิ่น เป็นต้น

2.2.2 การสร้างภาพ 3 มิติ

วัสดุประเภทต่างๆ มีวิธีนำมาขึ้นรูปที่แตกต่างกันออกไป รูป 3 มิติในคอมพิวเตอร์กราฟิกก็เป็นวัตถุที่มีปริมาตรเช่นเดียวกับวัตถุจริง จะแตกต่างกันก็ตรงที่เป็นภาพ 3 มิติที่ให้การรับรู้เหมือนวัตถุจริงเท่านั้น การเปลี่ยนตำแหน่งมุมมองและการแสดงผลด้านต่างๆ ของรูปวัตถุ 3 มิติจะใช้การแปลงรูป (Transforming) บนระนาบ 2 มิติ การมองภาพ 3 มิตินี้บางครั้งทำให้ผู้ชมเกิดความสับสนได้ ดังจะพบได้ใน รูปโครงลวด (Wire frame) ซึ่งเป็นการแสดงผลด้วยเส้นเวกเตอร์ที่เชื่อมต่อกันระหว่างจุดยอด (Vertex) เกิดเป็นภาพ 3 มิติที่คอมพิวเตอร์สามารถแสดงผลได้เร็วที่สุด แต่มีจุดอ่อนที่สร้างความกำกวมแก่ผู้ชมได้ เนื่องจากยากต่อการแยกแยะมิติ ส่วนรูป 3 มิติที่ผู้ชมรับรู้ได้ง่ายที่สุดคือภาพไล่สีน้ำหนักแสงเงาโดยเลียนแบบ

ลักษณะของวัตถุในธรรมชาติ สิ่งหนึ่งที่จำเป็นสำหรับผู้สร้างภาพ 3 มิติคือ การฝึกฝนวิธีมองภาพโครงลวดที่ปรากฏบนพื้นระนาบแบน ด้วยทักษะที่เรียกว่า มิติสัมพันธ์ (Spatial relationship) ซึ่งเป็นทักษะอย่างหนึ่งของการรับรู้ความสัมพันธ์ระหว่างช่องว่างกับรูปวัตถุ

2.2.3 วิธีการสร้างรูปทรง 3 มิติ

รูปทรง 3 มิติถูกสร้างขึ้นด้วยการนำตัวแปรซึ่งเป็นตัวเลขมาประมวลผลโดยกระบวนการทางเรขาคณิตและตรีโกณมิติ ตัวแปรเสริมหรือค่าพารามิเตอร์เหล่านี้จะถูกส่งเข้าไปโดยอาศัยอุปกรณ์นำเข้าข้อมูลที่หลากหลาย ตั้งแต่อุปกรณ์ที่ป้อนข้อมูลเป็นอักขระและตัวเลข เช่น แป้นพิมพ์ อุปกรณ์ที่แปลงการเคลื่อนที่ให้กลายเป็นข้อมูลตัวเลข เช่น เมาส์ กระดานไฟฟ้า เป็นต้น จากนั้นเรายังสามารถนำรูป 2 มิติมาปรับปรุงให้เป็นรูป 3 มิติได้ด้วยกระบวนการที่มีในซอฟต์แวร์ 3 มิติเช่น การเพิ่มความหนาของวัตถุทำให้เกิดมีปริมาตรขึ้นมา นอกจากนี้ยังรวมถึงอุปกรณ์วัดระยะโดยเจาะจงถึงตำแหน่งพิกัดของจุด เซล ซึ่งเป็นหน่วยที่เล็กที่สุดในวัตถุ 3 มิติ วิธีหนึ่งที่เป็นพื้นฐานการสร้างรูป 3 มิติคือ การสร้างรูปทรงเรขาคณิตเบื้องต้น เช่น ทรงกระบอก กล่อง พีระมิด กรวย โดนัท เนื่องจากรูปทรงเหล่านี้เป็นรูปทรงพื้นฐานและเป็นแก่นเค้าโครงของวัตถุ 3 มิติส่วนใหญ่ เราจึงเรียกรูปทรงประเภทนี้ว่า รูปทรงปฐมฐาน (Primitive shape) โปรแกรมที่ใช้สร้างรูปทรงเหล่านี้จะบรรจุฟังก์ชันคณิตศาสตร์และตรีโกณมิติ ที่ต้องการเพียงพารามิเตอร์เพื่อนำไปคำนวณสร้างเป็นรูปวัตถุ

การสร้างรูปทรง 3 มิติ มีวิธีการสร้างหลักๆ 2 วิธีได้แก่

- การสร้างรูปทรง 3 มิติจากตัวเลข
- การสร้างรูปทรง 3 มิติจากโพลีกอน

- <u>การสร้างรูปทรง 3 มิติจากตัวเลข</u> เป็นวิธีการสร้างรูปทรง 3 มิติที่ง่ายที่สุดคือการกำหนดตำแหน่งจุดด้วยการใช้แป้นพิมพ์ ซึ่งเป็นวิธีการที่เหมาะที่สุดในกรณีที่ต้องการสร้างรูปวัตถุปฐมฐานหลาย ๆ รูปที่มีขนาดเท่ากัน หรือแตกต่างกันเล็กน้อยเพื่อนำมารวมเข้าด้วยกันหรือตัดทอนกันให้เกิดเป็นรูป 3 มิติที่มีความซับซ้อนขึ้น นอกจากนี้ยังเหมาะกับการสร้างรูป 3 มิติที่ต้องการความละเอียดและเที่ยงตรงในระดับจุดทศนิยม ซึ่งการใช้อุปกรณ์ลากเส้นไม่สามารถทำได้ดีนัก ตัวอย่างเช่น การขึ้นรูปปฐมฐานรูปกล่องจะมีเพียงตำแหน่งของจุดพิกัดที่ต้องการเพียง 2 จุด นั่นคือการลากเส้นตรงจากจุดเริ่มต้นเป็นแนวทแยงไปสู่จุดที่สอง ตำแหน่ง

ทั้งสองเป็นค่าตัวแปรที่ฟังก์ชั่นรูปกล่องต้องการ ผู้ใช้เพียงป้อนค่าทั้งสองลงในฟังก์ชันก็เพียงพอในการสร้างรูปทรง 3 มิติ

- <u>การสร้างรูปทรง 3 มิติจากโพลีกอน</u> ความหมายของโพลีกอนตามนิยามของ James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes, 2003 คือการต่อเส้นหลายเส้นเข้าด้วยกันโดยมีจุดของเส้น (Vertices) ต่อกันทำให้เกิดขอบเขต (Edges) ของระนาบ และเมื่อมีการเชื่อมต่อขอบเขตของระนาบเข้าด้วยกันทั้งหมดจนเกิดรูปปิด (Closed sequence of edges) จะเรียกว่าโพลีกอน โพลีกอนอาจนำมาสร้างเป็นรูปทรง 3 มิติได้ด้วยวิธีการกวาดโพลีกอน (Sweep Operation)

การกวาดโพลีกอน (Sweep Operation) เกิดจากการพิจารณาวัตถุรอบตัวเรา ซึ่งพบว่าวัตถุจะมีความหนาเกิดขึ้นจากการเรียงซ้อนกันของชั้นบางๆ ของวัสดุหลายๆ ชั้นเข้าด้วยกัน ซึ่งวิธีการนี้เป็นการนำวัตถุแบนๆ มาเรียงซ้อนกันเป็นชั้น ซึ่งถ้าเส้นทางที่กวาดไปนี้เป็นเส้นตรงก็จะได้วัตถุแท่งยาวๆ หรือถ้าเส้นทางนี้เป็นวงกลมรอบจุดศูนย์กลางก็จะได้รูปทรงกระบอก วิธีการนี้เมื่อนำมาใช้กับการสร้างรูปทรง 3 มิติ ก็จะเรียกว่าการกวาดโพลีกอนนั่นเอง รูปทรงของวัตถุที่ได้จากวิธีการกวาด จะขึ้นกับองค์ประกอบต่อไปนี้

เส้นทางเดิน (Path) เป็นเส้นตรงที่โยงระหว่างจุดยอดสองจุด เมื่อนำโพลีกอนมาวางที่ตำแหน่งจุดยอดทั้งสองนี้ก็จะสามารถโยงจุดต่างๆ บนเส้นขอบของโพลีกอนทั้งสองถึงกันเป็นความหนาของวัตถุขึ้นมา ทำให้เกิดเป็นรูปทรงแท่ง และในกรณีที่เส้นทางเดินถูกกำหนดเป็นเส้นคดเคี้ยวรูปวัตถุก็จะเป็นไปตามรูปของเส้นทางเดินนั้น

การยืด (Extrusion) เป็นเทคนิคเพิ่มความหนาหรือความลึกโดยการยืดออกจากเส้นรอบโพลีกอน ทำให้โพลีกอนเกิดมีปริมาตรขึ้น เช่นเดียวกับยาสีฟันที่ถูกบีบออกมาจากหลอดเป็นลำยาว โดยระยะความลึกของรูปโพลีกอนจะกลายเป็นด้านๆ หนึ่งหรือความหนาของวัตถุไป ในโปรแกรมสร้างภาพ 3 มิติจะใช้การสำเนารูปโพลีกอนด้านหน้าไปวางปิดท้าย แล้วจึงโยงเส้นระหว่างรูปโพลีกอนเข้าด้วยกันเกิดเป็นวัตถุ 3 มิติขึ้นมา

ในการสร้างภาพ 3 มิติด้วยการยืด จะเริ่มจากการสร้างรูปโพลีกอนจากอุปกรณ์นำเข้าข้อมูล เช่น เมาส์ หรือ ปากกาอิเล็กทรอนิกส์ แล้วนำรูปโพลีกอนที่ได้วางเรียงกันเชื่อมต่อเป็นส่วนเดียวกัน โดยอาศัยการประมวลผลของโปรแกรมกราฟิก จากนั้นจึงมีการสร้างระนาบเหลี่ยมหรือหน้าเหลี่ยม (Face) เชื่อมระหว่างรูปโพลีกอน

ด้านหน้าและด้านหลัง จนกลายเป็นวัตถุชิ้นเดียวกันทั้งหมด อย่างไรก็ดีหาต้องการส่วนโค้งมน โปรแกรมจะต้องทำการสร้างรูประนาบเหลี่ยมให้เล็กลงอีก ซึ่งจะต้องคำนวณสร้างระนาบเหลี่ยมมากขึ้นตามลำดับ

2.3 ขั้นตอนวิธีทางคอมพิวเตอร์ในการรับรู้ภาพวาดลายเส้น

ในการรับรู้ภาพวาดลายเส้นจะต้องอาศัยทฤษฎีทางคณิตศาสตร์และข้อมูลจากงานวิจัยที่เกี่ยวข้องเพื่อนำมาเป็นแนวทางในการพัฒนาขั้นตอนวิธีที่เหมาะสมกับการรับรู้ภาพวาดลายเส้น โดยสามารถแบ่งขั้นตอนวิธีที่เกี่ยวข้องได้ดังนี้

2.3.1 การรับรู้ภาพกราฟิก

2.3.2 การวิเคราะห์มุมจากภาพวาดลายเส้น

2.3.3 การวิเคราะห์รูปแบบที่เหมือนกันของภาพวาดลายเส้น

2.3.4 การทดสอบความขนานกันของเส้น

2.3.5 การทดสอบความยาวของเส้น

2.3.6 การวาดเส้นตรงแบบ ดีดีเอ

2.3.7 การหาระยะห่างระหว่างจุดภาพ

2.3.8 การหาเส้นขอบบนวัตถุ

2.3.9 การค้นหามุมของวัตถุในภาพ

2.3.10 การสร้างรูปวงกลม

2.3.11 ผลกระทบของการใช้คอมพิวเตอร์ในการออกแบบขั้นต้น

2.3.12 การเก็บข้อมูลแบบเว็กเตอร์และแรสเตอร์

2.3.1 การรับรู้ภาพกราฟิก (Jin Xiangyu, Liu Wenyin, Sun Jianyong, Zhengxing Sun: 2002)

วิธีการรับรู้ภาพกราฟิกเป็นความรู้ที่มีขอบเขตครอบคลุมความรู้มากมายหลายแขนง แต่สำหรับด้านที่เกี่ยวข้องกับการพัฒนาโปรแกรมรับรู้วัตถุจากภาพวาดลายเส้นนี้ จะเกี่ยวข้องกับการรับรู้ภาพกราฟิกในเรื่องการจำแนกรูปร่าง หรือ Shape Recognition

การจำแนกรูปร่าง ประกอบด้วยการแก้ปัญหาสองส่วนคือ ส่วน ป้อนข้อมูล ส่วนแสดงผล

ส่วนป้อนข้อมูล ได้แก่ การแก้ปัญหาการ Stroke รูปร่างแบบปิด Closed-shape

ส่วนแสดงผล ได้แก่ การแบ่งประเภทของรูปร่างพื้นฐาน เช่น รูปสามเหลี่ยม รูปสี่เหลี่ยม และรูปหลายเหลี่ยม เป็นต้น

ในการแก้ปัญหาทั้งสองส่วนนี้ได้มีทฤษฎีในการแก้ปัญหา แบ่งออกได้ 4 ประเภทคือ

- Off-line Approaches
- Filter-based Approaches
- Energy Minimization Approaches
- Machine Learning based Approaches

- *Off-line Approaches*

เป็นการทำงานโดยอาศัยการตัดสินใจแบบ Tree โดยอาศัยขั้นตอนการทำงานดังนี้

1. แยกส่วนที่เป็นเส้นตรงออกจากรูปภาพวาดลายเส้นที่วาดเป็นเส้นเรขาคณิต
2. ใช้กฎการตัดสินใจแบบ Tree โดยแต่ละสาขาของ Tree จะเป็นตัวแทนของรูปร่างแต่ละแบบเช่น สามเหลี่ยม หรือ วงกลม เป็นต้น

ยกตัวอย่างเช่น หากมีการจำแนกได้ว่ามีส่วนที่เป็นเส้นตรง 3 เส้น ก็แสดงว่าภาพวาดลายเส้นนั้นเป็นรูปร่างของสามเหลี่ยม

- *Filter-based Approaches*

เป็นการทำงานโดยอาศัยตัวกรอง (filter) เป็นตัวจำแนกรูปร่าง โดยการวาดภาพลายเส้น ให้ขนานกับแกน X และแกน Y จากนั้นตัวกรอง จะทำหน้าที่แปลงภาพวาดลายเส้นให้เหลือแต่เฉพาะเส้นที่ขนานกับแกนดังกล่าวและจำแนกเป็นรูปร่างเพื่อแสดงผลลัพธ์ต่อไป

- *Energy Minimization Approaches*

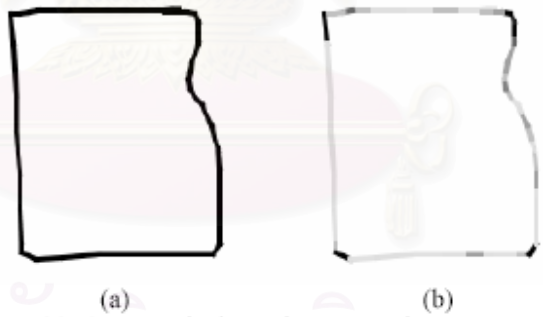
เป็นการทำงานที่อาศัยการเปรียบเทียบภาพวาดลายเส้นกับสมมติฐานของรูปร่าง โดยทุกๆครั้งที่มีการวาดภาพลายเส้น จะมีการตั้งสมมติฐานของรูปร่างว่าเป็นสามเหลี่ยม สี่เหลี่ยม หรือวงรี แล้วจึงทำการคำนวณหาความแตกต่างระหว่างภาพวาดลายเส้นกับรูปร่างที่เป็นสมมติฐาน และเพื่อให้สามารถเปรียบเทียบหาความแตกต่างได้รูปร่างที่เป็นสมมติฐานจะถูกย่อขยาย หรือหมุนเพื่อให้มีความใกล้เคียงกับภาพวาดลายเส้นมากที่สุด ด้วยการวัดระยะห่างระหว่างจุดยอดของรูปร่างสมมติฐานกับเส้นของภาพวาดลายเส้นเรียกว่า energy และถ้ารูปร่างใดมี energy น้อยที่สุดก็จะเลือกเอารูปร่างนั้นมาแสดงผล

- *Machine Learning based Approaches*

เป็นการทำงานที่อาศัยการคำนวณทางสถิติเรื่อง Machine-Learning based approaches ซึ่งใช้พื้นฐานของหลักการด้าน neural networks และการคำนวณทางสถิติเรื่อง support vector machine ซึ่งในการทำงานนั้นให้ผลถูกต้องถึง 95.8% แต่เนื่องจากต้องอาศัยประสิทธิภาพการทำงานของเครื่องสูงและราคาในการปฏิบัติงานสูง วิธีการนี้จึงไม่ได้รับความนิยมกับงานวิจัยทางด้านนี้เท่าที่ควรเพราะไม่สามารถนำไปประยุกต์ใช้กับอุปกรณ์ที่มีความสามารถในการคำนวณต่ำเช่น PDA ได้

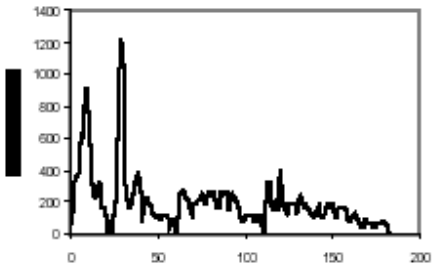
2.3.2 การวิเคราะห์มุมจากการวาดภาพลายเส้น

การรับรูปรูปร่างหลายเหลี่ยมจะช่วยให้โปรแกรมสามารถจำแนกรูปร่างได้หลากหลายมากขึ้น นอกจากการจำแนกเส้นตรงและเส้นโค้ง โดยโปรแกรมจะสามารถรับรูปรู้ รูปสามเหลี่ยม สี่เหลี่ยม หรือหกเหลี่ยม ได้ และในการรับรูปรูปหลายเหลี่ยมนั้นได้มีการวิจัยการวาดภาพลายเส้นว่าจะสามารถจำแนกรูปหลายเหลี่ยมจากภาพวาดลายเส้นได้อย่างไร โดยผลการวิจัยได้ค้นพบว่าวิธีการหนึ่งที่จะจำแนกได้ว่าผู้ใช้งานกำลังวาดรูปหลายเหลี่ยมอยู่ได้แก่ การวิเคราะห์หามุมของการวาดภาพลายเส้น (Peter Agar, Keven Novins: 2003)



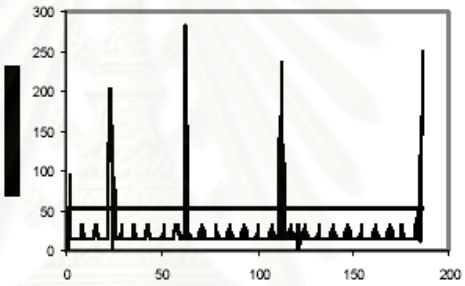
รูปภาพ 2.2 แสดงภาพที่ผู้ใช้งานวาดขึ้น (a) กับภาพที่คอมพิวเตอร์เก็บข้อมูล (b)

จากรูปภาพ 2.2 รูป a เป็นภาพวาดลายเส้นที่ผู้ใช้งานวาดขึ้น รูป b เป็นภาพที่สร้างขึ้นเพื่อแสดงให้เห็นถึงความเข็มของเส้นที่เพิ่มขึ้นในบางจุดเมื่อเทียบกับเวลาในการวาดทั้งหมด จะเห็นได้ว่าผู้ใช้งานส่วนใหญ่จะใช้เวลาในการวาดภาพที่จุดมุมของการวาดนานกว่าส่วนอื่นๆ



รูปภาพ 2.3 กราฟแสดงความเร็วและเวลาของการวาดภาพสี่เหลี่ยม

จากรูปภาพ 2.3 เป็นกราฟแสดงความเร็วของการวาดภาพรูปสี่เหลี่ยม ซึ่งแสดงให้เห็นถึงความเร็วในการวาดในแกนตั้งจะลดลงสี่ครั้งจากการลาดเส้นเพื่อวาดรูปสี่เหลี่ยม แต่จากกราฟยังพบอีกว่าวิธีการนี้มีความไม่แน่นอนของข้อมูล (Noise) อยู่มาก



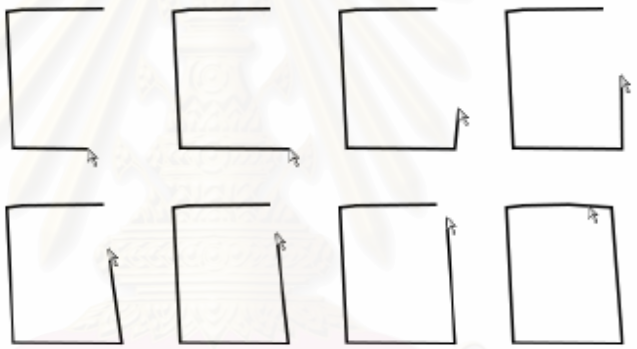
รูปภาพ 2.4 กราฟแสดงความแตกต่างของเวลาในการลากเส้นแต่ละจุด

จากรูปภาพ 2.4 เป็นกราฟแสดงการวิเคราะห์การวาดเส้นอีกแบบ โดยใช้ข้อมูลการวาดภาพเหมือนกราฟที่ผ่านมาแต่เน้นการวัดความแตกต่างของเวลาในการวาดเส้นจะเห็นว่ามีสี่จุดที่ใช้เวลาในการวาดมากกว่าจุดอื่นๆ ในภาพ

จากการวิเคราะห์กราฟทำให้เกิดแนวคิดในการหาจุดมุมของการวาดภาพลายเส้นสองแนวทาง โดยจะใช้เวลา (time-stamp) กับความเร็วมาเป็นตัววัดในแต่ละจุดที่วาด และอีกแนวทางหนึ่งจะใช้ความแตกต่างของเวลาที่ใช้ในการวาดจุดแต่ละจุดจากการลากเมาส์มาเป็นตัววัด โดยจะเป็นการวิเคราะห์ภาพวาดลายเส้นแบบ Real-time และจากการศึกษาการวาดภาพลายเส้นพบว่า ในการวาดภาพลายเส้นนั้นผู้วาดแต่ละคนจะมีความเร็วในการวาดภาพแตกต่างกัน จึงทำให้ไม่สามารถหาค่าเฉลี่ยที่ชัดเจนสำหรับวัดความเร็วในการวาดภาพได้

แนวคิดเรื่องการใช้ความแตกต่างของเวลาในการวาดภาพลายเส้นจึงเป็นหัวใจสำคัญที่จะช่วยให้สามารถวิเคราะห์จุดที่เป็นมุมของรูปหลายเหลี่ยมได้

จากแนวคิดเรื่องความแตกต่างของเวลาในการวิเคราะห์หามุมในรูปภาพ นำไปสู่ขั้นตอนวิธี (Algorithm) สำหรับการวิเคราะห์ โดยกำหนดให้มีการวิเคราะห์ภาพทุกๆ 30 ms เริ่มต้นเมื่อผู้ใช้เริ่มลากเมาส์เพื่อวาดภาพจะมีการสร้างเส้นลากไปยังจุดที่เมาส์อยู่ และเมื่อถึงจุดที่เมาส์ไม่มีการเคลื่อนไหวโปรแกรมจะทำการวิเคราะห์ว่าจุดนี้คือจุดที่เป็นมุมของภาพวาดลายเส้นหรือไม่ ด้วยวิธีการวิเคราะห์หาความแตกต่างของเวลาระหว่างจุดที่มีการเคลื่อนที่ของเมาส์จุดสุดท้ายกับจุดที่เมาส์หยุด และเมื่อพบว่าเวลาระหว่างนี้มากกว่าค่าเฉลี่ยทั้งหมดของเวลาที่วัดได้ในการลากเส้นทั้งหมดคูณกับค่าคงที่ (ศึกษาและกำหนดให้เป็น 3.0) จะมีการกำหนดให้จุดนี้เป็นมุมของภาพ



รูปภาพ 2.5 แสดงการวาดภาพอย่างต่อเนื่องในขณะที่โปรแกรมจะทำการวิเคราะห์หาจุดที่เป็นมุมของภาพ

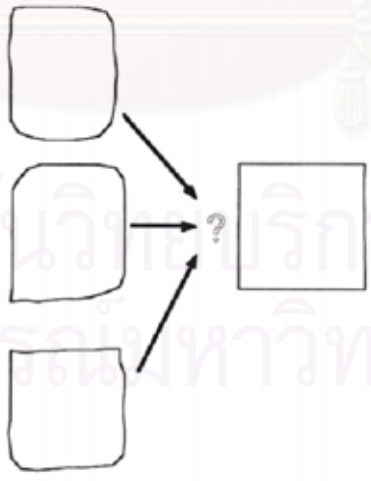
2.3.3 **การวิเคราะห์รูปแบบที่เหมือนกันของภาพวาดลายเส้น** (Sketch patterns recognition)

การวาดภาพลายเส้นนั้นจะมีรูปแบบและวิธีการวาดที่หลากหลายตามแต่ความถนัดและประสบการณ์ของผู้วาดแต่ละคน ในการวิเคราะห์ภาพวาดลายเส้นเพื่อหาวิธีการในการรับรู้ข้อมูลที่ถูกต้องจึงต้องอาศัยการศึกษาลักษณะการวาดภาพและรูปแบบ (Pattern) ที่สามารถนำมาใช้เป็นหลักเกณฑ์ในการวิเคราะห์เพื่อจำแนกภาพวาดลายเส้นที่สื่อความหมายตรงกับความต้องการของผู้วาดได้

รูปภาพ 2.6 แสดงการวาดภาพลายเส้นทางสถาปัตยกรรมอย่างอิสระ

ได้มีผู้ศึกษาการวิเคราะห์วาดภาพลายเส้นทางสถาปัตยกรรมเอาไว้ (Fatos Bengi Durgun, Bulent Ozguc, 1990) เพื่อเก็บรวบรวมข้อมูลที่ครอบคลุมการวาดภาพลายเส้นทั้งหมด โดยใช้แบบสอบถามไปยังสถาปนิก 85 คน โดยมีบางส่วนเป็นนักศึกษาสถาปัตยกรรมชั้นปีที่สี่

ซึ่งในการทำแบบสอบถามมีการใช้ทั้งวิธีการสัมภาษณ์และใช้ภาพวาดสื่อความหมาย เพราะการใช้ภาพเพียงอย่างเดียวอาจทำให้มีการตีความหมายภาพวาดไม่ตรงกับความต้องการของผู้วาดก็เป็นได้



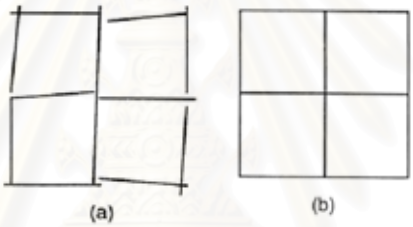
รูปภาพ 2.7 ภาพแสดงลักษณะการวาดภาพลายเส้น ที่คอมพิวเตอร์ต้องตีความหมายให้ถูกต้อง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ระหว่างการทำแบบสอบถามผู้เข้าทดสอบจะต้องทำการวาดภาพให้ตรงกับรูปร่างที่แสดงเป็นตัวอย่างให้ดู โดยขั้นแรกเป็นภาพเรขาคณิตเบื้องต้นเช่น สี่เหลี่ยม สามเหลี่ยม วงกลม จากนั้นจะทำการทดสอบโดยให้วาดภาพงานสถาปัตยกรรมที่ทุกคนเคยเห็นมาก่อน และสุดท้ายจะเป็นการวาดภาพผังของมหาวิทยาลัย

จากนั้นจะเป็นการสัมภาษณ์ถึงความหมายของภาพที่วาดออกมา และความตั้งใจของผู้วาด นอกจากนี้ระหว่างการวาดภาพจะสามารถเก็บข้อมูลลักษณะการวาดภาพ ตำแหน่งที่เริ่มวาดภาพ และวิธีในการพัฒนาการวาดแผนผังอีกด้วย

หลังจากรวบรวมข้อมูลทั้งหมดแล้ว จึงทำการประเมินเพื่อหารูปแบบที่เหมือนกัน โดยพบว่ามีสองหัวข้อที่ถูกใช้มากที่สุดในการวาดภาพลายเส้น คือ การทดสอบความขนานกันของเส้น และการทดสอบความยาวของเส้น ซึ่งจะเป็นตัวบ่งชี้ว่าเส้นที่วาดนั้นเป็นเส้นเดียวกันหรือไม่ หรือเส้นที่วาดนั้นเป็นการเว้นว่างสำหรับช่องเปิดหรือไม่



รูปภาพ 2.8 การเปรียบเทียบความขนานกันของเส้น จากภาพวาดลายเส้น (a) มาสู่การปรับให้ขนานกัน (b)

2.3.4 **การทดสอบความขนานกันของเส้น** คือการหาค่าความชันของเส้นสองเส้นที่จะเป็นตัวกำหนด ความขนานกันของเส้นทั้งหมด โดยสามารถนำไปทดสอบการทับกันของเส้นหรือการตัดกันของเส้นเป็นต้น จากการทดสอบพบว่า ค่าความขนานกันสามารถเขียนให้อยู่ในรูปแบบ

$$Pi\%*Ms = Ma \pm \Delta Pi$$

P คือ ค่าจากการทดสอบ

Ms คือ ค่าความชันมาตรฐาน

Ma คือ ค่าความชันที่ยอมรับได้

ΔP คือ หน่วยของความผกผัน

i คือ ประเภทในการอ้างอิง (Fatos Bengi Durgun, Bulent Ozguc, 1990: page 6)

2.3.5 **การทดสอบความยาวของเส้น** คือการหาค่าระยะต่อของเส้น (extend) หรือระยะที่ขาดไป(missing) ของเส้นเมื่อเทียบกับความยาวทั้งหมด โดยสามารถนำไปหาความยาวที่เหมาะสมของเส้น ที่ผู้วาดต้องการวาดจริงๆ ได้ จากการทดสอบพบว่า ค่าความยาวของเส้นสามารถเขียนให้อยู่ในรูปแบบ

$$Li\%*Ds = Da \pm \Delta Li$$

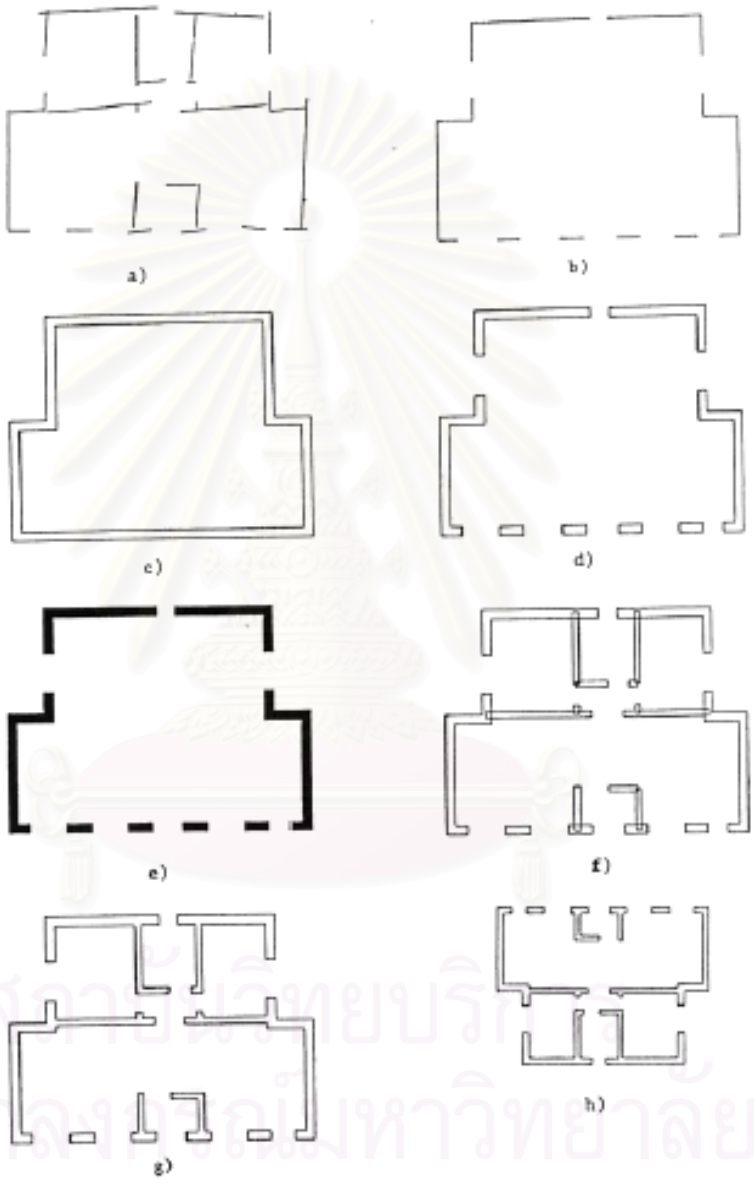
Li คือ ค่าจากการทดสอบ
Ds คือ ค่าความยาวมาตรฐานของเส้น
Da คือ ค่าความยาวของเส้นที่ยอมรับได้
ΔLi คือ หน่วยของความยาวผกผัน

การทดสอบความยาวช่วยจำแนกเส้นที่ตั้งใจหรือไม่ตั้งใจจะวาด เช่น หากความยาวของเส้นมีค่ามากกว่า Da จะนับว่าเป็นเส้นที่ตั้งใจวาด และในทางกลับกันหากความยาวของเส้นมีค่าน้อยกว่า Da จะถือว่าเป็นเส้นที่ไม่ได้ตั้งใจวาด
จากการประเมินได้ข้อสรุปดังนี้

- วิธีการวาดภาพที่พบมากที่สุด คือ การวาดจากขอบนอกเข้ามาสู่ภายใน โดยเริ่มจากวาดเส้นรอบรูปของแผนผังจากนั้นจึงเติมเส้นแบ่งเป็นผนังภายใน และสุดท้ายคือการตกแต่งรายละเอียดให้เสร็จ
- มีการใช้ เส้น เป็นองค์ประกอบของภาพมากที่สุดเมื่อเทียบกับองค์ประกอบอื่นๆ เช่น จุด เส้นโค้ง การแรเงา เป็นต้น โดยที่เส้นพบโดยมากจะเป็นเส้นตรงซึ่งมีความเบี่ยงไปมาตลอดความยาวของเส้น และกลุ่มของเส้นอาจมีลักษณะลากมาตัดกันหรือวาดให้ปลายซ้อนทับกันได้
- มีข้อสรุปเกี่ยวกับคุณลักษณะของเส้นที่พบเป็นส่วนใหญ่ คือ จะพบค่าผกผันซึ่งต่ำกว่า P% ในการทดสอบการขนานกับแนวตั้ง การขนาดกับแนวนอน การซ้อนทับหรือการขนานกัน โดยเป็นค่าที่เกิดขึ้นอย่างไม่ตั้งใจหรืออาจมองได้ว่าเป็นผลลัพธ์จากการวาดแบบหยาบๆ

- ข้อสรุปสุดท้ายคือ พบวิธีการหาความแตกต่างของการวาดช่องเปิดกับการวาดโดยไม่ได้ตั้งใจให้มีช่องว่างเอาไว้ โดยแนวความคิดการหาความแตกต่างได้มาจากข้อมูลจากแบบทดสอบ ซึ่งพบว่าสถาปนิกส่วนใหญ่จะมีการวาดช่องเปิดอย่างระมัดระวัง

เพื่อให้ขอบเขตอยู่ในระยะที่ต้องการ นั่นคือ หากมีการวาดให้เว้นช่องเป็นระยะมากกว่าค่า D% (อ้างจากสูตรการทดสอบความยาวของเส้น) จะถือว่าเป็นการตั้งใจจะเว้นช่องเปิด หากน้อยกว่านี้จะถือว่าไม่ใช่การเว้นช่องเปิดนั่นเอง



รูปภาพ 2.9 การแปลงภาพวาดลายเส้น โดยใช้ข้อสรุปจากการศึกษาเป็นเกณฑ์ในการแปลงภาพ โดยเริ่มแปลงภาพจากขอบนอกก่อนแล้วจึงสร้างช่องเปิดและแบ่งพื้นที่ภายใน

2.3.6 การวาดเส้นตรงแบบดีดีเอ (กิตติมา มะแก้ว, 2538: 12)

การใช้วิธีการวาดเส้นตรงแบบดีดีเอ (Digital Differential Analyzer, DDA) เป็นวิธีการวาดเส้นตรงที่ง่ายและสะดวก เป็นวิธีการพื้นฐานที่นิยมใช้กันมาก

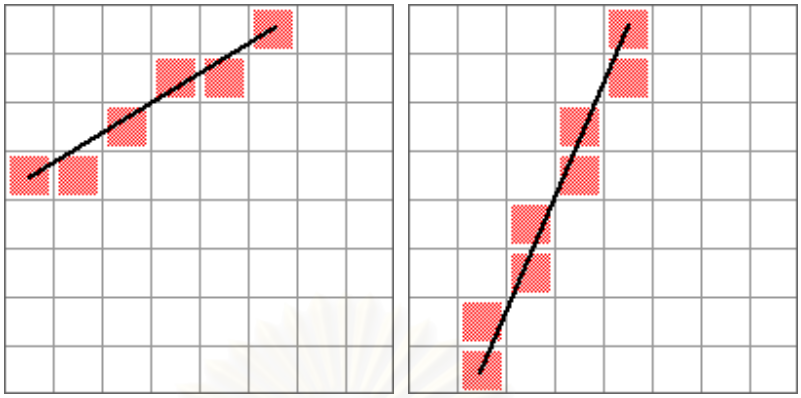
หลักการของการวาดเส้นตรงดีดีเอ คือ การหาค่าแตกต่างในแนวแกน X และในแนวแกน Y ของจุดปลายทั้งสองของเส้นตรง แนวแกนที่มีค่าแตกต่างมากกว่าจะเป็นแนวแกนที่ถูกเปลี่ยนแปลงค่าทีละ 1 ส่วนอีกแนวแกนหนึ่งจะถูกเปลี่ยนแปลงค่าตามแนวแกนแรก
ขั้นตอนวิธีการวาดเส้นตรงแบบดีดีเอมีดังนี้

1. ให้จุดปลายของเส้นตรงอยู่บนคู่ลำดับ (X1,Y1) และ (X2,Y2) ตามลำดับ
2. ให้ M คือความชันของเส้นตรงซึ่งหาได้จากสมการ

$$M = \frac{Y2-Y1}{X2-X1}$$

3. DX คือผลต่างของ X1 กับ X2
   DY คือผลต่างของ Y1 กับ Y2
   AX คือค่าที่เปลี่ยนแปลงในแนวแกน X
   AY คือค่าที่เปลี่ยนแปลงในแนวแกน Y
4. ถ้า DX>DY และ X1<X2 ให้ AX = 1 และ AY = M
   ถ้า DX>DY และ X1>X2 ให้ AX = -1 และ AY=-M
   ถ้า DX<DY และ X1<X2 ให้ AX=1/M และ AY=1
   ถ้า DX<DY และ X1>X2 ให้ AX=-1/M และ AY=-1
5. ให้ (X,Y) คือจุดภาพที่เลือกเพื่อวาดเส้นตรง มีค่าเริ่มต้นเป็น X = X1 และ Y = Y1
6. เลือกจุดภาพต่อไปโดยให้ X' = X + AX และ Y' = Y + AY
7. ทำข้อ 6 จนกระทั่งจุดภาพที่เลือกคือจุดภาพ (X2,Y2)

จากตัวอย่างรูปภาพ 2.10 จะเปลี่ยนแปลงค่าตามแนวแกน X ทีละ 1 (AX=1) และเปลี่ยนแปลงค่าตามแนวแกน Y ตามค่าความชัน (AY=M) ส่วนรูปที่ 2 เปลี่ยนค่าตามแนวแกน Y ทีละ 1 (AY=1) และเปลี่ยนค่าตามแนวแกน X ตามค่าผกผันของความชัน (AX=1/M)



รูปภาพ 2.10 แสดงการเลือกจุดภาพเพื่อวาดเส้นตรง ภาพซ้าย เปลี่ยนค่าตามแนวแกน X ภาพขวาเปลี่ยนค่าตามแกน Y

2.3.7 **การหาระยะห่างระหว่างจุดภาพ** (กิตติมา มะแก้ว, 2538: 5)

ให้จุดภาพ P Q และ Z อยู่บนคู่ลำดับ (Xp, Yp) (Xq, Yq) และ (Xz, Yz) ตามลำดับ โดย D เป็นระยะห่างระหว่างจุดภาพ ซึ่งมีคุณสมบัติดังนี้

1. $D(P,Q) \geq 0$ โดย $D(P,Q) = 0$ เมื่อ $P=Q$
2. $D(P,Q) = D(Q,P)$
3. $D(P,Z) \leq D(P,Q)+D(Q,Z)$
4. ระยะห่างระหว่างจุดภาพ P กับ จุดภาพ Q สามารถหาได้ดังนี้

$D(P,Q) = [(X_p-X_q)^2+(Y_p-Y_q)^2]^{1/2}$

2.2.8 **การหาเส้นขอบบนวัตถุ** (Edge Detection) (กิตติมา มะแก้ว, 2538: 6)

วิธีการหาเส้นขอบของวัตถุคือการแยกจุดภาพที่อยู่บนเส้นขอบของวัตถุออกจากจุดภาพที่เป็นส่วนพื้นและจุดภาพที่อยู่ภายในวัตถุ การตัดสินว่าจุดภาพใดเป็นจุดภาพที่อยู่บนเส้นขอบของวัตถุทำได้ดังนี้

1. กำหนดให้จุดภาพที่เป็นวัตถุมีค่าเป็น **1** และจุดภาพในส่วนพื้นมีค่าเป็น **0**

2. ให้ P1(X,Y) เป็นจุดภาพที่ต้องการทดสอบว่าเป็นจุดภาพที่อยู่บนเส้นขอบหรือไม่

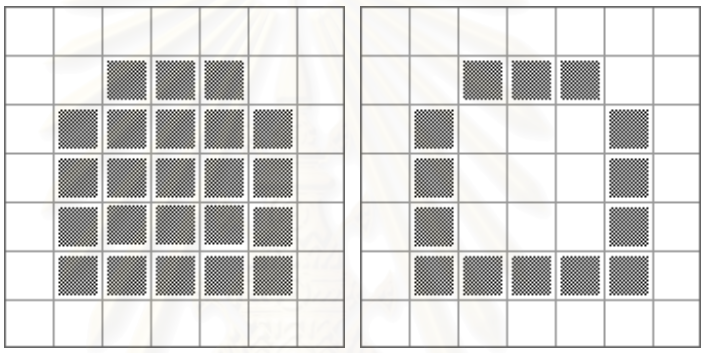
3. **P1** จะเป็นจุดภาพที่อยู่บนเส้นขอบของวัตถุ หากมีคุณสมบัติตรงตามเงื่อนไข **2** ข้อ ดังนี้

เงื่อนไขที่ 1 P1 มีค่าเป็น 1

เงื่อนไขที่ 2 จำนวนจุดภาพใกล้เคียง 4 จุดของจุดภาพ P1 ไม่มีค่าเป็น 1 ทุกจุด นั่นคือ $P2+P4+P6+P8 \neq 4$

รูปภาพ 2.11 แสดงการตัดสินจุดขอบ โดย P1 ไม่เป็นจุดภาพบนเส้นขอบ

จากภาพ 2.11 จุดภาพ P1 ไม่เป็นจุดภาพบนเส้นขอบของวัตถุเพราะจุดภาพใกล้เคียงของจุดภาพ P1 มีค่าเป็น 1 ทุกจุด

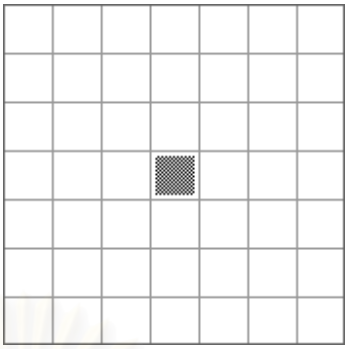


รูปภาพ 2.12 แสดงวัตถุและเส้นขอบวัตถุ ภาพซ้าย ก่อนหาเส้นขอบ ภาพขวา หลังหาเส้นขอบ

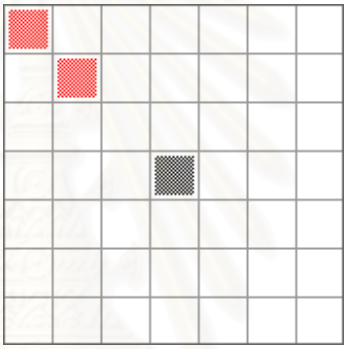
2.3.9 **การค้นหามุมของวัตถุในภาพ** (กิตติมา มะแก้ว, 2538: 13)

การค้นหามุมของวัตถุในภาพเป็นคุณลักษณะที่สำคัญในการวิเคราะห์รูปภาพ เช่น ใช้มุมวิเคราะห์ความแตกต่างกันของรูปภาพ ใช้มุมในการรับรู้ภาพวาดลายเส้น เป็นต้น ด้วยเหตุนี้วิธีการค้นหามุมได้ถูกพัฒนาขึ้นมาหลายวิธีเพื่อตอบสนองความต้องการที่แตกต่างกัน ตามวิธีดังต่อไปนี้

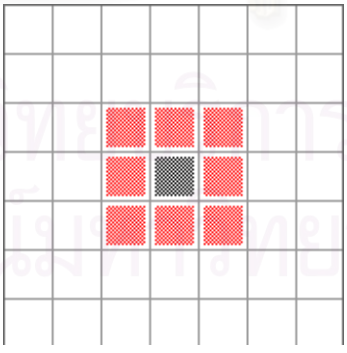
1. กำหนดหน้าต่างสี่เหลี่ยมจัตุรัสที่มีจุดศูนย์กลางอยู่ที่จุดภาพจุดหนึ่งที่ต้องการ
2. กำจัดจุดภาพในหน้าต่างที่ไม่ใช่มุมของวัตถุในภาพออกไป โดยทั่วไป จุดภาพที่ไม่ใช่มุมสามารถแบ่งออกได้เป็น 4 ลักษณะ คือ
    a. จุดภาพโดดๆ คือจุดภาพที่ไม่มีจุดภาพใดๆ ต่อเชื่อมด้วย ดังนั้นจึงไม่เกิดมุม แสดงได้ดังรูป



b. จุดภาพสิ้นสุด คือจุดภาพที่มีเส้นตรงเชื่อมต่อกับจุดภาพนี้เพียงเส้นเดียวเท่านั้น จะไม่ถือว่าเป็นมุมของวัตถุ แสดงได้ดังรูป

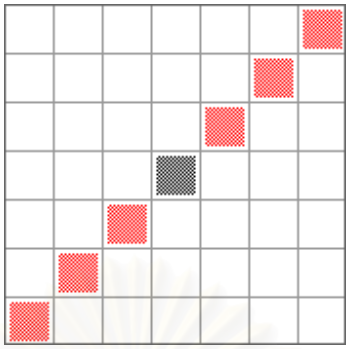


c. จุดภาพที่ไม่อยู่บนเส้นขอบของวัตถุ คือจุดภาพที่มีจุดภาพหลายจุดมาเชื่อมต่อกับจุดภาพนี้ จุดภาพนี้จะถือว่าเป็นส่วนภายในของวัตถุ ไม่ถือว่าเป็นมุมของวัตถุ ดังรูป



d. จุดภาพบนเส้นขอบของวัตถุที่ไม่ใช่มุมของวัตถุ คือจุดภาพที่มีเส้นตรง 2 เส้นเชื่อมต่อกับจุดภาพจุดนี้ โดยเส้นตรงทั้ง 2 เส้นทำมุมประมาณ 180 องศา จุดภาพนี้จะไม่ใช่มุมของวัตถุ ดังรูป

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

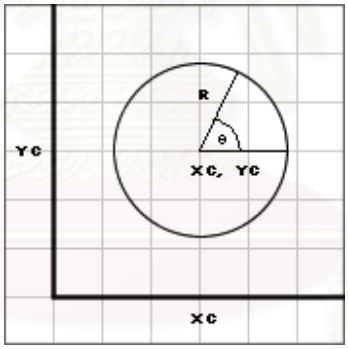


3. หาจุดภาพที่เป็นมุมจากจุดภาพในหน้าต่างที่ไม่ถูกจำกัดจากข้อ 2 โดยเลือกจุดภาพที่เส้นตรง 2 เส้น ที่เชื่อมต่อกับจุดภาพนี้ทำมุมขนาดที่กำหนดให้ยอมรับได้

2.3.10 การสร้างรูปวงกลม (ธรรมศักดิ์ โกศลกิจจา, 2532: 9)

การสร้างวงกลมจะกำหนดจุดศูนย์กลางของวงกลม (Center Coordinate) (xc, yc) และรัศมี (r)

$$(x - x_c)^2 + (y - y_c)^2 = r^2$$



กรณีที่จุดศูนย์กลางอยู่ที่จุด (0,0) สมการวงกลมจะอยู่ในลักษณะ

$$x^2 + y^2 = r^2$$

หรือ

$$y = yc \pm \sqrt{r^2 - (x - xc)^2}$$

สมการนี้ค่อนข้างยุ่งยากในการคำนวณหาค่า y ถ้าแปลงสมการนี้ให้อยู่ในระบบพิกัดขั้ว (Polar Coordinate) จะคำนวณค่า x และ y ได้ง่ายขึ้น

$x = xc + r\ \mathrm{Cos}\theta$
$y = yc + r\ \mathrm{Sin}\theta$
θ มีค่าตั้งแต่ 0 ถึง 2 π

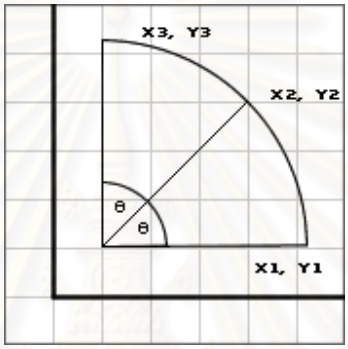
r คือรัศมี

ความยาวของส่วนโค้ง (arch length) คือส่วนโค้งที่เกิดจากการเปลี่ยนแปลงของมุมเท่ากับ θ

$$\text{Arch length} = r \times \theta$$

การสร้างวงกลมตามระบบพิกัดขั้ว จะพบปัญหาเกี่ยวกับการคำนวณค่า Cosθ และ Sinθ ทุกครั้งที่เปลี่ยนแปลงค่าของมุม θ เพื่อความสะดวกในการสร้างวงกลมสามารถแปลงค่ามุม θ ให้อยู่ในรูปของ dθ และทำการคำนวณค่าเพียงครั้งเดียวดังนี้

กำหนดจุด 2 จุดให้อยู่บนวงกลม (x1, y1), (x2,y2)



$$x1 = r\cos\theta$$
$$y1 = r\sin\theta$$
$$x2 = r\cos(\theta + d\theta)$$
$$y2 = r\sin(\theta + d\theta)$$

กำหนดให้ dθ เป็นค่ามุมที่มีการเปลี่ยนแปลงคงที่

หลักการทางตรีโกณมิติ

$$x2 = r\cos(\theta)\cos(d\theta) - r\sin(\theta)\sin(d\theta)$$
$$y2 = r\sin(\theta)\cos(d\theta) + r\cos(\theta)\sin(d\theta)$$

แทนค่า x1 และ y1 จากสูตร

$$x2 = x1\cos(d\theta) - y1\sin(d\theta)$$
$$y2 = y1\cos(d\theta) + x1\sin(d\theta)$$

ดังนั้นการสร้างภาพวงกลม เป็นการคำนวณหาตำแหน่งใหม่ตามเส้นรอบวงของวงกลมจากจุดเริ่มต้น คือ x1 เท่ากับ 0 และ y1 เท่ากับรัศมี และกำหนดค่ามุมที่เปลี่ยนแปลงคงที่ เท่ากับ dθ

ขั้นตอนวิธีสร้างวงกลมจะทำการคำนวณจุดของเส้นรอบวงของวงกลมล้อมรอบจุดศูนย์กลาง ณ จุดกำเนิด และเปลี่ยนตำแหน่งจุดศูนย์กลางจากจุดกำเนิดไปยังตำแหน่งใดๆ โดยบวกค่า xc เท่ากับ x และบวกค่า yc เท่ากับ y ดังนั้นจุดศูนย์กลางวงกลมจะเปลี่ยนไปยังตำแหน่ง (xc, yc)

2.3.11 ผลกระทบของการใช้คอมพิวเตอร์ในการออกแบบขั้นต้น

เนื่องจากในปัจจุบันมีการนำเอาคอมพิวเตอร์มาใช้ในการนำเสนองานทางสถาปัตยกรรมอย่างแพร่หลาย จึงเกิดคำถามระหว่างสถาปนิกที่ใช้เครื่องมือชนิดใหม่นี้กับสถาปนิกที่ใช้เครื่องมือแบบเดิม ว่าการใช้คอมพิวเตอร์จะมีผลกระทบกับการทำงานหรือไม่อย่างไร และจากการศึกษาของ (Gerard Cesar Gabriel, Mary Lou Maher, 1998) โดยใช้อาสาสมัครประมาณ 50 คนพบว่าการนำเสนองานด้วยคอมพิวเตอร์นั้นจะสามารถทำได้สะดวกรวดเร็วถ้านำเสนอด้วยรูปเรขาคณิตหรือถ้ามีการทำซ้ำการคัดลอกงาน และการใช้ปากกาดินสอแบบเก่าจะทำได้รวดเร็วถ้าใช้วาดรูปทรงอิสระซึ่งก็เป็นที่แน่นอนว่าการใช้คอมพิวเตอร์นั้นจะมีผลกระทบต่อการทำงานอย่างแน่นอน แต่หากเรานำเอาคอมพิวเตอร์มาใช้วาดรูปทรงเรขาคณิตแล้วก็จะสามารถทำได้สะดวกรวดเร็ว กล่าวคือการออกแบบขั้นต้นของงานสถาปัตยกรรมที่เน้นการใช้รูปทรงเรขาคณิต สามารถนำคอมพิวเตอร์มาช่วยเพื่อให้ทำงานได้สะดวกรวดเร็วมากยิ่งขึ้น

2.3.12 การเก็บข้อมูลแบบเว็กเตอร์และแรสเตอร์ (กิตติมา มะแก้ว, 2538: 1)

คอมพิวเตอร์กราฟิกมีวิธีในการแทนรูปภาพอยู่ 2 วิธีคือ แผนที่บิต (bitmap) หรือแรสเตอร์ (raster) และ เว็กเตอร์ (vector) การแทนรูปภาพทั้งสองวิธีนี้จะมีลักษณะและความเหมาะสมกับงานที่แตกต่างกันไปนั่นคือ การแทนรูปภาพด้วยวิธีแรสเตอร์จะแบ่งรูปภาพออกเป็นชิ้นส่วนเล็กๆ ที่เรียกว่า จุดภาพ (pixel) แล้วจัดเก็บค่าสีของแต่ละจุดภาพเอาไว้ วิธีแรสเตอร์นี้จึงเหมาะกับงานที่ใช้รูปภาพที่ซับซ้อนทั้งในด้านสี แสง เงา และองค์ประกอบของรูปภาพ ส่วนการแทนรูปภาพด้วยวิธีเว็กเตอร์นั้น จะเก็บชุดอักขระที่ใช้อธิบายรูปภาพนั้นไว้ วิธีเว็กเตอร์นั้นจึงเหมาะกับงานรูปภาพที่มีรูปทรงเรขาคณิต เช่นแผนที่ แผนผัง กราฟ ชาร์ต และงานด้านการใช้คอมพิวเตอร์ช่วยออกแบบ (CAD) เป็นต้น

2.4 โปรแกรมที่เกี่ยวข้องกับการวาดภาพลายเส้น

จากการศึกษางานวิจัยที่เน้นการวาดภาพลายเส้นเป็นหลัก พบว่านอกจากจะเป็นการพัฒนาโปรแกรมที่เน้นภาพวาดลายเส้นทั้งในแบบเรขาคณิตและแบบเส้นอิสระแล้ว ยังมีการพัฒนาโปรแกรมที่เกี่ยวข้องกับสร้างหุ่นจำลองอีกด้วย โดยสามารถรวบรวมโปรแกรมที่เกี่ยวข้องทั้งหมดได้ดังนี้

2.4.1 โปรแกรม Smooth Teddy

Website: http://www-ui.is.s.u-tokyo.ac.jp/~takeo/java/smoothteddy/index.html

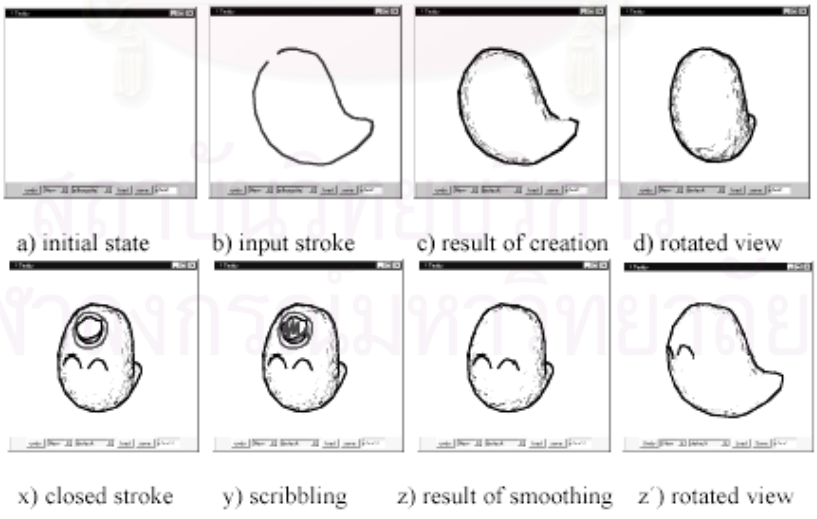
โดย Takeo Igarashi ประเทศญี่ปุ่น

วัตถุประสงค์

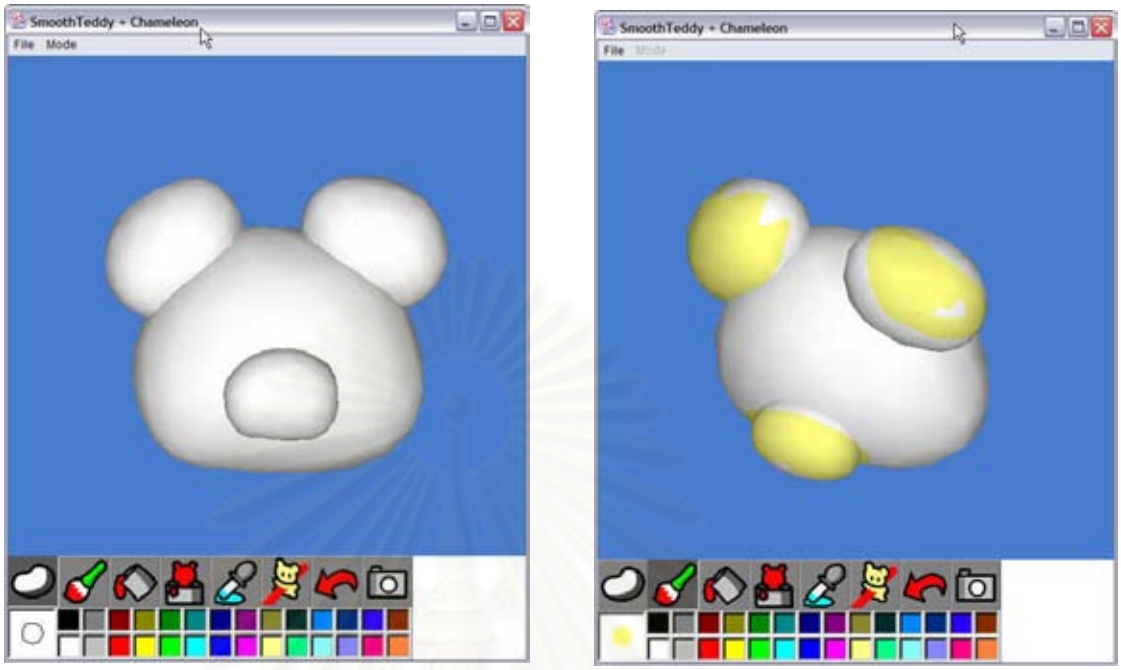
เป็นงานวิจัยทางด้าน user interface ที่เน้นการวาดภาพลายเส้นเพื่อให้สามารถออกแบบหุ่นจำลองรูปทรงอิสระได้อย่างรวดเร็ว โดยพัฒนาเป็นโปรแกรมสร้างหุ่นจำลอง 3 มิติจากภาพวาดลายเส้นอย่างง่าย ให้สามารถสร้างหุ่นจำลองทรงกลมเช่น ตุ๊กตาหมี ได้และใช้ลักษณะการวาดภาพลายเส้นอย่างอิสระเป็นเครื่องมือในการออกแบบ เมื่อผู้ใช้งานมีการลากเส้นแบบอิสระให้เกิดรูปร่างที่เป็นขอบของวัตถุ จากนั้นโปรแกรมจะทำการสร้างหุ่นจำลอง 3 มิติขึ้นมาตามรูปร่างนั้น โดยที่ผู้ใช้งานไม่จำเป็นต้องแก้ไขจุดควบคุมของหุ่นจำลอง (Control Point) เหมือนอย่างโปรแกรมสร้างหุ่นจำลอง 3 มิติอื่นๆ ซึ่งในการใช้งานครั้งแรกผู้ใช้จะสามารถสร้างหุ่นจำลองได้ทันทีจากการวาดภาพลายเส้นภายในเวลาอันรวดเร็ว และให้ความรู้สึกเหมือนการแกะสลักหุ่นจำลองด้วยมือ

จุดเด่น - โปรแกรมสามารถสร้างหุ่นจำลอง 3 มิติจากการวาดภาพลายเส้นได้อย่างรวดเร็ว
- สามารถลบส่วนที่ไม่ต้องการได้อย่างรวดเร็ว
- โปรแกรมนี้สามารถปรับเปลี่ยนสีของหุ่นจำลองได้อย่างอิสระ

จุดด้อย - โปรแกรมไม่สามารถสร้างรูปร่างเรขาคณิตได้ เหมาะกับการสร้างรูปทรงอิสระเท่านั้น



รูปภาพ 2.13 ตัวอย่างลำดับขั้นตอนการทำงานของโปรแกรมในการสร้างหุ่นจำลอง 3 มิติจากการวาดภาพลายเส้นอย่างง่าย



รูปภาพ 2.13 ภาพจากหน้าจอของโปรแกรม Smooth Teddy

2.4.2 โปรแกรม Chateau

Website: http://www-ui.is.s.u-tokyo.ac.jp/~takeo/chateau/chateau.html

โดย Takeo Igarashi ประเทศญี่ปุ่น

วัตถุประสงค์

เป็นการพัฒนาโปรแกรมเพื่อนำเสนอระบบช่วยเหลือผู้ใช้งานรูปแบบใหม่ ซึ่งช่วยพัฒนาการใช้งานโปรแกรมสร้างหุ่นจำลองให้เหมาะสมมากขึ้น จากระบบช่วยเหลือผู้ใช้ของโปรแกรมสร้างหุ่นจำลองแบบเดิม โดยที่ผู้ใช้งานจะได้รับข้อมูลช่วยเหลือในการทำงานกับโปรแกรมหลังจากที่เลือกหุ่นจำลองที่ต้องการในพื้นที่ทำงาน จากนั้นโปรแกรมจะทำการสร้างระบบช่วยเหลือที่สัมพันธ์กับหุ่นจำลองที่เลือก แล้วแสดงผลออกมาเป็นทางเลือกด้วยภาพตัวอย่างขนาดเล็กทางหน้าจอ ระบบนี้เป็นการตีความการทำงานของผู้ใช้งานโดยทำงานคู่ขนานไปพร้อมๆ กัน

เป้าหมายในการพัฒนาโปรแกรมนี้ก็เพื่อปรับปรุงระบบตอบโต้กับผู้ใช้ที่สัมพันธ์กับพฤติกรรมการใช้งาน โดยคาดหวังว่าโปรแกรมในลักษณะนี้จะสามารถช่วยเหลือในการสร้างหุ่นจำลองได้เทียบเท่ากับการใช้โปรแกรมที่เน้นการใช้คำสั่งแบบเดิม โดยโปรแกรมที่มีระบบ

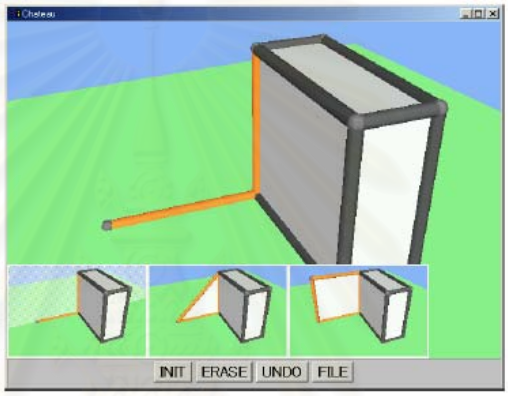
สร้างทางเลือกนี้จะช่วยให้ผู้ใช้งานมือใหม่สามารถสร้างหุ่นจำลองได้โดยง่ายและสามารถใช้งานกับคำสั่งที่ซับซ้อนได้เพียงเลือกทางเลือกที่ระบบสร้างขึ้นมาให้เท่านั้น

จุดเด่น - โปรแกรมสามารถสร้างรูปทรงเรขาคณิตได้อย่างรวดเร็ว

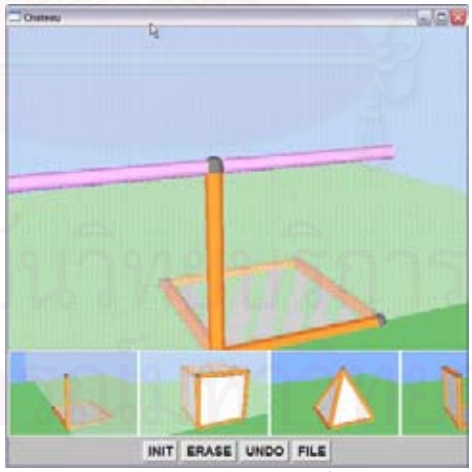
- โปรแกรมไม่มีปุ่มคำสั่งให้เลือก แต่สามารถสร้างหุ่นจำลองด้วยการลากเส้นได้ทันที
- โปรแกรมจะสร้างทางเลือกในการทำงานช่วยให้สร้างหุ่นจำลองได้สะดวกขึ้น

จุดด้อย - ไม่สามารถเคลื่อนย้ายเส้นที่สร้างไว้แล้วได้ ลบเส้นได้เพียงอย่างเดียว

- เนื่องจากไม่มีคำสั่งช่วยในการวาดเส้นทำให้สร้างหุ่นจำลองได้ช้า



รูปภาพ 2.14 ภาพตัวอย่างของโปรแกรม Chateau



รูปภาพ 2.14 โปรแกรมจะสร้างทางเลือกในการทำงานที่สัมพันธ์กับการวาดภาพของผู้ใช้งานในขณะนั้น

2.4.3 โปรแกรม Quick Sketch

Website: http://rabbit.prakinf.tu-ilmenau.de/qsketch.html

โดย Dae-Hyun Kim, Paul Michalik, Stephan Hofer ประเทศเยอรมันนี

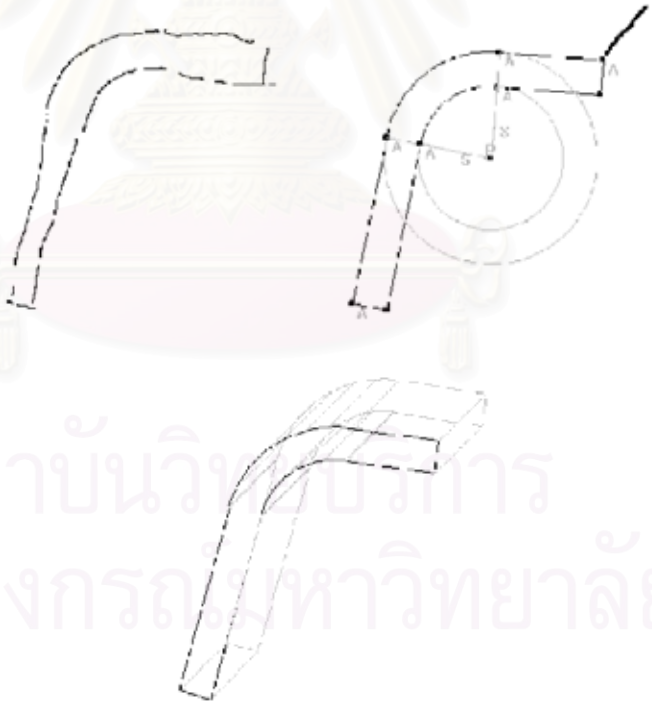
วัตถุประสงค์

โปรแกรม Quick Sketch เป็นโปรแกรมสร้างหุ่นจำลอง 2 มิติและ 3 มิติสำหรับคอมพิวเตอร์ที่ใช้ปากกาอิเล็คทรอนิคส์ โดยผู้ใช้สามารถสร้างหุ่นจำลองด้วยการวาดภาพลายเส้นลงไปบนจอภาพของคอมพิวเตอร์ โดยโปรแกรมจะแปลงรูปร่างและความสัมพันธ์ทางเรขาคณิตลงไปในจอภาพจากการลากปากกาอิเล็คทรอนิคส์ในเวลาเดียวกัน (real time)

จุดเด่น - โปรแกรมสามารถสร้างหุ่นจำลอง 3 มิติจากภาพวาดลายเส้นได้อย่างรวดเร็ว

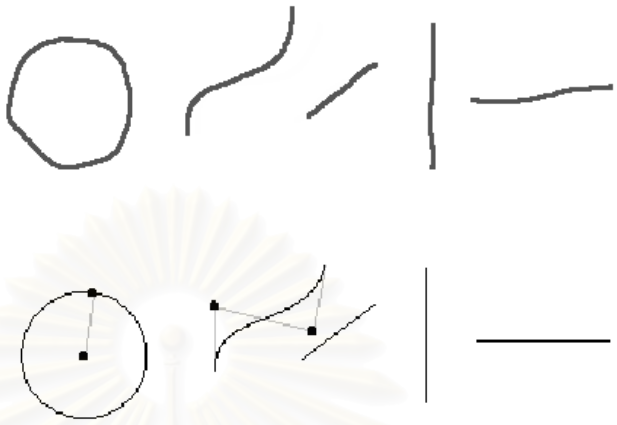
- สามารถสร้างเส้นโค้งและเจาะช่องเปิดจากภาพวาดลายเส้นได้
- โปรแกรมสามารถรับรู้ภาพวาดลายเส้นด้วยวิธีทางเรขาคณิต

จุดด้อย - โปรแกรมไม่สามารถปรับขนาดหุ่นจำลองที่สร้างเสร็จแล้ว ต้องกำหนดขนาดและรูปร่างจากภาพวาดลายเส้นเท่านั้น



รูปภาพ 2.15 ภาพตัวอย่างขั้นตอนการทำงานของโปรแกรม เริ่มจากการวาดภาพลายเส้นจากนั้นโปรแกรมจะทำการแปลงภาพลายเส้นให้เป็นเส้นเรขาคณิตในคอมพิวเตอร์เพื่อสร้างหุ่นจำลอง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปภาพ 2.15 ภาพตัวอย่างขั้นตอนการแปลงภาพวาดลายเส้นให้เป็นเส้นเรขาคณิต ในคอมพิวเตอร์ของโปรแกรม Quick Sketch

2.4.4 โปรแกรม Sketching Interfaces Like Krazy (SILK)

Website: http://www.acm.org/sigchi/chi96/proceedings/videos/Landay/jal1txt.htm

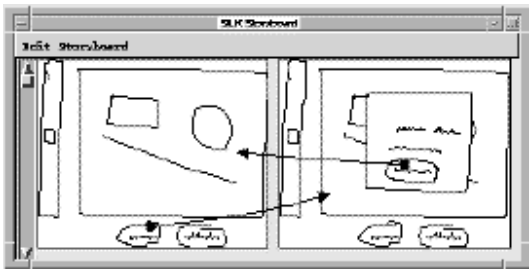
โดย James A. Landay ประเทศสหรัฐอเมริกา

วัตถุประสงค์

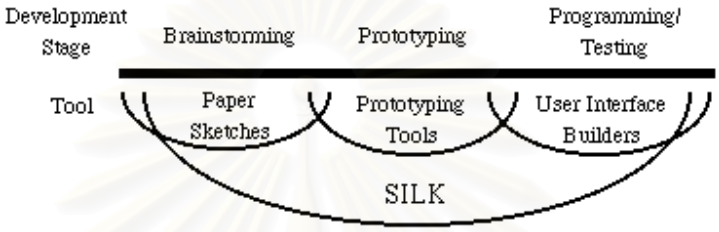
เป็นโปรแกรมที่ช่วยออกแบบส่วนโต้ตอบกับผู้ใช้งาน ของโปรแกรมคอมพิวเตอร์โดยใช้การวาดภาพลายเส้นเป็นหลัก เพื่อให้เหมาะสมกับการใช้งานของนักออกแบบเพราะในการออกแบบขั้นต้นนักออกแบบมักจะวาดภาพคร่าวของงานออกแบบเอาไว้มากกว่าที่จะเน้นที่รายละเอียด ดังนั้นจึงพัฒนาโปรแกรมเพื่อให้นักออกแบบสามารถวาดภาพได้อย่างรวดเร็วด้วยปากกาอิเลคทรอนิกส์ และพยายามเลียนแบบคุณสมบัติของกระดาษให้มากที่สุดเช่น ความรวดเร็วในการวาด และพื้นที่ในการวาดที่ยืดหยุ่น เป็นต้น และยังเสริมคุณสมบัติที่สามารถวาดภาพต่อเนื่องแบบสตอรี่บอร์ด เพื่อแสดงการเปลี่ยนแปลงของส่วนโต้ตอบกับผู้ใช้งานได้อีกด้วย

จุดเด่น - โปรแกรมสามารถสร้างระบบโต้ตอบกับผู้ใช้งานเพื่อแสดงแนวคิดในการออกแบบได้

จุดด้อย - โปรแกรมไม่สามารถรับรู้ภาพวาดลายเส้นและแปลงให้เป็นภาพเรขาคณิตได้ เหมาะกับการแสดงแนวความคิดในการออกแบบเท่านั้น



ภาพตัวอย่างการใช้งานโปรแกรม SILK



รูปภาพ 2.16 ภาพลักษณะการใช้งานโปรแกรม SILK ในขั้นต่างๆ ของการพัฒนาโปรแกรมคอมพิวเตอร์

2.4.5 โปรแกรม PerSketch (A Perceptually-Supported Sketch Editor)

Website: http://portal.acm.org/citation.cfm?id=192494&dl=ACM&coll=portal

โดย Thomas Morand และ Eric Saund ประเทศสหรัฐอเมริกา

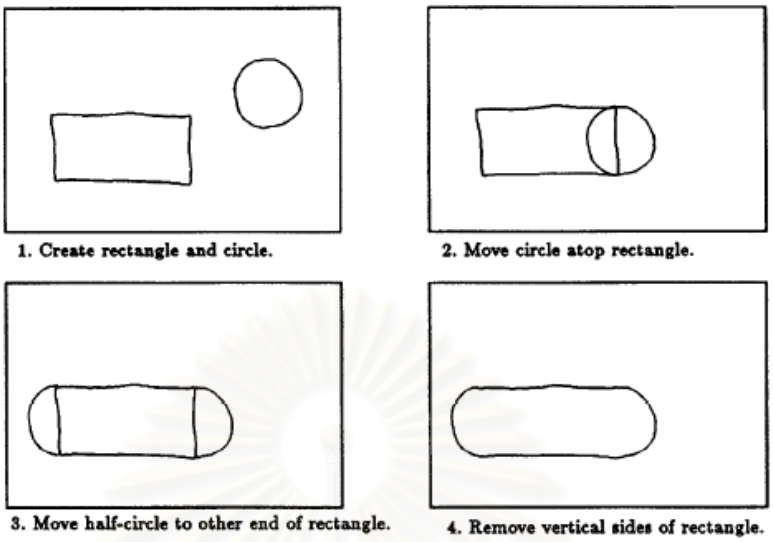
วัตถุประสงค์

เป็นโปรแกรมที่มีวัตถุประสงค์เพื่อพัฒนาขั้นตอนวิธีที่สามารถเข้าใจความหมายของสิ่งที่ผู้ใช้งานวาดภาพลงไป โดยจะเน้นที่การรับรู้ของผู้ใช้งานเป็นหลัก ซึ่งจะทำได้โดยสร้างวิธีการทางคอมพิวเตอร์ให้รวบรวมเส้นที่วาดไว้อย่างกระจัดกระจาย ให้กลายเป็นหน่วยเดียวกัน แล้วจัดใหม่ให้แต่ละหน่วยสร้างความสัมพันธ์ระหว่างกันหรือที่เรียกว่า WYPZWYG (What You Perceive is what you get)

โปรแกรมนี้จะใช้เครื่องหมายเป็นตัวแทนของโครงสร้างของภาพทั้งหมด โดยจะมีคุณลักษณะแตกต่างกันไปตามตำแหน่งที่ตั้ง สัดส่วน และทิศทางเป็นต้น

จุดเด่น - โปรแกรมสามารถพัฒนาระบบรับรู้ภาพวาดลายเส้นให้วิเคราะห์ความสัมพันธ์ทางเรขาคณิตได้

จุดด้อย - ผู้ใช้ไม่มีทางเลือกในการเลือกการรับรู้ภาพวาดลายเส้น โปรแกรมจะตีความหมายของภาพด้วยขั้นตอนวิธีที่กำหนดไว้เท่านั้น



รูปภาพ 2.17 ตัวอย่างการใช้งานโปรแกรม PerSketch

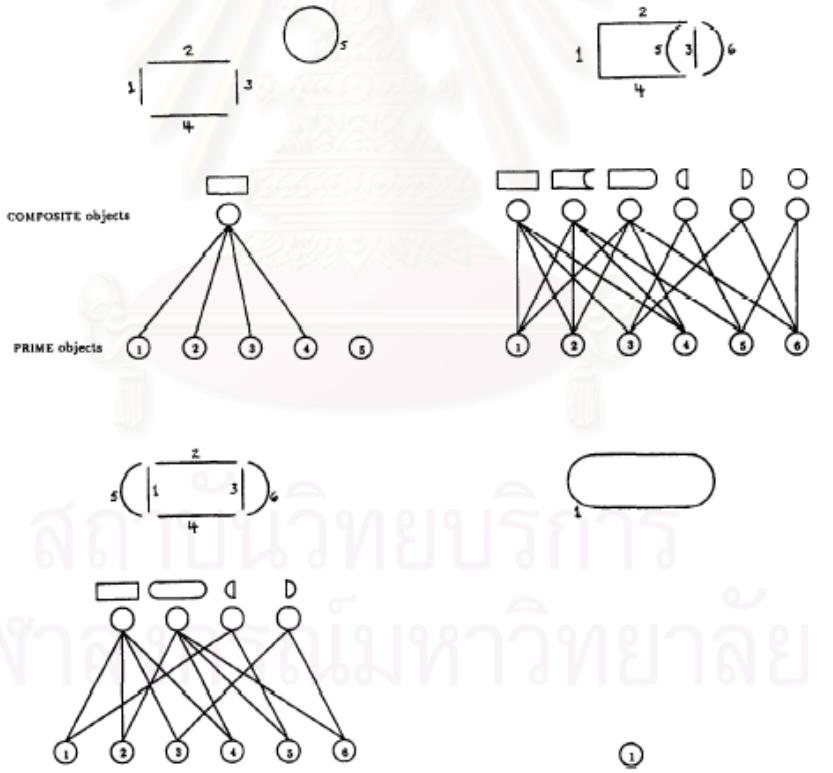


Figure 3: The Object Lattice of PRIME objects (elemental curve fragments) and COMPOSITE objects (emergent figures) underlying the sketch creation and editing steps of figure 1.

รูปภาพ 2.18 ตัวอย่างวิธีการทางคอมพิวเตอร์ที่ใช้ในการแปลงภาพวาดลายเส้น

2.4.6 โปรแกรม AutoDesk Architectural Studio

Website: http://usa.autodesk.com/adsk/section/0,,585585-123112,00.html

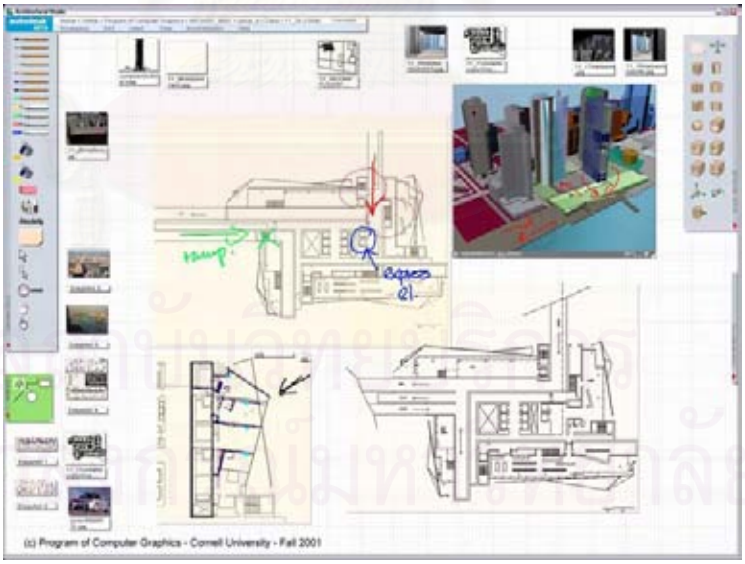
โดยบริษัท AutoDesk Inc. ประเทศสหรัฐอเมริกา

วัตถุประสงค์

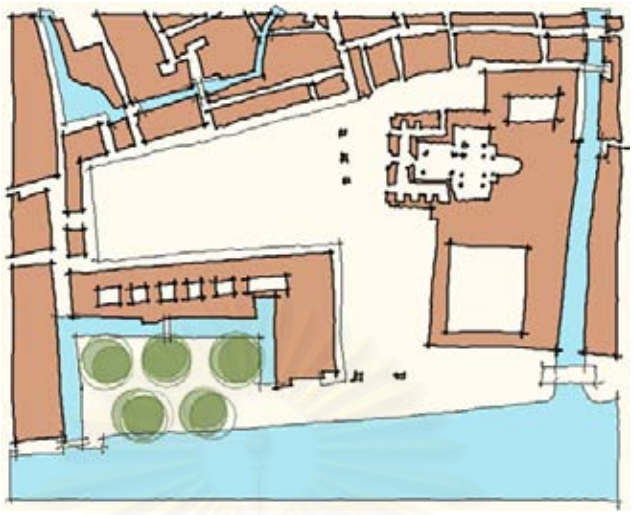
เป็นโปรแกรมสำหรับช่วยสถาปนิกในการออกแบบร่างและสร้างหุ่นจำลอง โดยมีการออกแบบโปรแกรมให้มีสภาพแวดล้อมและเครื่องมือในการทำงานคล้ายกับการทำงานบนโต๊ะเขียนแบบ ในการใช้งานผู้ใช้จะสามารถวาดภาพลายเส้นลงไปบนรูปถ่ายหรือผังเมืองได้ โดยใช้เทคนิคของระบบ Layer ที่มีลักษณะโปร่งใสคล้ายกับกระดาษร่าง และเน้นที่ความเรียบง่ายของส่วนติดต่อกับผู้ใช้ User Interface ที่เข้าใจได้ง่าย

จุดเด่น - โปรแกรมสามารถใช้ภาพวาดลายเส้นช่วยในการออกแบบสถาปัตยกรรมขั้นต้นได้
- สามารถใช้ภาพถ่ายหรือสร้างหุ่นจำลอง 3 มิติมาประกอบกับภาพวาดลายเส้นได้
- มีเครื่องมือวัดระยะและเครื่องมือวาดภาพหลายรูปแบบให้เลือกใช้

จุดด้อย - โปรแกรมไม่สามารถรับรู้ภาพวาดลายเส้นได้ โดยจะแสดงภาพวาดลายเส้นตามที่ผู้ใช้วาดทุกประการ



รูปภาพ 2.19 ตัวอย่าง User Interface ของโปรแกรม Architectural Studio



รูปภาพ 2.20 ตัวอย่างการใช้โปรแกรม Architectural Studio

2.4.7 **งานวิจัย** The Computer's Role in Sketch Design: A Transparent Sketching Medium

Website: http://www.arct.cam.ac.uk/research/pubs/pdfs/trinder99a.pdf

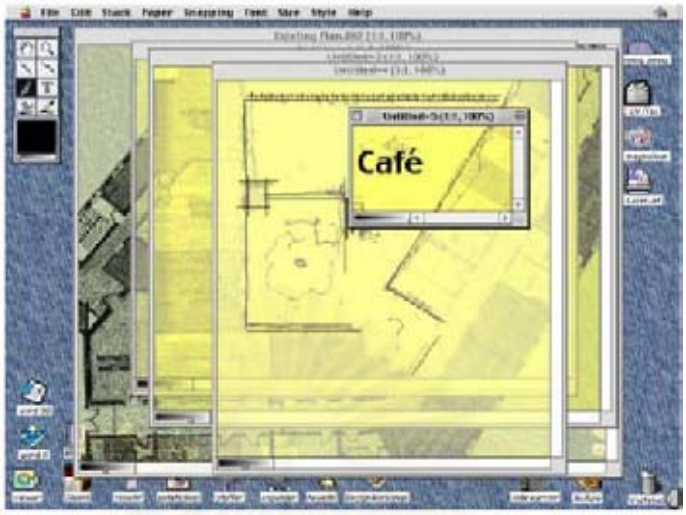
วัตถุประสงค์

เป็นงานวิจัยที่วิเคราะห์ข้อดีข้อเสียของโปรแกรมช่วยออกแบบที่เกี่ยวข้องกับการวาดภาพลายเส้น และเน้นไปที่การศึกษาความโปร่งใสของสภาพแวดล้อมในการทำงานบนคอมพิวเตอร์ ที่มีลักษณะเหมือนกับการใช้กระดาษร่างซึ่งผู้ออกแบบจะมองเห็นแผ่นงานที่ซ้อนกันได้ และในงานวิจัยนี้ยังได้ทดลองสร้างโปรแกรมเพื่อช่วยสนับสนุนแนวคิดดังกล่าวอีกด้วย

จุดเด่น - โปรแกรมสามารถแสดงลำดับชั้นของแผ่นงานได้พร้อมๆ กัน ด้วยคุณสมบัติความโปร่งใส

- สามารถวาดภาพลายเส้นหรือใส่ตัวอักษรในแผ่นงานแต่ละชั้นได้
- มีการประยุกต์คุณลักษณะของกระดาษร่างเช่นสี และความโปร่งใสมาพัฒนาเป็นแผ่นงานในโปรแกรม

จุดด้อย - โปรแกรมไม่มีความสามารถรับรู้ภาพวาดลายเส้น



รูปภาพ 2.21 ตัวอย่างโปรแกรมทดลองประกอบงานวิจัยจะเห็นว่า
มีการสร้าง Layer ของแผ่นงานให้โปร่งใสเลียนแบบกระดาษร่าง

2.4.8 **โปรแกรม** Electronic Cocktail Napkin

Website: http://depts.washington.edu/napkin

พัฒนาโดย Mark D. Gross และ Ellen Yi-Luen Do ประเทศสหรัฐอเมริกา

วัตถุประสงค์

เป็นโปรแกรมที่สร้างขึ้นโดยสถาปนิกเพื่อพัฒนาระบบรับรู้ภาพวาดลายเส้น ให้สามารถนำไปใช้กับระบบช่วยจดจำข้อมูลของสถาปนิกหรือเพื่อนำไปใช้เรียกฐานข้อมูลทางสถาปัตยกรรมขึ้นมา โดยพัฒนามาจากพฤติกรรมการทำงานของสถาปนิกที่มักจะจดบันทึกความคิดต่างๆ เกี่ยวกับงานออกแบบ ลงบนเศษกระดาษ เช่น กระดาษเช็ดปาก (Napkin)

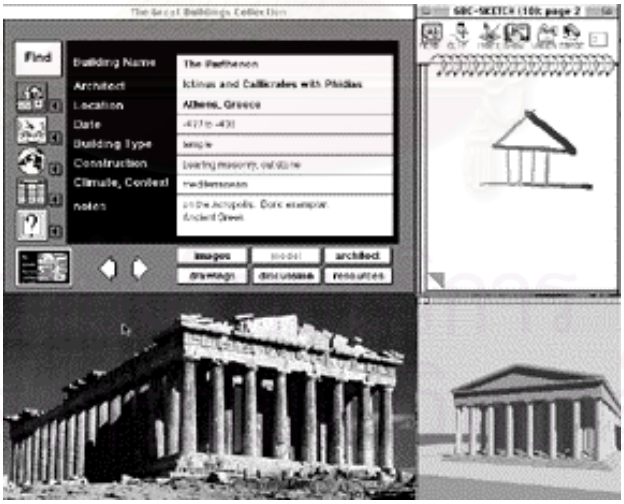
การพัฒนาโปรแกรมนี้เป็นการพัฒนาแนวทางใหม่ของโปรแกรมช่วยออกแบบทางสถาปัตยกรรม (CAD) ซึ่งเน้นไปที่การใช้คำสั่งในการสร้างเส้นเรขาคณิตเพื่อช่วยเขียนแบบ ในขณะที่สถาปนิกมักจะคิดงานออกแบบลงบนกระดาษ ทำให้โปรแกรม CAD แบบเดิมไม่สามารถตอบสนองการใช้งานในจุดนี้ได้แล้ว ฉะนั้นโปรแกรมนี้จึงสามารถตอบสนองต่อการทำงานของสถาปนิกมากขึ้นโดยใช้วาดภาพลายเส้นมาช่วยจดจำสัญลักษณ์ต่างๆ ที่วาดลงไป เช่น หากเราวาดภาพสี่เหลี่ยมจัตุรัสและวาดรูปสามเหลี่ยมอยู่ด้านบน โปรแกรมจะแปลงข้อมูลให้เป็นความหมายของ “บ้าน” เป็นต้น

จุดเด่น - โปรแกรมสามารถรับรู้ภาพวาดลายเส้นเพื่อสื่อความหมายทางสถาปัตยกรรมได้

- โปรแกรมสามารถนำภาพวาดลายเส้นมาประยุกต์ใช้กับการค้นหาฐานข้อมูลทางสถาปัตยกรรมที่บันทึกเอาไว้แล้วได้อย่างน่าสนใจ
- โปรแกรมมีระบบวาดภาพลายเส้นได้อย่างอิสระ

จุดด้อย - โปรแกรมไม่สามารถแปลงภาพวาดลายเส้นให้เป็นเส้นเรขาคณิตทางสถาปัตยกรรม
- โปรแกรมไม่มีเครื่องมือช่วยในการวาดเส้นเรขาคณิต

รูปภาพ 2.22 ภาพตัวอย่างการใช้งาน Electronic Cocktail Napkin



รูปภาพ 2.23 ภาพแสดงตัวอย่างการนำเอาโปรแกรมมาประยุกต์ใช้งานเพื่อเป็นระบบค้นหาข้อมูลจากภาพวาดลายเส้น

2.5 สรุปลักษณะสำคัญร่วมกันของโปรแกรมที่เกี่ยวข้องกับการวาดภาพลายเส้น

ชื่อโปรแกรม	ทัศนียภาพ **	เลเยอร์	ปากกาอิเล็กทรอนิกส์	ระบบรับรู้วัตถุ***	วาดเส้นเรขาคณิต	วาดเส้นอิสระ	สร้างหุ่นจำลอง****
Smooth Teddy	●	-	-	-	-	●	●
Chateau	●	-	-	-	●	-	●
Quicksketch	●	-	●	-	●	●	●
SILK	●	-	●	●	-	●	-
Persketch	-	-	-	●	●	●	-
Arch. Studio	●	●	●	-	●	●	●
Transparency	-	●	-	-	●	●	-
Cocktail Napkin	●	-	●	●	●	●	-
Vector Raster*	-	-	-	●	●	-	-

● **มีคุณสมบัติ** - **ไม่มีคุณสมบัติ**

* Vector Raster **คือ การเก็บข้อมูลแบบราสเตอร์แล้วแปลงให้เป็นแบบเว็กเตอร์** (กิตติมา มะแก้ว, 2538)

** **ทัศนียภาพ คือ การแสดงทัศนียภาพ 3 มิติหรือการแสดงภาพเคลื่อนไหวเพื่อช่วยเหลือในการทำงาน**

*** **ระบบรับรู้วัตถุ** (Recognition) **คือ ระบบที่ช่วยวิเคราะห์วัตถุจากข้อมูลที่ผู้ใช้งานป้อนให้**

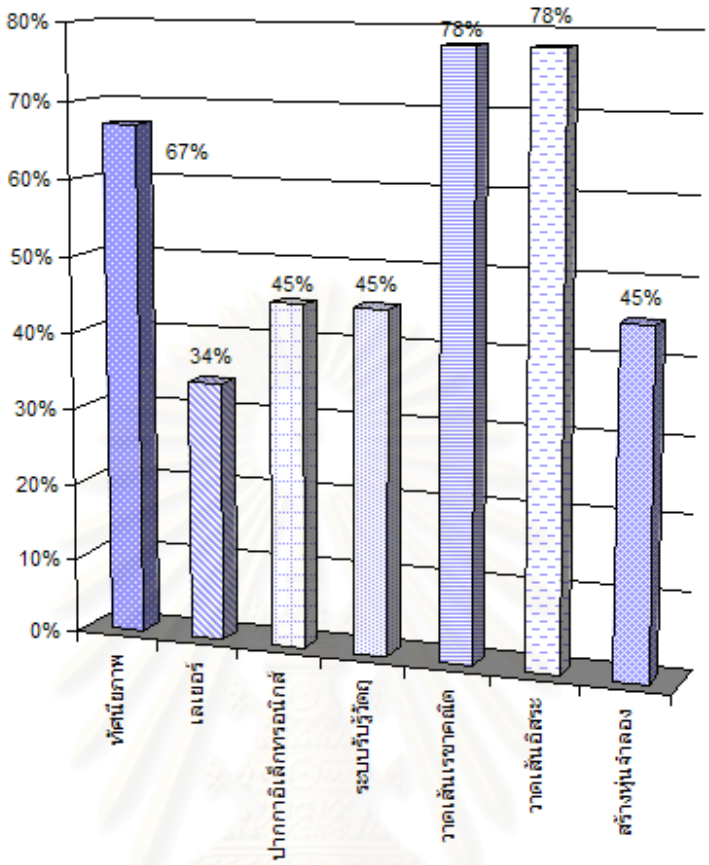
**** **สร้างหุ่นจำลอง คือ ระบบสร้างหุ่นจำลอง 3 มิติ**

ตาราง 2.1 สรุปลักษณะร่วมกันของโปรแกรมที่เกี่ยวข้อง

2.5.1 **เปรียบเทียบความถี่ของคุณลักษณะที่สำคัญของโปรแกรม** จากตาราง 2.1 สามารถนำความถี่มาเปรียบเทียบเพื่อหาคุณลักษณะที่สำคัญในการพัฒนาโปรแกรมได้ดังนี้

คุณลักษณะ	ความถี่ที่พบ(เปอร์เซ็นต์)
ทัศนียภาพ	67%
เลเยอร์	34%
ปากกาอิเล็กทรอนิกส์	45%
ระบบรับรู้วัตถุ	45%
วาดเส้นเรขาคณิต	78%
วาดเส้นอิสระ	78%
สร้างหุ่นจำลอง	45%

ตาราง 2.2 ตารางเปรียบเทียบความถี่ของคุณลักษณะที่สำคัญของโปรแกรมที่เกี่ยวข้องกับการวาดภาพลายเส้น



รูปภาพ 2.24 คุณลักษณะที่พบมากที่สุดคือ การวาดเส้นเรขาคณิต กับการวาดเส้นแบบอิสระ 78% เท่ากัน และมีระบบแสดงทัศนียภาพ 67% เป็นอันดับรองลงมา

จากรูปภาพ 2.28 สามารถสรุปลักษณะร่วมกันของโปรแกรมที่เกี่ยวข้องได้ว่า มีความนิยมในการพัฒนาระบบวาดเส้นมากที่สุดเพื่อรองรับการรูปแบบการวาดภาพลายเส้นที่มีหลากหลาย เช่นการวาดภาพลายเส้นทางสถาปัตยกรรม และการวาดภาพลายเส้นแบบอิสระ เป็นต้น ทั้งนี้เมื่อพบว่าความนิยมในการพัฒนาระบบวาดเส้นทั้งสองลักษณะมีความถี่เท่าๆกัน จึงน่าจะนำไปสู่การพัฒนาโปรแกรมสำหรับรับรู้ภาพวาดลายเส้นที่สามารถวาดเส้นได้ทั้งสองลักษณะ และสามารถสรุปคุณลักษณะอื่นๆ ที่น่าจะนำไปพัฒนาเป็นโปรแกรมรับรู้ภาพวาดลายเส้นได้ดังนี้

- ระบบทัศนียภาพ (3 มิติ)
- ระบบป้อนข้อมูลด้วยปากกาอิเล็กทรอนิกส์
- ระบบรับรู้วัตถุ
- ระบบวาดเส้นเรขาคณิตและวาดเส้นอิสระ
- ระบบสร้างหุ่นจำลอง

บทที่ 3

แนวทางการออกแบบโปรแกรม

หลังจากที่ได้ศึกษาเอกสารและงานวิจัยที่เกี่ยวข้องแล้ว ทำให้มองเห็นภาพรวมของการวิจัยขั้นตอนวิธีรับรู้วัตถุว่ามีความเกี่ยวข้องกับพฤติกรรมการวาดภาพของสถาปนิก ทฤษฎีทางคณิตศาสตร์เกี่ยวกับเรขาคณิต และเทคนิคการวาดเส้นแบบอิสระหรือแบบเรขาคณิต เป็นต้น จากนั้นจึงนำเอาข้อมูลทั้งหมดมาสรุปเป็นพื้นฐานในการออกแบบโปรแกรมต่อไป โดยสามารถลำดับขั้นตอนการออกแบบโปรแกรมได้เป็น 4 ขั้นตอนคือ

3.1 การเลือกเครื่องมือเพื่อใช้ในการวิจัย

3.2 การวิเคราะห์ข้อมูลจากเอกสารและงานวิจัยที่เกี่ยวข้อง

3.3 การกำหนดตัวแปรเพื่อนำมาใช้ในการพัฒนาโปรแกรม

3.4 การออกแบบและพัฒนาโปรแกรม

3.1 การเลือกเครื่องมือเพื่อใช้ในการวิจัย

ในการวิจัยและพัฒนาโปรแกรมจะต้องเลือกใช้เครื่องมือที่เหมาะสม โดยเครื่องมือสำหรับการพัฒนาโปรแกรมคอมพิวเตอร์สามารถแบ่งออกได้ 2 ประเภท คือ

3.1.1 Hardware

3.1.2 Software

3.1.1 Hardware

โดยอุปกรณ์ Hardware ที่เลือกได้แก่

- *คอมพิวเตอร์ที่ใช้ระบบปฏิบัติการ Microsoft Windows*[6] ซึ่งเป็นเครื่องมือที่ใช้กันอย่างแพร่หลาย เมื่อวิจัยและพัฒนาโปรแกรมเสร็จแล้วสามารถเผยแพร่ให้ผู้ใช้คอมพิวเตอร์ที่มีระบบปฏิบัติการ Microsoft Windows สามารถนำไปทดลองใช้ให้เกิดประโยชน์ได้ นอกจากนี้จะต้องมีอุปกรณ์แสดงผลกราฟิก 3 มิติที่รองรับคลังกราฟิก OpenGL version 1.2[7] ขึ้นไป

---

[6] Microsoft Corporation

[7] Silicon Graphic Inc. http://www.opengl.org

- *ปากกาอิเล็กทรอนิกส์* โดยอ้างอิงจากตาราง 2.2 เปรียบเทียบความถี่งานวิจัยที่เกี่ยวข้อง ทำให้เลือกใช้ปากกาอิเล็กทรอนิกส์ในงานวิจัยนี้ โดยปากกาอิเล็กทรอนิกส์จะทำหน้าที่ส่งข้อมูล input เพื่อสร้างจุดลงในจอภาพ ความถี่ของจุดจะขึ้นอยู่กับความเร็วในการสร้างกรอบภาพของคอมพิวเตอร์ต่อ 1/1000 วินาที (frame rates)

3.1.2 Software ได้เลือกโปรแกรมดังต่อไปนี้

- *Microsoft Visual C++* version 6.0 [8] เป็นเครื่องมือสำหรับพัฒนาโปรแกรมประยุกต์บนด้วยภาษา C++ ที่นิยมใช้อย่างแพร่หลายและมีความสามารถในด้านการเขียนโปรแกรมเชิงวัตถุสูง และสามารถติดต่อกับ คลังกราฟิก (Graphic Library) ได้หลายระบบ
- *Qt version 2.03* [9] เป็นเครื่องมือสำหรับพัฒนาส่วนต่อประสานกับผู้ใช้ (User Interface) ที่มีเครื่องมือแสดงส่วนโต้ตอบให้เลือกใช้หลายประเภทและสามารถเชื่อมต่อกับคลังกราฟิก (Graphic Library) ได้หลายระบบเช่นกัน อีกทั้งยังสามารถใช้ได้กับระบบปฏิบัติการหลายประเภท (Cross platform)
- *OpenGL version 1.2* [10] เป็นคลังกราฟิก (Graphic Library) ที่สามารถสร้างภาพ 2 มิติและ 3 มิติได้อย่างมี ประสิทธิภาพ โดยสามารถสร้างหุ่นจำลอง 3 มิติได้อีกด้วย อีกทั้งจะเป็นส่วนแสดงผลที่สำคัญของการวิจัยและพัฒนาโปรแกรมทั้งหมด

3.2 การวิเคราะห์ข้อมูลจากเอกสารและงานวิจัยที่เกี่ยวข้อง

จากการสรุปเอกสารและงานวิจัยที่ได้ศึกษามา (*ตาราง 2.1 และ 2.2)* พบว่าส่วนใหญ่มีแนวโน้มของการวิจัยเน้นไปที่การสร้างเครื่องมือวาดเส้นเป็นหลัก และเน้นการวิจัยเรื่องการวาดเส้นแบบเรขาคณิตและวาดเส้นแบบอิสระรองลงมา โดยสามารถสรุปได้ 6 ประการที่สำคัญดังนี้คือ

3.2.1 การพัฒนาเครื่องมือวาดเส้น (User Interface)

3.2.2 การพัฒนาระบบเลเยอร์

3.2.3 การพัฒนาระบบวาดเส้นเรขาคณิตและระบบวาดเส้นอิสระ

3.2.4 การพัฒนาระบบแสดงทัศนียภาพ

3.2.5 การพัฒนาระบบรับรู้วัตถุ

3.2.6 การนำเอาปากกาอิเล็กทรอนิกส์มาใช้ในการวิจัย

---

[8] Microsoft Corporation

[9] Trolltech http://www.trolltech.no

[10] Silicon Graphic Inc. http://www.opengl.org

3.2.7 การพัฒนาระบบสร้างหุ่นจำลอง 3 มิติ

3.2.1 การพัฒนาเครื่องมือวาดเส้น การพัฒนาโปรแกรมให้สามารถรับข้อมูลภาพวาดลายเส้นได้ จะมีประโยชน์อย่างยิ่งกับการออกแบบของสถาปนิก เพราะจะช่วยลดขั้นตอนการทำงานลงจากการวาดภาพลายเส้นลงในกระดาษไปสู่การสแกนกระดาษให้เป็นข้อมูลในคอมพิวเตอร์ แล้วจึงนำเอาไปทำเป็นแบบจริงในโปรแกรม CAD อีกครั้งหนึ่ง และยังช่วยพัฒนาโปรแกรมทางด้านการออกแบบสถาปัตยกรรมให้ใช้งานได้ง่ายขึ้นอีกด้วย (Takeo Igarashi, 1999) โดยหลักเกณฑ์สำคัญที่จะช่วยให้สามารถพัฒนาเครื่องมือวาดเส้นที่เหมาะสมจะต้องหาตัวแปรที่ใช้จำแนกเส้นลักษณะต่างๆ เพื่อแยกเส้นตรง หรือเส้นโค้ง ช่องว่าง ออกจากกัน และจะต้องหาเกณฑ์ในการตีความหมายของเส้นของสถาปนิกส่วนใหญ่ เช่นการวาดเส้นแบบใดที่สถาปนิกต้องการให้ขนานกัน หรือการวาดเส้นลักษณะใดที่สถาปนิกต้องการให้เป็นมุมของภาพ เป็นต้น

ในวิธีการหาตัวแปรและหาเกณฑ์ในการตีความหมายดังกล่าว ได้อ้างอิงข้อมูลจากการวิจัยเรื่อง Architectural Sketch Recognition (Fatos Bengi Durgun, BÜlent ÖzgÜc, 1990) โดยการสัมภาษณ์สถาปนิกและนักศึกษาสถาปัตยกรรมชั้นปีที่ 4 จำนวน 85 คนและให้ทำแบบทดสอบโดยการวาดภาพลายเส้นตามแบบที่กำหนดให้ จากนั้นนำผลสรุปมาเป็นพื้นฐานเพื่อกำหนดเป็นเกณฑ์ที่เหมาะสมในการพัฒนาเครื่องมือวาดเส้นต่อไป จากผลสรุปนี้ทำให้ได้หลักเกณฑ์ในการพัฒนาและเห็นถึงความสำคัญของการพัฒนาเครื่องมือวาดเส้นที่ต้องมีลักษณะที่เหมาะสมกับการวาดภาพลายเส้นของสถาปนิก

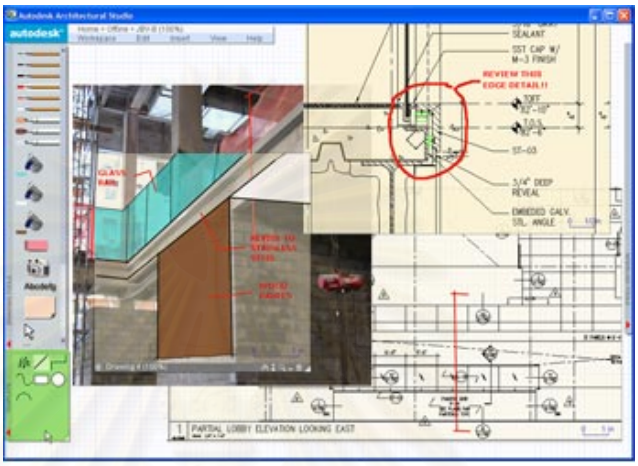
3.2.2 การพัฒนาระบบเลเยอร์ คุณลักษณะที่สามารถทำงานแบบแบ่งเป็นชั้น หรือเลเยอร์เป็นคุณสมบัติที่มีการวิจัยและพัฒนาร่วมกับโปรแกรมวาดภาพลายเส้นน้อยที่สุด สาเหตุน่าจะเป็นเพราะระบบเลเยอร์เป็นคุณสมบัติที่มีใช้กันมานานแล้วโดยใช้กับโปรแกรมช่วยเขียนแบบเช่น AutoCAD[11], FormZ[12], ArchiCAD[13] เป็นต้น ซึ่งระบบเลเยอร์จะมีความเหมาะสมกับการทำงานเขียนแบบมากกว่าเพราะทำให้สามารถมองเห็นข้อมูลจากหลายๆ แผ่นงานได้พร้อมๆ กัน คล้ายกับการทำงานเขียนแบบตามปกติของสถาปนิกที่ใช้กระดาษไขมาวาง

[11] Autodesk http://www.autodesk.com

[12] auto-des-sys http://www.formz.com

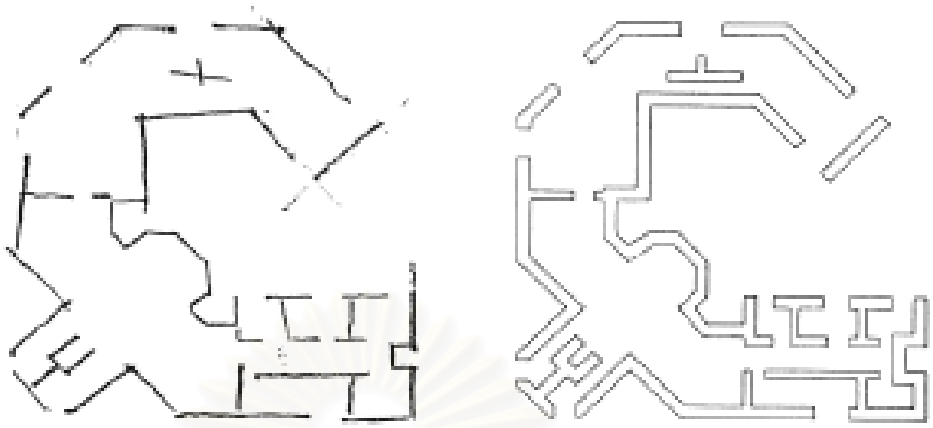
[13] Graphisoft http://www.graphisoft.com

ซ้อนทับกัน ในทางตรงกันข้ามการวาดภาพลายเส้นมักจะเป็นการบันทึกความคิดของผู้ออกแบบ โดยเน้นความฉับไว สะดวกรวดเร็ว โดยที่ข้อมูลอาจไม่สัมพันธ์กันเลยก็ได้ การพัฒนาโปรแกรมเพื่อรับรู้ภาพวาดลายเส้นจึงมีความจำเป็นต้องใช้คุณสมบัตินี้น้อยกว่าคุณสมบัติอื่นๆ



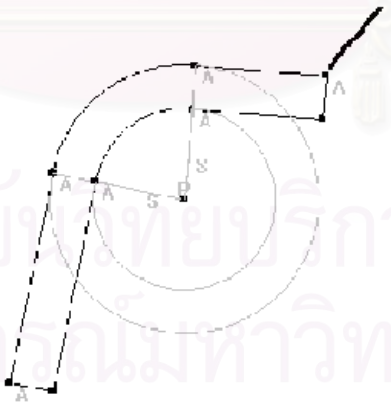
รูปภาพ 3.1 ระบบเลเยอร์มีความเหมาะสมกับการทำงานเขียนแบบที่ต้องแสดงข้อมูลหลายประเภทพร้อมๆ กัน

3.2.3 **การพัฒนาระบบวาดเส้นเรขาคณิตและระบบวาดเส้นอิสระ** คุณลักษณะที่สำคัญรองลงมาคือความสามารถในการวาดเส้นแบบอิสระและการวาดเส้นแบบเรขาคณิต คือความสามารถที่โปรแกรมจะสร้างเส้นที่เหมาะสมกับรูปแบบของการออกแบบสถาปัตยกรรมได้ เพราะเส้นในงานสถาปัตยกรรมที่ใช้กันเป็นหลักคือเส้นแบบเรขาคณิต (Gerard Cesar Gabriel, Mary Lou Maher, 1998) ซึ่งก็มีผู้พัฒนาโปรแกรมมาช่วยออกแบบสถาปัตยกรรมหรือ CAD จำนวนมากให้มีความสามารถในการวาดเส้นแบบนี้อยู่แล้ว ส่วนเส้นแบบอิสระนี้ถือได้ว่าเป็นแนวคิดใหม่ที่กำลังมีการนำมาวิจัยเพื่อพัฒนาโปรแกรมช่วยออกแบบสถาปัตยกรรมที่เหมาะสมต่อไป เพราะเป็นเส้นที่วัดด้วยรูปแบบทางเรขาคณิตได้ยากและมีรูปแบบเฉพาะตัวสำหรับคนวาดแต่ละคน ทำให้มีความแปรผันของเส้นแตกต่างกันมาก ในการวิจัยจึงต้องอาศัยการเก็บข้อมูลจากสถาปนิกเป็นหลัก (Fatos Bengi Durgun, Bülent Özgüc, 1990: 4) โดยเก็บข้อมูลจากสถาปนิกและนักศึกษาสถาปัตยกรรมจำนวน 85 คน เพื่อหาขั้นตอนการวาดภาพลายเส้น การวาดแผนผังและการตีความหมายของภาพวาดลายเส้น ซึ่งจะทำให้ทราบถึงลักษณะที่นำไปใช้ร่วมกันได้ของการวาดลายเส้น และทราบตัวแปรที่จะนำมาวัดค่าความแตกต่างกันของการวาดภาพลายเส้น



รูปภาพ 3.2 ตัวอย่างแผนผัง (ขวา) ที่นำมาใช้เป็นแบบทดสอบสำหรับเก็บวิธีการวาดภาพของสถาปนิก (ซ้าย) เพื่อหาตัวแปรในการวาดภาพทางสถาปัตยกรรม

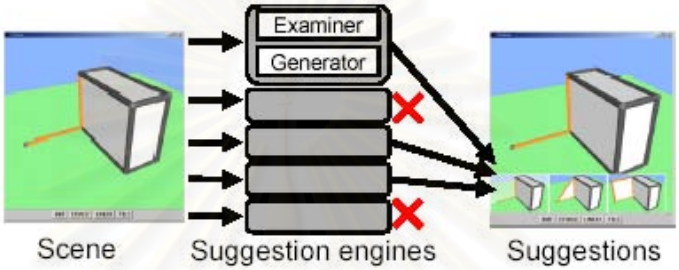
3.2.4 **การพัฒนาระบบแสดงทัศนียภาพ** คุณสมบัติการแสดงทัศนียภาพแบบ 3 มิติ เป็นอีกประเด็นหนึ่งที่มีผู้พัฒนาโปรแกรมเอาไว้เป็นจำนวนมาก โดยนอกจากจะแสดงผลของการวาดภาพลายเส้นให้ถูกต้องแล้วยังมีการประยุกต์ให้เห็นความต่อเนื่องของกระบวนการแปลงเส้นหรือการปรับเปลี่ยนเส้นจากการใช้มือวาดให้เห็นขั้นตอนการเปลี่ยนแปลงให้เป็นเส้นเรขาคณิตทางคอมพิวเตอร์ในทันที เช่นงานวิจัยและพัฒนาโปรแกรม Quick Sketch (Dae-Hyun Kim, Paul Michalik, Stephan Hofer, 2002)



รูปภาพ 3.2 โปรแกรม Quick Sketch จะแปลงภาพวาดลายเส้นให้เป็นข้อมูลทางคอมพิวเตอร์ด้วยวิธีทางเรขาคณิตทันที

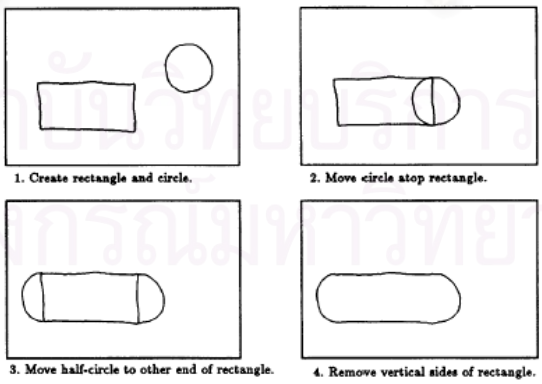
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

หรืออีกแนวทางหนึ่งคือการเน้นพื้นที่วาดภาพมากกว่าการใช้ปุ่มหรือเครื่องมือจำนวนมากเพื่อวาดภาพเช่นงานวิจัยและพัฒนาโปรแกรม Chateau (Takeo Igarashi, 2001) ซึ่งในหน้าจอโต้ตอบกับผู้ใช้จะไม่มีปุ่มคำสั่งให้กด โดยจะต้องวาดภาพลงไปในพื้นที่ทำงานโดยตรง แล้วโปรแกรมจึงจะทำการสร้างทางเลือกในการปรับปรุงภาพวาดหรือหุ่นจำลองที่สร้างขึ้น โดยผู้ใช้สามารถปรับเปลี่ยนทางเลือกในการออกแบบได้โดยการเลือกทางเลือกในขณะนั้น และไม่ต้องมีการจดจำคำสั่งต่างๆ เลย



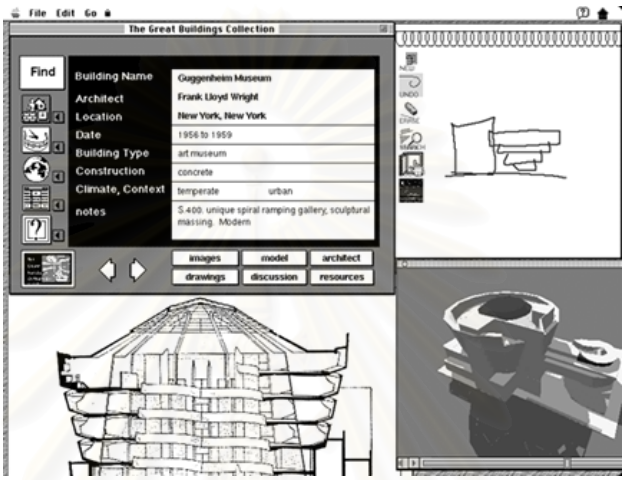
รูปภาพ 3.3 การทำงานของโปรแกรม Chateau จะมีระบบสร้างทางเลือก (Suggestion Engines) ที่คอยตรวจสอบหน้าจอเพื่อสร้างทางเลือกที่สัมพันธ์กับการทำงานในขณะนั้น

3.2.5 **การพัฒนาระบบรับรู้วัตถุ** งานวิจัยที่เน้นการใช้ระบบรับรู้วัตถุเพื่อแปลงเส้นจากการวาดภาพลายเส้นเพื่อให้เป็นข้อมูลทางคอมพิวเตอร์ เป็นแนวคิดที่ต้องการให้ระบบรับรู้วัตถุที่เป็นภาพวาดเพื่อช่วยให้การทำงานสะดวกรวดเร็วขึ้น เช่นโปรแกรม Persketch (Eric Saund, Thomas Moran, 1994) ที่พัฒนาโปรแกรมให้ตอบสนองต่อการรับรู้ของผู้ใช้งานโดยใช้หลักของเรขาคณิตสำหรับแปลงภาพวาดลายเส้นให้เป็นภาพที่ใกล้เคียงกับการรับรู้มากที่สุด



รูปภาพ 3.4 ลำดับการทำงานอย่างง่ายของโปรแกรม Persketch หลังจากวาดภาพ (1) แล้วจะเข้าสู่กระบวนการแปลง (2-4) ซึ่งระบบนี้โปรแกรมวาดภาพทั่วไปไม่สามารถทำได้

อีกโปรแกรมหนึ่งที่เป็นที่นิยมได้แก่ Electronic Cocktail Napkin ซึ่งเป็นการพัฒนาโปรแกรมเพื่อตีความหมายของภาพวาดลายเส้น โดยใช้ระบบเรขาคณิตแปลงภาพวาดลายเส้นให้เป็นข้อมูลทางคอมพิวเตอร์เพื่อนำมาเปรียบเทียบกับข้อมูลจากระบบฐานข้อมูลหากตรงกันก็จะสามารถเลือกเอาข้อมูลรูปภาพหรือข้อมูลอื่นๆ ที่เกี่ยวข้องกันขึ้นมาแสดงได้



รูปภาพ 3.5 โปรแกรม Electronic Cocktail Napkin ลักษณะการนำโปรแกรมไปประยุกต์ใช้กับการค้นหาข้อมูลด้วยการวาดภาพลายเส้น

การพัฒนาโปรแกรมที่มีคุณสมบัติการรับรู้วัตถุจึงมีประโยชน์เป็นอย่างยิ่ง เพราะสามารถช่วยให้การทำงานของสถาปนิกสะดวกรวดเร็วขึ้นโดยช่วยลดขั้นตอนการแปลงภาพวาดในกระดาษให้เป็นข้อมูลทางคอมพิวเตอร์ และช่วยปรับภาพวาดลายเส้นให้เหมาะสมกับการทำงานออกแบบสถาปัตยกรรมเพื่อให้สามารถภาพวาดลายเส้นนั้นไปพัฒนาเป็นงานออกแบบขั้นต่อไปได้ทันที

3.2.6 การนำเอาปากกาอิเล็กทรอนิกส์มาใช้ในงานวิจัย จากการศึกษาพบว่ามีงานวิจัยหลายชิ้นที่นำปากกาอิเล็กทรอนิกส์มาใช้เป็นเครื่องมือนำเข้าข้อมูล (input) ของการวาดภาพลายเส้น อะไรเป็นสาเหตุทำให้มีความสนใจที่จะนำอุปกรณ์ชนิดนี้มาใช้เพิ่มขึ้น

จากการศึกษางานวิจัยเรื่อง Free Form User Interface (Takeo Ikarashi, 1999: 27) พบว่ามีความเกี่ยวข้องกันระหว่างอุปกรณ์ input กับการวาดภาพลายเส้นบนคอมพิวเตอร์ เพราะมีการวิจัยกันมายาวนานในอดีตทั้งในด้าน Software และ Hardware ที่นำเอาการใช้อุปกรณ์ input ด้วยปากกาอิเล็กทรอนิกส์และโปรแกรมคอมพิวเตอร์มาช่วยสนับสนุนการ

ทำงานด้านต่างๆ เช่น electronic whiteboard สำหรับการนำเสนองาน งานด้านการแพทย์เพื่อใช้จดบันทึกประวัติทางการแพทย์ของคนไข้ อุปกรณ์ช่วยเก็บข้อมูลส่วนตัว หรือ PDA ที่สามารถรับรู้ข้อมูลจากปากกาหรือ Stylus แล้วแปลงให้เป็นตัวอักษรทางคอมพิวเตอร์ และในปัจจุบันที่มีคอมพิวเตอร์แบบ Tablet PC ที่เน้นการใช้ปากกาเป็นเครื่องมือ input ข้อมูลแทนการใช้ Keyboard แบบเดิม เหล่านี้คือแนวโน้มที่สำคัญที่จะนำไปสู่การพัฒนาโปรแกรมทางด้านการวาดภาพลายเส้นที่นำเอาปากกาอิเล็กทรอนิกส์มาใช้เป็นเครื่องมือ input ข้อมูล เพราะการใช้ปากกาอิเล็กทรอนิกส์จะช่วยให้ผู้ใช้งานสามารถวาดภาพได้เหมือนการใช้ปากกา โดยเฉพาะการออกแบบสถาปัตยกรรมที่ใช้ปากกาเป็นเครื่องมือพื้นฐาน สถาปนิกจะสามารถวาดภาพด้วยการใช้ปากกาอิเล็กทรอนิกส์ได้ทันที การพัฒนาโปรแกรมวาดภาพลายเส้นโดยใช้ปากกาอิเล็กทรอนิกส์มาเป็นเครื่องมือ input ข้อมูลจึงเป็นวิธีที่จะสนับสนุนการทำงานของสถาปนิกให้สะดวกขึ้น

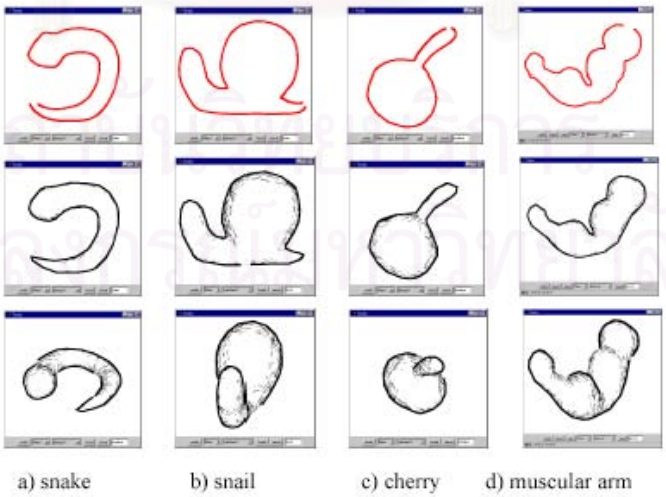
a) Hand held PDA (Sharp Zaurus [160]) b) Pen-based portable computer (Mitubishi AMiTY [2])

รูปภาพ 3.6 ตัวอย่างอุปกรณ์รับข้อมูลด้วยปากกาอิเล็กทรอนิกส์สำหรับงานเฉพาะด้าน เช่นการแพทย์ อุตสาหกรรม

รูปภาพ 3.7 ตัวอย่างการใช้อุปกรณ์รับข้อมูลด้วยปากกาอิเล็กทรอนิกส์สำหรับงานเก็บข้อมูลส่วนบุคคลเช่น PDA (แหล่งที่มา: http://www.sony.com)

รูปภาพ 3.8 อุปกรณ์รับข้อมูลด้วยปากกาอิเล็กทรอนิกส์สำหรับงานออกแบบและงานกราฟิก (แหล่งที่มา: http://www.wacom.com)

3.2.7 การพัฒนาระบบสร้างหุ่นจำลอง 3 มิติ มีการวิจัยและพัฒนาระบบสร้างหุ่นจำลอง 3 มิติมาตั้งแต่ปี ค.ศ. 1977 โดยกลุ่ม ACM SIGGRAPH (James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes, 2003: 16) และในปีเดียวกันก็มีการพัฒนาโปรแกรมชื่อ CATIA เพื่อช่วยในงานอุตสาหกรรมที่ต้องการการสร้างหุ่นจำลอง 3 มิติ (ธิดาสิริ ภัทรากาญจน์, 2546: 65) การพัฒนาโปรแกรมให้สามารถสร้างหุ่นจำลอง 3 มิติได้ จึงไม่ใช่เรื่องใหม่ แต่ในการพัฒนาโปรแกรมที่เน้น*การช่วยวาดภาพลายเส้นให้สามารถสร้างหุ่นจำลอง 3 มิติ*ได้ เพิ่งได้รับความสนใจเมื่อไม่นานมานี้ เริ่มจากโปรแกรมชื่อ Teddy โดย Takeo Igarashi (Takeo Ikarashi, 1999) โดยผู้ใช้งานสามารถวาดภาพลายเส้นแบบอิสระเพื่อสร้างหุ่นจำลองได้ทันที



รูปภาพ 3.9 งานวิจัย Teddy โดย Takeo Igarashi เน้นการสร้างหุ่นจำลอง 3 มิติจากการวาดภาพลายเส้น

ในการวาดภาพลายเส้นหากสามารถนำเอาภาพลายเส้นมาสร้างเป็นหุ่นจำลองได้แล้วจะทำให้เกิดความสะดวกในการทำงานมากยิ่งขึ้น และทำให้สามารถพัฒนาแนวคิดในการออกแบบไปพร้อมๆ กันกับการวาดภาพลายเส้นอีกด้วย เพราะในการออกแบบร่างของสถาปนิกจะมีการทำหุ่นจำลองสำหรับการศึกษาการออกแบบ (ธิดาศิริ ภัทรากาญจน์, 2546: 78) เพื่อใช้เป็นเครื่องมือสำหรับพัฒนาและทดสอบความคิดในลักษณะ 3 มิติ

ฉะนั้นเมื่อมองจากแนวโน้มในการพัฒนาโปรแกรมทางด้าน CAD และประโยชน์ในการช่วยเหลือสถาปนิกออกแบบและทดสอบความคิดแล้ว ทำให้เห็นถึงความจำเป็นที่จะต้องพัฒนาให้โปรแกรมมีคุณสมบัติในการวาดภาพลายเส้นและสามารถสร้างหุ่นจำลอง 3 มิติไปพร้อมๆ กัน

3.3 การกำหนดตัวแปรเพื่อนำมาใช้ในการพัฒนาโปรแกรม

หลังจากการวิเคราะห์ข้อมูลพื้นฐานในการพัฒนาโปรแกรมแล้ว ทำให้สามารถกำหนดแนวทางที่เหมาะสมในการพัฒนาโปรแกรมได้ และในการพัฒนาโปรแกรมตามแนวทางเหล่านี้จำเป็นต้องมีการกำหนดตัวแปร เช่น การพัฒนาระบบรับรู้วัตถุจำเป็นจะต้องมีการกำหนดตัวแปรในการวัดรูปแบบของภาพวาดลายเส้นที่สามารถรับรู้วิธีการวาดภาพลายเส้นของสถาปนิกส่วนใหญ่ได้ เป็นต้น ในการกำหนดตัวแปรที่เหมาะสมจำเป็นต้องอาศัยข้อมูลจากการวิเคราะห์และอาศัยทฤษฎีที่เกี่ยวข้อง โดยสามารถแบ่งประเภทของตัวแปรที่สำคัญได้ดังนี้

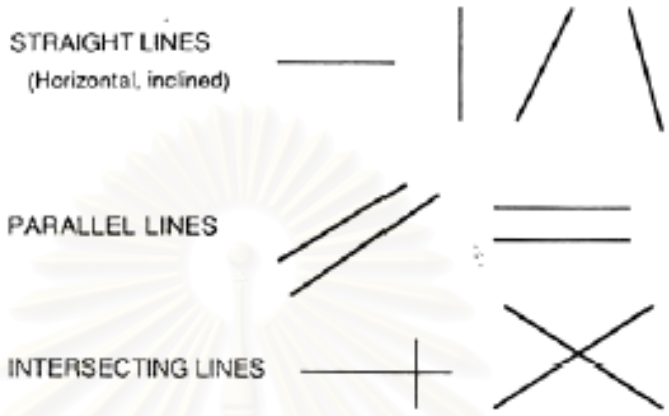
3.3.1 ตัวแปรในการพัฒนาระบบวาดเส้นเรขาคณิตและวาดเส้นอิสระ

3.3.2 ตัวแปรในการพัฒนาระบบรับรู้วัตถุ

3.3.1 ตัวแปรในการพัฒนาระบบวาดเส้นเรขาคณิตและวาดเส้นอิสระ

ในการวาดเส้นเรขาคณิตและวาดเส้นอิสระโดยทั่วไปสถาปนิกสามารถวาดเส้นได้หลายรูปแบบตามแต่ความถนัดของแต่ละบุคคลทำให้มีรูปแบบของเส้นที่หลากหลายยากแก่การกำหนดตัวแปร แต่ในการวาดเส้นที่เกี่ยวข้องกับการออกแบบสถาปัตยกรรมพบว่ามีการใช้เส้นทั้งสองแบบเพื่อสื่อความหมายถึงภาพแผนผังหรืองานออกแบบเขียนแบบที่มีลักษณะของเส้นที่มีลักษณะเป็นเส้นตรง เส้นขนานกันหรือเส้นตัดกันเป็นส่วนใหญ่ (Fatos Bengi Durgun, Bülent Özgüc, 1990: 6) ดังนั้นในงานวิจัยนี้จึงเลือกวิธีพัฒนาระบบวาดเส้นให้เหมาะกับรูปแบบการวาดเส้นทางสถาปัตยกรรม โดยสามารถกำหนดตัวแปรหลักในการพัฒนาระบบวาดเส้นได้ดังนี้

- *ตัวแปรในการวัดความขนานกันของเส้น*
- *ตัวแปรในการวัดความยาวของเส้น*



รูปภาพ 3.10 การพัฒนาระบบวาดวาดเส้นเพื่อรองรับการวาดเส้นที่นิยมใช้ในงานสถาปัตยกรรม

- *ตัวแปรในการวัดความขนานกันของเส้น* จากงานวิจัยของ Fatos Bengi Durgun และ Bülent Özgüc พบว่าในการวาดภาพของสถาปนิกความชันของเส้นจะเป็นตัวแปรหลักที่กำหนดว่าเส้นตรงสองเส้นขนานกันหรือไม่ และตัวแปรนี้ยังสามารถนำไปวัดการทับกันของเส้น หรือการต่อกันของเส้นได้อีกด้วย

- *ตัวแปรในการวัดความยาวของเส้น* คือการหาค่าระยะต่อของเส้น (extend) หรือระยะที่ขาดไป (missing) ของเส้นเมื่อเทียบกับความยาวทั้งหมด โดยสามารถนำไปหาความยาวที่เหมาะสมของเส้น ที่ผู้วาดต้องการวาดจริงๆ ได้ จากการทดสอบพบว่า ค่าความยาวของเส้นมีตัวแปรหลักคือ จุดที่ตัดกันของเส้น และมีตัวแปรตามคือระยะห่างระหว่างจุดต้นและจุดปลายของเส้นวัดจากจุดที่ตัดกัน

3.3.2 ตัวแปรในการพัฒนาระบบรับรู้วัตถุ

ในการรับรู้วัตถุโปรแกรมจะต้องสามารถจำแนกรูปแบบของวัตถุหรือภาพวาดลายเส้นลักษณะต่างๆ กันได้ ในการพัฒนาระบบรับรู้จะต้องมีความสัมพันธ์กับตัวแปรหลักคือเครื่องมือพื้นฐานที่เลือกไว้ในการพัฒนาโปรแกรมทั้ง Hardware และ Software เพราะจะต้องพัฒนาระบบ Software ให้สามารถรับข้อมูลจากอุปกรณ์ input ซึ่งได้แก่ Mouse หรือ ปากกาอิเล็กทรอนิกส์

จากนั้น Software จะสร้างจุดขึ้นโดยอัตโนมัติตามรอบการทำงานของคอมพิวเตอร์ และระบบรับรู้วัตถุจะเริ่มทำงานตั้งแต่มีจุดเกิดขึ้นเป็นต้นไป จากการวิจัยสามารถแบ่งตัวแปรที่เกี่ยวข้องกับการระบบรับรู้วัตถุได้ดังนี้

- ตัวแปรของระบบรับรู้เส้นตรง
- ตัวแปรของระบบรับรู้ความต่อเนื่องหรือการทับกันของเส้น
- ตัวแปรของระบบรับรู้การตัดกันของเส้น
- ตัวแปรของระบบรับรู้ระยะห่างระหว่างเส้น
- ตัวแปรของระบบรับรู้ระยะยื่นของเส้น
- ตัวแปรของระบบรับรู้จุดมุมของภาพ

- *ตัวแปรของระบบรับรู้เส้นตรง* เป็นระบบรับรู้ขั้นพื้นฐานที่สำคัญเพราะเป็นส่วนสำคัญของภาพวาดลายเส้นทางสถาปัตยกรรม (Fatos Bengi Durgun, Bulent Ozguc, 1990: page 8) โดยจะเริ่มขั้นตอนการรับรู้ตั้งแต่การจรดปากกาลงบนพื้นที่วาดรูปจนถึงการยกปากกาขึ้น ทำให้สามารถกำหนดตัวแปรหลัก คือ ความชันระหว่างจุดสองจุด และมีตัวแปรตามคือจำนวนจุดที่เกิดขึ้นระหว่างการจรดปากกาลงและยกปากกาขึ้นโดยนำพิกัดของจุดสองจุดจากการลากปากกา มาหาความชันตามสูตรเรขาคณิตตามวิธีการวาดเส้นตรงแบบดีดีเอ (กิตติมา มะแก้ว, 2538: 12) โดยที่จำนวนจุดที่นำมาหาความชันนี้จะเกิดขึ้นระหว่างการจรดปากกาลง (Mouse Down)และยกปากกา (Mouse Up) ขึ้น

$$M = \frac{Y2-Y1}{X2-X1}$$

M คือ ความชันของเส้นตรงที่มีจุดต้น (X1,Y1) และจุดปลาย (X2,Y2)

หลังจากแทนพิกัดทั้งสองของจุดลงในสูตรจะได้ความชันของเส้นที่หนึ่ง M1 จากนั้นจึงย้อนกลับไปแทนค่าจุดสองจุดถัดไปในสูตรเรขาคณิตเพื่อหาความชันของเส้นที่สอง M2 แล้วจึงนำเอาค่าความชันมาเปรียบเทียบเพื่อหาความแตกต่างของความชันว่ามีค่าไม่เกินค่าที่กำหนดหรือไม่ ด้วยการใช้สูตรต่อไปนี้ (Fatos Bengi Durgun, Bulent Ozguc, 1990: page 8)

$$ABS(M2-M1) \leq \Delta Pi$$

M1 คือความชันของเส้นตรงเส้นแรก

M2 คือความขันของเส้นตรงเส้นที่สอง

ΔPi คือค่าความชันที่กำหนดให้ไม่เกินค่านี้

หลังจากแทนค่าในสูตรแล้วจะสามารถรับรู้ได้ว่าจุดที่ได้จากการวาดอยู่บนเส้นตรงเดียวกันหรือไม่ หาก M2-M1 มีค่าไม่เกิน ΔPi (กำหนดแบบ empirical ให้เป็น 5)

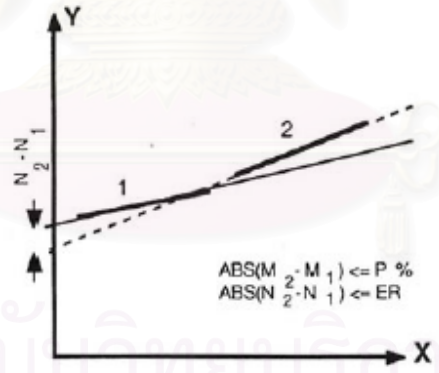
- *ตัวแปรของระบบรับรู้ความต่อเนื่องกันหรือทับกันของเส้น* จะมีวิธีการรับรู้เหมือนกับระบบรับรู้เส้นตรงโดยมีตัวแปรหลักคือค่าความแตกต่างของความชันของเส้นสองเส้น ΔPi (กำหนดให้เป็น 5) และค่าความชันของเส้นตรงสองเส้น โดยสามารถหาความชันของเส้นด้วยการนำเอาพิกัดของเส้นมาแทนค่าในสูตร

$$M = \frac{Y2-Y1}{X2-X1}$$

M คือ ความชันของเส้นตรงที่มีจุดต้น (X1,Y1) และจุดปลาย (X2,Y2)

เมื่อได้ความชันของเส้นแรก M1 แล้วจึงนำเอาพิกัดจากเส้นถัดไปมาหาค่าความชัน M2 แล้วจึงนำค่ามาแทนในสูตร

$$ABS(M2-M1) \leq \Delta Pi \text{ (กำหนดให้เป็น 5)}$$



รูปภาพ 3.11 ภาพแสดงตัวอย่างการเปรียบเทียบเส้นสองเส้นโดยมีค่าความชัน M1 และ M2 ตามลำดับ

จากการแทนค่าในสูตรจะทำให้สามารถรับรู้ได้ว่าเส้นตรงสองเส้นมีความต่อเนื่องกันหรือทับกันหรือไม่

- *ตัวแปรของระบบรับรู้การตัดกันของเส้น* เป็นระบบที่ใช้รับรู้จุดพิกัดของการตัดกันของเส้นสองเส้น โดยมีตัวแปรหลักคือความชันของเส้นสองเส้นที่ตัดกันและค่า Y intercept ของเส้น

ตรงแต่ละเส้น โดยมีตัวแปรตามคือค่า X และ Y ของจุดที่ตัดกันโดยมีวิธีการหาค่า Y intercept ได้โดยการแทนค่าพิกัดของเส้นแต่ละเส้นลงในสูตร

$$C = Y - MX$$

C คือค่า Y intercept หรือค่าการตัดแกน Y ของสมการเส้นตรง

Y คือ พิกัด Y ของเส้นตรง

X คือ พิกัด X ของเส้นตรง

M คือ ความชันของเส้นตรง

และแทนค่าพิกัดของเส้นแต่ละเส้นลงในสูตรหาความชัน

$$M = \frac{Y2-Y1}{X2-X1}$$

เมื่อได้ค่า C1 และ C2 และ M1 และ M2 ของเส้นสองเส้นแล้วจึงนำมาแทนค่าในสูตรหาจุดพิกัด X

$$X = (C2 - C1)/(M1 - M2)$$

C1,C2 คือค่า Y intercept หรือค่าการตัดแกน Y ของสมการเส้นตรง 2 เส้น

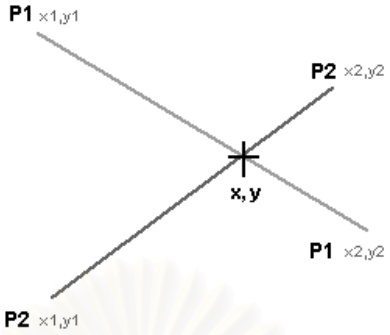
M1,M2 คือ ความชันของเส้นตรงสองเส้น

X คือ จุดตัดของเส้นตรงทั้งสองเส้น

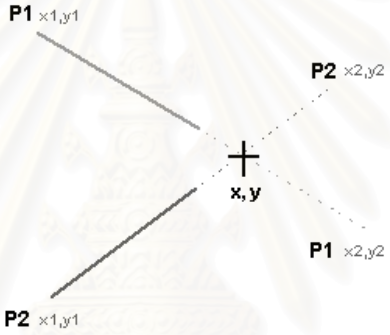
จะได้ค่าพิกัด X ของจุดตัดแล้วจึงนำมาแทนค่าในสูตร C = Y – MX ของเส้นใดเส้นหนึ่งเพื่อหาค่าพิกัด y ของจุดตัด จะได้ค่าจุดพิกัดของจุดตัดของเส้นสองเส้นคือ X, Y

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

หมายเหตุ: จุดพิกัด X และ Y ของจุดตัดที่ได้นั้นอาจเป็นจุดตัดที่ไม่ได้อยู่บนส่วนของเส้นตรงทั้งสองเส้นก็เป็นได้เนื่องจากวิธีการหาค่าจุดตัดนี้เป็นการหาจุดตัดจากสมการเส้นตรง ฉะนั้นจุดตัดที่เกิดขึ้นจึงเป็นจุดที่อยู่บนเส้นตรงเดียวกันกับเส้นทั้งสองแต่ไม่จำเป็นต้องอยู่บนส่วนของเส้นตรงทั้งสองก็เป็นได้ (รูปภาพ 3.12)



รูปภาพ 3.12 ภาพแสดงจุดตัดของเส้นสองเส้นโดยที่จุด x, y อยู่บนส่วนของเส้นตรงทั้งสองเส้น



รูปภาพ 3.13 ภาพแสดงจุดตัดของเส้นสองเส้นโดยที่จุด x, y ไม่ได้อยู่บนส่วนของเส้นตรงใดเลย

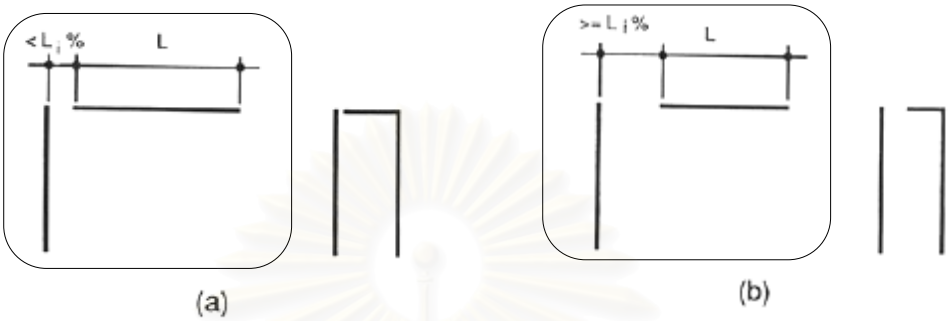
เมื่อเกิดกรณีตามภาพ 3.12 จึงต้องมีวิธีการรับรู้เพื่อหาว่าจุดตัด X, Y อยู่บนส่วนของเส้นตรงที่ตัดกันหรือไม่ โดยแทนค่าจุดตัด X = X, X2 = Y และค่า X1 = X1 (ของเส้นตรงใดก็ได้) ลงในสูตร

$$t = (X - X1)/(X2 - X1)$$

t คือตัวแปรที่ใช้วัดจุดพิกัด X ว่าอยู่บนส่วนของเส้นตรงหรือไม่

จากสูตรจะได้ค่า t หากค่า t อยู่ระหว่าง 0 และ 1 ($0 \leq t \leq 1$) แสดงว่าจุดตัด X, Y จะอยู่บนส่วนของเส้นตรงที่ตัดกัน หากค่า t อยู่นอกเหนือจากนี้แสดงว่า จุดตัด X, Y ไม่ได้อยู่บนส่วนของเส้นตรงใดเลยและแสดงว่าเส้นตรงสองเส้นไม่ได้ตัดกันอย่างแท้จริง จากการหาค่า t จึงสามารถเลือกเอาค่าพิกัด X, Y ที่ถูกต้องมาใช้เป็นจุดตัดของเส้นต่อไป

- *ตัวแปรของระบบรับรู้ระยะห่างระหว่างเส้น* คือวิธีการรับรู้จุดปลายของเส้นสองเส้นที่อยู่ใกล้กัน เพื่อวิเคราะห์ว่าผู้วาดต้องการเขียนเส้นที่เป็นช่องว่างหรือเขียนเส้นที่มีจุดปลายชนกัน

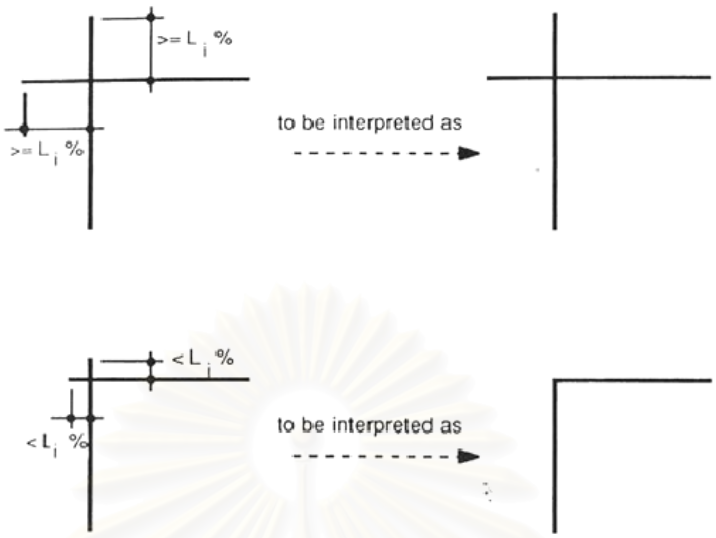


รูปภาพ 3.14 ภาพ (a) แสดงการวาดเส้นที่มีการเว้นระยะห่างน้อยกว่าค่าที่กำหนด (Li%)
ภาพ (b) แสดงระยะที่มากกว่าค่าที่กำหนด (Li%)

ในการรับรู้จะใช้ค่า Li% เป็นตัวแปรหลัก (กำหนดแบบ empirical ให้เท่ากับ 40) เทียบกับตัวแปรระยะห่างของจุดปลายทั้งสอง ถ้ามีค่าน้อยกว่า Li% แสดงว่าผู้วาดต้องการวาดเส้นให้จุดปลายต่อกัน แต่ถ้าหากระยะห่างมากกว่า Li% แสดงว่าผู้วาดต้องการเว้นระยะห่างระหว่างจุดปลายทั้งสอง การรับรู้ระยะห่างระหว่างจุดปลายของเส้นจะมีประโยชน์ในการสร้างจุดมุมของภาพและการสร้างเว้นระยะเพื่อสร้างช่องเปิด

- *ตัวแปรระบบรับรู้ระยะยื่นของเส้น* เป็นวิธีการรับรู้ความยาวของเส้นจากจุดปลายที่ตัดกันของเส้นสองเส้น เพื่อวิเคราะห์ว่าผู้วาดต้องการวาดเส้นให้ยื่นออกไปหรือต้องการวาดเส้นให้มีจุดปลายชนกัน

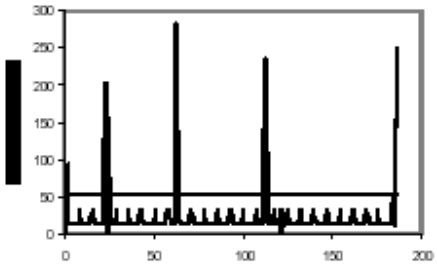
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปภาพ 3.15 ภาพแสดงรูปแบบของการวาดเส้นที่ผู้วาดจะลากเส้นยื่นออกไปจากจุดตัดของเส้นสองเส้น

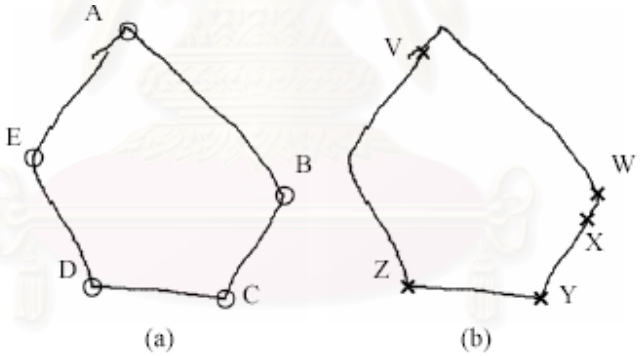
จากภาพ 3.15 สามารถตีความหมายได้สองลักษณะคล้ายกับวิธีการรับรู้ระยะห่างระหว่างเส้น (Fatos Bengi Durgun, Bulent Ozguc, 1990: page 12) คือ แบบแรกผู้วาดต้องการลากเส้นให้มีระยะยื่นออกมา และแบบที่สองคือการวาดโดยต้องการให้จุดที่ตัดกันเป็นจุดมุมของภาพ โดยใช้ตัวแปรหลักคือ Li% เป็นตัวกำหนดระยะที่เหมาะสมของการยื่นของเส้น ในงานวิจัยนี้จะกำหนดค่า Li% แบบ empirical ให้เป็น 40

- *ตัวแปรของระบบรับรู้จุดมุมของภาพ* ในการวาดภาพลายเส้นแบบเรขาคณิตสามารถรับรู้จุดมุมของภาพด้วยวิธีการรับรู้ระยะห่างและระยะยื่นของจุดปลายของเส้นได้ แต่ในการวาดภาพลายเส้นแบบอิสระ (Free Hand) ผู้วาดสามารถวาดเส้นได้หลากหลายรูปแบบจนกำหนดตัวแปรในการรับรู้ได้ยาก แต่จากงานวิจัยของ Peter Agar และ Kevin Novins ทำให้สามารถรับรู้รูปแบบของการวาดเส้นแบบอิสระและนำมาใช้กับการวาดภาพลายเส้นทางสถาปัตยกรรมได้ คือการรับรู้จุดมุมของภาพวาดรูปหลายเหลี่ยม (Polygon) โดยวิธีการรับรู้จุดมุมของภาพสำหรับการวาดเส้นแบบอิสระจะเกี่ยวข้องกับตัวแปรหลักคือ ระยะเวลาในการวาดเส้นโดยวัดระหว่างจุดสองจุดที่เกิดจากการลากปากกาอิเล็กทรอนิกส์ และความแตกต่างของระยะเวลาเป็นตัวแปรตาม (Peter Agar, Kevin Novins, 2003: 148)



รูปภาพ 3.16 ภาพแสดงระยะเวลาในการวาดภาพรูปหลายเหลี่ยมจะเห็นความแตกต่างของเวลาที่เพิ่มขึ้นเมื่อถึงจุดมุมของภาพ

การเก็บค่าตัวแปรระยะเวลาในการวาดเส้นจะกระทำโดยการเก็บค่าระยะเวลาที่ใช้ระหว่างจุดสองจุดโดยเก็บค่าเป็น 1/1000 วินาที (millisecond) จากนั้นจะนำเอาตัวแปรระยะเวลามาเปรียบเทียบกันเพื่อเก็บค่าตัวแปรความแตกต่างของระยะเวลา ระยะเวลาทั้งสองจุดมีความแตกต่างกันไม่เกินค่าที่กำหนด (กำหนดแบบ empirical ให้เป็น 180) จะถือว่าไม่ใช่จุดมุมของภาพ แต่ถ้าหากมีความแตกต่างกันมากกว่าค่าที่กำหนดจะถือว่าเป็นจุดมุมของภาพ



รูปภาพ 3.17 ภาพ (a) แสดงการวาดภาพรูปหลายเหลี่ยมโดยผู้วาดต้องการให้จุดที่มีวงกลมคือจุดมุมของภาพและภาพ (b) คือการกำหนดจุดมุมด้วยวิธีการหาความแตกต่างของเวลา

3.4 **การออกแบบและพัฒนาโปรแกรม** จากแนวทางการพัฒนาโปรแกรมและขั้นตอนวิธีในการรับรู้ที่ศึกษามาทั้งหมด ทำให้สามารถออกแบบโปรแกรมที่มีขั้นตอนวิธีในการรับรู้ภาพวาดลายเส้นได้ โดยออกแบบให้โปรแกรมนี้มีการทำงานที่ตอบสนองต่อวิธีการวาดภาพของสถาปนิก ซึ่งมีรูปแบบของเส้นที่มีลักษณะเฉพาะตัว และโปรแกรมนี้จะต้องสามารถรับรู้ภาพวาดลายเส้นทั้งแบบเรขาคณิตและแบบ

อิสระได้โดยมีความถูกต้องในระดับหนึ่ง และใช้ขั้นตอนวิธีการรับรู้ที่เหมาะสมกับการทำงานโดยยึดจากข้อมูลที่ศึกษามา แบ่งการพัฒนาโปรแกรมออกเป็น 2 ขั้นตอน ได้แก่

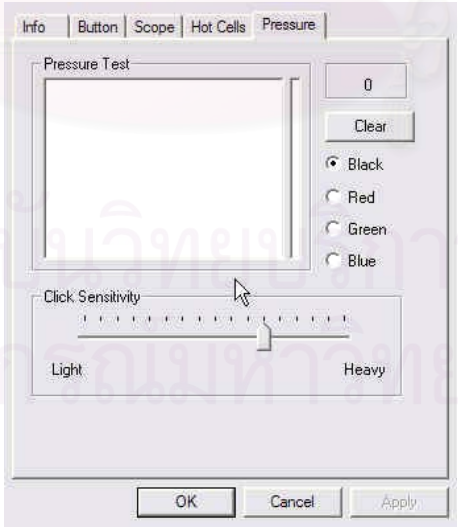
3.4.1 การพัฒนาส่วนรับข้อมูล input จากการวาดภาพลายเส้น

3.4.2 การพัฒนาโปรแกรมด้วยวิธีรับรู้ข้อมูลจากภาพวาดลายเส้น

3.4.3 การพัฒนาส่วนโต้ตอบผู้ใช้

3.4.1 การพัฒนาส่วนรับข้อมูล input จากการวาดภาพลายเส้น

จากการศึกษางานวิจัยและโปรแกรมที่เกี่ยวข้องทำให้เลือกระบบ input ข้อมูลด้วยปากกาอิเล็กทรอนิกส์มาใช้ร่วมกับการพัฒนาโปรแกรม และจากการเลือกอุปกรณ์ input ข้อมูลชนิดนี้ทำให้มีตัวแปรในการพัฒนาโปรแกรมเพิ่มขึ้นนั่นคือ ตัวแปรแรงกดของปากกาในการวาดภาพ (Pen Pressure) เนื่องจากปากกาอิเล็กทรอนิกส์แต่ละชนิดจะมี Software และการตั้งค่าแรงกดของปากกาแตกต่างกัน บางชนิดสามารถปรับแรงกดได้มากบางชนิดปรับได้น้อย โดยมีผลทำให้จำนวนจุดที่เกิดขึ้นจากการวาดภาพมีความถี่สัมพันธ์กับแรงกดของปากกาและทำให้ตัวแปรที่ได้จากพิกัดของจุดมีจำนวนมากจนทำให้ไม่สามารถนำมาวิเคราะห์ได้ ในงานวิจัยนี้จึงกำหนดให้ใช้ปากกาอิเล็กทรอนิกส์ที่สามารถปรับแรงกดได้ในระดับ 70% เพื่อให้จุดที่เกิดจากการวาดภาพมีความถี่ที่สามารถนำมาวิเคราะห์ได้และมีความใกล้เคียงกับการวาดภาพลายเส้นในกระดาษได้ในระดับหนึ่ง



รูปภาพ 3.18 ภาพแสดงการตั้งค่าแรงกดของปากกาในระดับ 70% โดยใช้ปากกาอิเล็กทรอนิกส์ของ Genius รุ่น Wizard Pen 4x3

และเนื่องจากความถี่ของจุดที่เกิดขึ้นยังขึ้นอยู่กับตัวแปรในด้านความเร็วของคอมพิวเตอร์อีกด้วย เพราะความถี่ของจุดที่ได้รับค่าจากปากกาจะขึ้นอยู่กับจำนวนกรอบภาพที่คอมพิวเตอร์สร้างขึ้นในเวลา 1 วินาที ในงานวิจัยนี้จึงกำหนดให้ใช้คอมพิวเตอร์ PC ที่มีหน่วยประมวลผลกลางความเร็ว 1.8 Ghz หน่วยความจำแบบ DDR 256 MB และต้องมีกราฟิกการ์ดที่รองรับ OpenGL 1.2 ใช้ระบบปฏิบัติการ WindowsXP

3.4.2 การพัฒนาโปรแกรมด้วยวิธีรับรู้ข้อมูลจากภาพวาดลายเส้น

ในการพัฒนาโปรแกรมจะเริ่มจากการพัฒนาขั้นตอนวิธีทางคอมพิวเตอร์โดยนำเอาขั้นตอนวิธีในการรับรู้ภาพวาดลายเส้นมาใช้ และนำตัวแปรที่เกี่ยวข้องมาประกอบกับการพัฒนาโปรแกรม ซึ่งสามารถแบ่งวิธีการพัฒนาโปรแกรมตามวิธีรับรู้ข้อมูลได้ดังนี้

- การพัฒนาระบบรับรู้เส้นตรงจากการวัดความต่อเนื่องหรือการทับกันของเส้น
- การพัฒนาระบบรับรู้การตัดกันของเส้น
- การพัฒนาระบบรับรู้ระยะห่างระหว่างเส้น
- การพัฒนาระบบรับรู้ระยะยื่นของเส้น
- การพัฒนาระบบรับรู้จุดมุมของภาพ
- การพัฒนาระบบรับรู้รูปแบบที่เหมือนกันของภาพวาดลายเส้น

- *การพัฒนาระบบรับรู้เส้นตรงจากการวัดความต่อเนื่องหรือทับกันของเส้น*

จากการกำหนดตัวแปรในการรับรู้เส้นตรงสามารถพัฒนาขั้นตอนวิธีในการรับรู้ได้ดังนี้ โปรแกรมจะรับค่าพิกัดของจุดจากการวาดเส้น จากนั้นจะนำมาคำนวณหาความชัน M ตามสูตรหาความชัน

$$M = \frac{Y2-Y1}{X2-X1}$$

M คือความชันของเส้นตรงที่มีจุดต้น (X1,Y1) และจุดปลาย (X2,Y2)

แล้วจึงนำความชันที่ได้ไปเปรียบเทียบในสูตร

$$ABS(M2-M1)<=PARALLEL$$

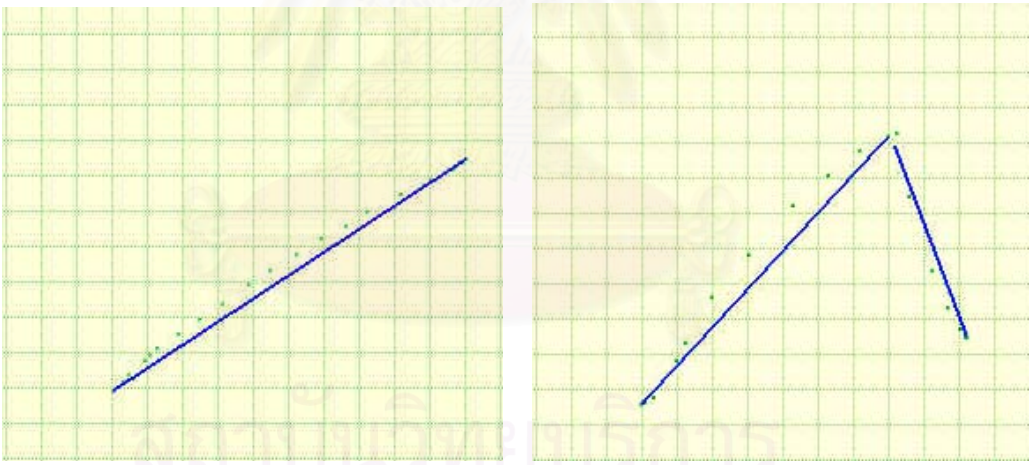
M1,M2 คือความชันของเส้นตรงสองเส้นที่ต้องการวัด

PARALLEL คือค่าที่ใช้วัดการขนานกัน

โดยในงานวิจัยนี้กำหนดให้ค่า PARALLEL = 5 หากความแตกต่างของ M2-M1 มากกว่า 5 จึงจะรับรู้ว่าเส้นที่วาดเป็นเส้นตรงต่างเส้นกัน

```
if(abs(m2 - m1)<=PARALLEL){
        line[pointer].x1 = pointX1;
        line[pointer].y1 = pointY1;
        line[pointer].x2 = pointX2;
        line[pointer].y2 = pointY2;
}else if(abs(m2 - m1)>PARALLEL){
        pointer++;
}
```

ตาราง 3.1 ตัวอย่างโปรแกรมที่ใช้วิธีรับรู้เส้นตรง



รูปภาพ 3.19 ภาพแสดงการวาดเส้นตรงที่มีความชันใกล้เคียงกัน (ซ้าย) โปรแกรมจะรับรู้ว่าเป็นเส้นตรงเดียวกัน และการวาดเส้นตรงที่มีความชันระหว่างจุดต่างกัน (ขวา) มากเกินกว่าตัวแปรที่กำหนดโปรแกรมจะรับรู้ว่าไม่เป็นเส้นตรงเดียวกัน

- *การพัฒนาระบบรับรู้การตัดกันของเส้น*

ในการพัฒนาโปรแกรมจะใช้ตัวแปรในการรับรู้การตัดกันของเส้นโดยนำค่าพิกัดของเส้นมาหาค่าการตัดกับแกน Y-intercept จากสูตร

$$C = Y - MX$$

C คือค่า Y-Intercept หรือค่าการจัดแกน Y ของสมการเส้นตรง

แล้วจึงหาความชันของเส้นทั้งสองเส้นจากสูตรหาความชัน

$$M = \frac{Y2-Y1}{X2-X1}$$

M คือความชันของเส้นตรงที่มีจุดต้น (X1,Y1) และจุดปลาย (X2,Y2)

จากนั้นให้นำค่า C1 C2 M1 และ M2 ของเส้นทั้งสองเส้นมาแทนค่าในสูตรหาจุดตัด โดยที่ความชันของเส้นใดเส้นหนึ่งจะต้องไม่เท่ากับ 0 หรือไม่มีเส้นที่ขนานกับแกน X หรือ Y

```
X = (C2-C1)/(M1-M2);
Y = (M1*X)+C1;
```

C1,C2 คือค่า Y intercept หรือค่าการตัดแกน Y ของสมการเส้นตรง 2 เส้น

M1,M2 คือ ความชันของเส้นตรงสองเส้น

X คือ จุดตัดที่แกน X ของเส้นตรงทั้งสองเส้น

Y คือ จุดตัดที่แกน Y ของเส้นตรงทั้งสองเส้น

สำหรับเส้นที่ขนานกับแกนใดแกนหนึ่งสามารถนำค่า Y1 ไปแทนค่าในสมการ Y2=MX1+C2 เพื่อหาค่า X1 ได้ทันที โดยที่จะต้องทดสอบ X และ Y ว่าอยู่บนเส้นตรงที่ตัดกันหรือเปล่าเพื่อพิสูจน์ว่าเส้นตรงสองเส้นตัดกันจริงตามสูตร

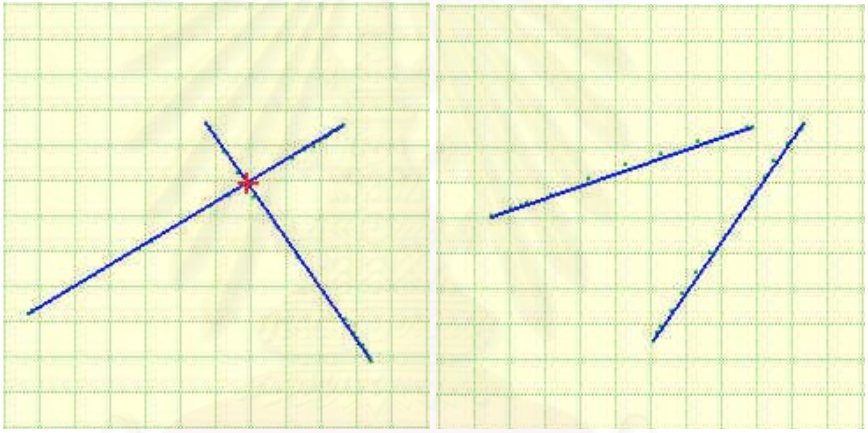
$$t = (X - X1)/(X2 - X1)$$

t คือตัวแปรที่ใช้วัดจุดพิกัด X ว่าอยู่บนส่วนของเส้นตรงหรือไม่

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
if(my1!==0&&mx2!==0){
        x = (c2-c1)/(m1-m2);
         y = (m1*x)+c1;
}
t = (x – line[pointer].x1)/(line[pointer].x2 - line[pointer].x1);
if(t>=0&&t<=1){
            intersect = true;
}
```

ตาราง 3.2 ตัวอย่างโปรแกรมรับรู้การตัดกันของเส้น



รูปภาพ 3.20 ภาพแสดงการทำงานของโปรแกรมที่สามารถรับรู้เส้นที่ตัดกันได้อย่างถูกต้อง (ซ้าย) และยังสามารถยกเลิกการจุดตัดในกรณีที่จุดตัดไม่ได้อยู่บนส่วนของเส้นตรงทั้งสองเส้น (ขวา)

- *การพัฒนาระบบรับรู้ระยะห่างระหว่างเส้น*

โปรแกรมจะทำงานโดยวัดระยะจุดปลายของเส้นสองเส้นของภาพวาดทั้งหมดหากเส้นคู่ใดมีจุดปลายที่มีระยะใกล้กันไม่เกินค่า EXTEND (กำหนดให้เป็น 40) โดยมีเงื่อนไขที่สัมพันธ์กับสูตร

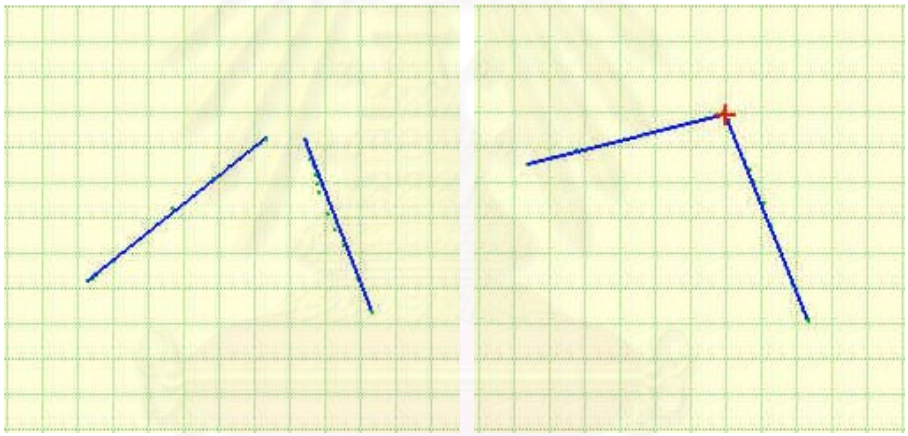
$$t = (X – X1)/(X2 – X1)$$

t คือตัวแปรที่ใช้วัดจุดพิกัด X ว่าอยู่บนส่วนของเส้นตรงหรือไม่

โดยกำหนดให้โปรแกรมทำงานเฉพาะกรณีที่ t มากกว่า 1 เพื่อไม่ให้ซ้ำกับระบบรับรู้ระยะยื่นของเส้น จากนั้นโปรแกรมจะทำการย้ายจุดปลายของเส้นทั้งสองเพื่อต่อเส้นให้มีจุดปลายชนกัน

```
dis = sqrt(pow(x1-x2,2)+pow(y1-y2,2));
if(dis<=EXTEND){
        line[pointer].x2 = x1;
        line[pointer].y2 = y1;
}
```

ตาราง 3.3 ตัวอย่างโปรแกรมรับรู้ระยะห่างของเส้น



รูปภาพ 3.21 ภาพแสดงการรับรู้ระยะห่างของจุดปลายของเส้นที่มีค่าเกินกว่าที่กำหนด (ซ้าย) และมีระยะห่างน้อยกว่าที่กำหนดไว้ (ขวา)

- *การพัฒนาระบบรับรู้ระยะยื่นของเส้น*

โปรแกรมจะมีวิธีการรับรู้และวิธีการทำงานเหมือนกับระบบรับรู้ระยะห่างของเส้น โดยกำหนดค่า TRIM ให้มีค่าเท่ากับ 40 และสัมพันธ์กับค่า t จากสูตร

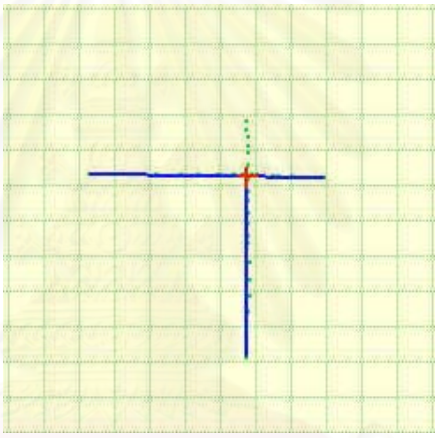
$$t = (X - X1)/(X2 - X1)$$

t คือตัวแปรที่ใช้วัดจุดพิกัด X ว่าอยู่บนส่วนของเส้นตรงหรือไม่

ค่า t จะต้องอยู่ระหว่าง 0 และ 1 เท่านั้น

```
dis = sqrt(pow(x1-x2,2)+pow(y1-y2,2));
if(dis<=TRIM){
        line[pointer].x2 = x1;
        line[pointer].y2 = y1;
}
```

ตาราง 3.4 ตัวอย่างโปรแกรมรับรู้ระยะยื่นของเส้น



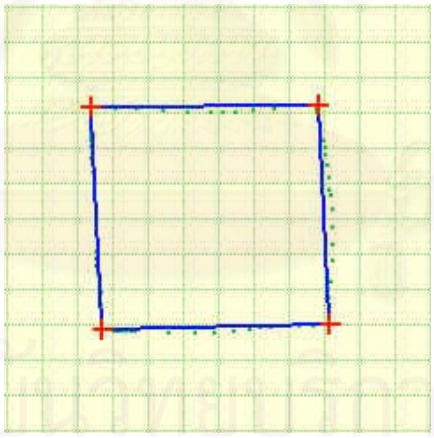
รูปภาพ 3.22 ภาพแสดงการทำงานของโปรแกรมโดยจะรับรู้ระยะยื่นเฉพาะเส้นที่ยื่นออกไปไม่เกินค่าที่กำหนด

- *การพัฒนาระบบรับรู้จุดมุมของภาพ*

ในการรับรู้จุดมุมของภาพโปรแกรมจะทำการจับเวลาที่ใช้ในการวาดจุดแต่ละจุด โดยเก็บความละเอียดในระดับ 1/1000 วินาที (millisecond) เมื่อได้ค่าตัวแปรระยะเวลาที่ใช้แล้ว ลำดับต่อไปจึงนำตัวแปรมาหาความแตกต่าง เทียบกับเวลาในการวาดจุดลำดับก่อนหน้าจุดในปัจจุบัน

```
timex = time;
realtime = currentTime();
time = realtime.second();
milsec = realtime.msec();
if(timex != time){
        lastpoint = firstpoint;
        firstpoint = milsec;
        lastpoint = abs(firstpoint-(999-lastpoint));
}else{
        lastpoint = firstpoint;
        firstpoint = milsec;
        lastpoint = abs(firstpoint-lastpoint);
}
```

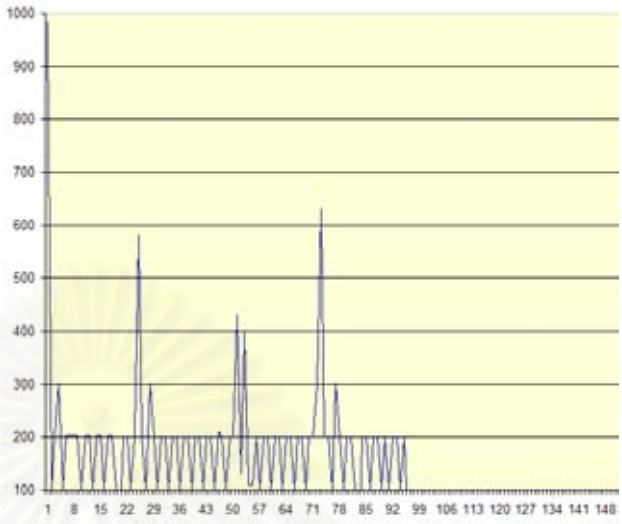
ตาราง 3.5 ตารางแสดงโปรแกรมหาความแตกต่างของเวลาในหน่วย 1/1000 วินาที



รูปภาพ 3.23 ภาพแสดงการทำงานของโปรแกรมรับรู้จุดมุมของภาพ

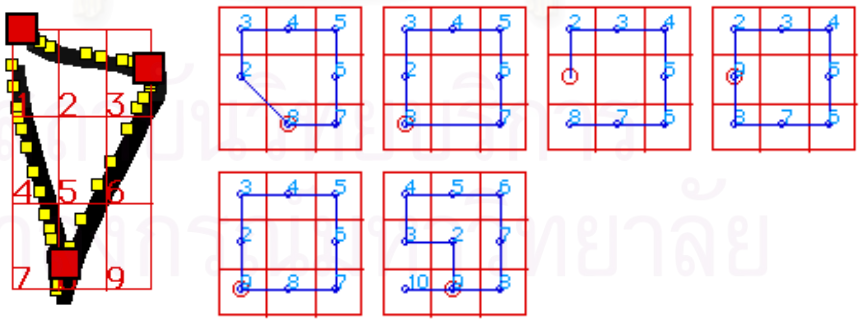
ในการพัฒนาโปรแกรมได้มีการพัฒนาระบบรายงานผลค่าความแตกต่างของเวลาในการวาดภาพลายเส้นเพื่อทดสอบความถูกต้องของโปรแกรมอีกด้วย โดยเก็บค่าเรียงตามลำดับจุดที่วาดภายในขอบเขตไม่เกิน 500 จุด แล้วบันทึกเป็นแฟ้มข้อมูลที่สามารถนำไปเปิดด้วยโปรแกรม MS Excel เพื่อวิเคราะห์ข้อมูลต่อไปได้

	A	B
1	Point No	Time Milliseconds
2	1	982
3	2	102
4	3	202
5	4	302
6	5	102
7	6	202
8	7	202
9	8	202
10	9	202
11	10	102
12	11	202
13	12	202
14	13	102
15	14	202
16	15	202
17	16	102
18	17	202
19	18	202
20	19	102
21	20	0
22	21	201
23	22	201
24	23	101



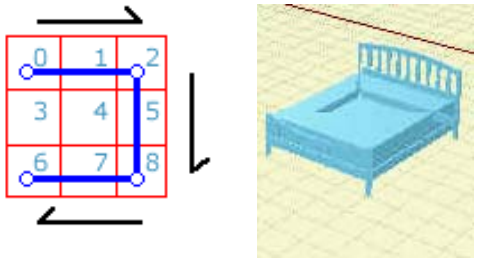
รูปภาพ 3.24 ภาพแสดงการเก็บค่าความแตกต่างของเวลาของจุดในการวาดภาพลายเส้น

- *การพัฒนาระบบรับรู้รูปแบบที่เหมือนกันของภาพวาดลายเส้น* (Sketches Pattern Recognition) จากงานวิจัยของ Fatos Bengi Durgun, Bulent Ozguc, 1990 และงานวิจัยของ Mark D. Gross, Ellen Yi-Luen Do, 1993 ทำให้สามารถกำหนดแนวทางในการพัฒนาระบบรับรู้รูปแบบที่เหมือนกันของภาพวาดลายเส้นได้ โดยกำหนดให้ใช้ขั้นตอนวิธีในการรับรู้ภาพวาดลายเส้นด้วยระบบตารางขนาด 3x3 โดยแต่ละช่องของตารางจะมีหมายเลขกำกับ หากการวาดเส้นมีรูปแบบตรงตามลำดับที่กำหนดไว้ โปรแกรมจะสามารถเลือกหุ่นจำลองจากฐานข้อมูลที่ตรงกันมาแสดง



(a) (b)

รูปภาพ 3.25 ภาพแสดงระบบตารางขนาด 3x3 (a) และวิธีการรับรู้ลำดับการวาดเส้น (b)



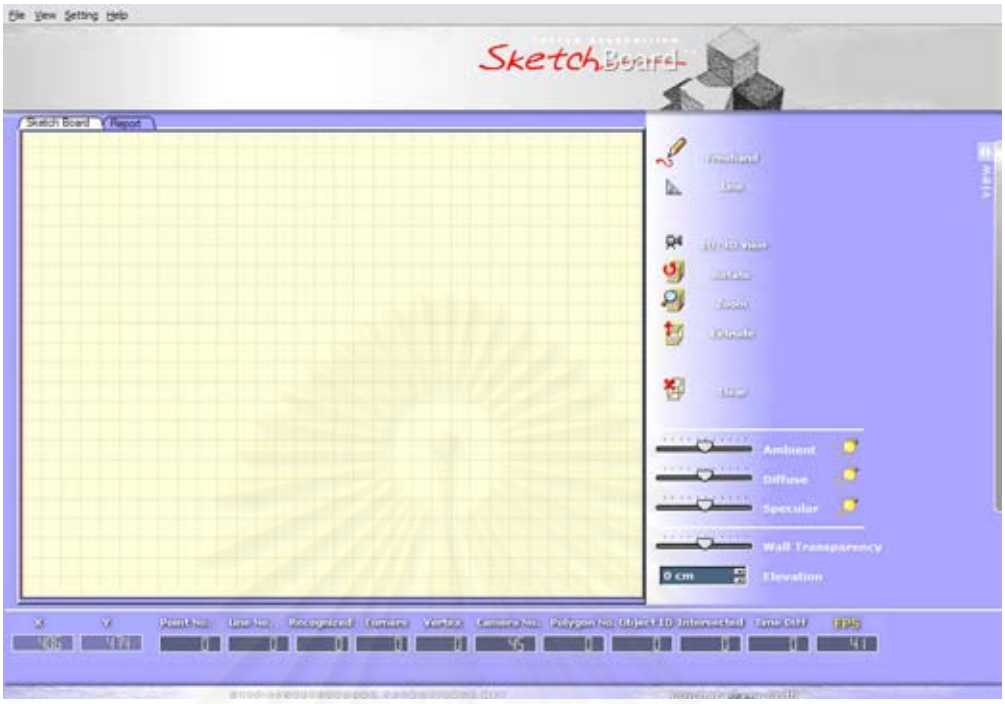
รูปภาพ 3.26 ภาพแสดงตัวอย่างการวาดเส้นตามลำดับ 0,2,8,6 (a)
และผลจากการรับรู้ตรงกันกับฐานข้อมูลเป็นหุ่นจำลอง Bed (b)

3.4.3 การพัฒนาส่วนโต้ตอบผู้ใช้

ในพัฒนาส่วนโต้ตอบจะต้องพัฒนาให้มีความเหมาะสมกับลักษณะการใช้งาน โดยจะต้องส่งเสริมการทำงานด้วยการวาดภาพลายเส้นเป็นหลัก ซึ่งมีการใช้ปากกาอิเล็กทรอนิกส์เป็นอุปกรณ์นำเข้าข้อมูล จึงออกแบบโดยอ้างอิงจากงานวิจัย Free Form User Interface (Takeo Igarashi, 1999) ซึ่งชี้ให้เห็นว่าการพัฒนาส่วนโต้ตอบผู้ใช้ของโปรแกรมทางด้านการวาดภาพลายเส้น จะต้องตอบสนองต่อการทำงานซึ่งเกี่ยวกับความคิดสร้างสรรค์ เป็นความคิดที่จะเกิดขึ้นอย่างรวดเร็ว จึงไม่ต้องการความซับซ้อนของขั้นตอนการทำงาน เช่น การทำงานที่ต้องกดปุ่มจำนวนมาก หรือต้องพิมพ์คำสั่งหลายขั้นตอน

การออกแบบส่วนโต้ตอบผู้ใช้จึงออกแบบให้มีความเรียบง่าย สร้างปุ่มในการทำงานให้น้อยที่สุดและเน้นพื้นที่การวาดภาพลายเส้นเป็นหลัก และกำหนดให้รับข้อมูลจากปากกาอิเล็กทรอนิกส์ซึ่งเน้นการลาก (Dragging) และการวาด (Stroking) มากกว่าการกดปุ่มขึ้นลง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปภาพ 3.27 ภาพแสดงการออกแบบส่วนโต้ตอบผู้ใช้โดยออกแบบให้มีปุ่มน้อยที่สุดและเน้นพื้นที่การวาดภาพ

รูปภาพ 3.28 ภาพแสดงส่วนแสดงรายงานข้อมูลที่สำคัญจากการวาดภาพลายเส้น
โดยสามารถบันทึกเป็นแฟ้มเพื่อนำไปวิเคราะห์ต่อด้วยโปรแกรม MS Excel ได้

บทที่ 4

ผลการออกแบบโปรแกรม

จากการศึกษาขั้นตอนวิธีในการรับรู้ภาพวาดลายเส้น ทำให้สามารถออกแบบและพัฒนาโปรแกรมที่สามารถทำงานโดยอาศัยขั้นตอนวิธีในการรับรู้ภาพวาดลายเส้นเพื่อประยุกต์ใช้ในการสร้างรูปทรง 3 มิติเบื้องต้นได้ โดยสามารถแบ่งขั้นตอนทั้งหมดในการใช้งานโปรแกรมได้ดังนี้

4.1 เตรียมอุปกรณ์

4.2 การติดตั้งโปรแกรม

4.3 ปุ่มและคำสั่งในการใช้งาน

4.4 การใช้งานโปรแกรมโดยทั่วไป

4.5 การปรับค่าในการใช้งาน

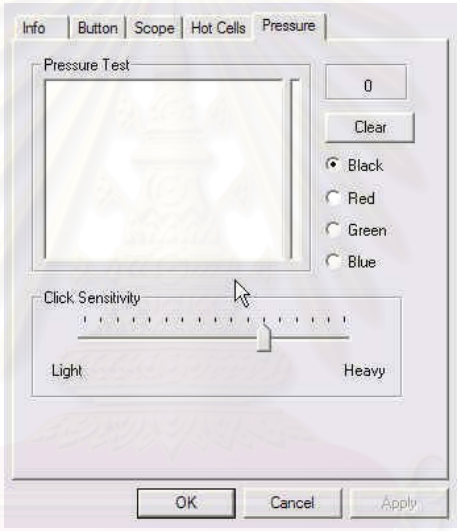
4.7 การจัดทำรายงาน

4.1 เตรียมอุปกรณ์ ในการทำงานกับโปรแกรมรับรู้ภาพวาดลายเส้น จำเป็นต้องใช้อุปกรณ์ที่สำคัญ คือ ปากกาอิเล็กทรอนิกส์ หรืออุปกรณ์อื่นๆ ที่สามารถรับข้อมูลด้วยปากกาอิเล็กทรอนิกส์ เช่น Tablet PC โดยอุปกรณ์ที่ใช้จะต้องมีการปรับตั้งดังต่อไปนี้

4.1.1 การปรับแรงกด ในการวาดภาพด้วยปากกาอิเล็กทรอนิกส์จำเป็นต้องมีการปรับตั้งแรงกดในการวาดภาพให้เหมาะสม เนื่องจากแรงกดจะเป็นตัวแปรที่มีความสัมพันธ์กับความถี่ในการสร้างจุดของโปรแกรม หากโปรแกรมมีการสร้างจุดมากเกินไปจะทำให้ไม่สามารถนำจุดที่ได้มาทำการคำนวณได้อย่างถูกต้อง จึงต้องมีการปรับแรงกดของปากกาให้เหมาะสม โดยในงานวิจัยนี้เลือกใช้อุปกรณ์ ปากกาอิเล็กทรอนิกส์จากบริษัท Genius[14] รุ่น WizardPen 4x3 โดยปรับแรงกดให้มีระดับ 70% และมีค่าความเร็วในการสร้างกรอบภาพต่อวินาที (Frame Per Second) ประมาณ 60 FPS

[14] Genius http://www.genius.com.tw

รูปภาพ 4.1 อุปกรณ์รับข้อมูลด้วยปากกาอิเล็กทรอนิกส์ (แหล่งที่มา: http://www.wacom.com)



รูปภาพ 4.2 การตั้งค่าแรงกดของปากกาเท่ากับ 70% โดยมีความเร็วในการสร้างกรอบภาพ 60 FPS

4.1.2 **ระบบปฏิบัติการที่ใช้** ระบบปฏิบัติการที่แนะนำให้ใช้ในการติดตั้งโปรแกรมเป็นระบบปฏิบัติการ Microsoft Windows XP Professional[15] หรือระบบปฏิบัติการอื่น ของ Microsoft Windows ที่สามารถติดต่อกับ คลังกราฟิก (Graphic Library) OpenGL[16] แบบ 32 bit

4.1.3 **อุปกรณ์เร่งความเร็วกราฟิก** โปรแกรมนี้จำเป็นต้องใช้อุปกรณ์แสดงภาพกราฟิก หรือการ์ดเร่งความเร็วกราฟิก ที่รองรับ คลังกราฟิก (Graphic Library) OpenGL เวอร์ชั่น 1.2 โดย

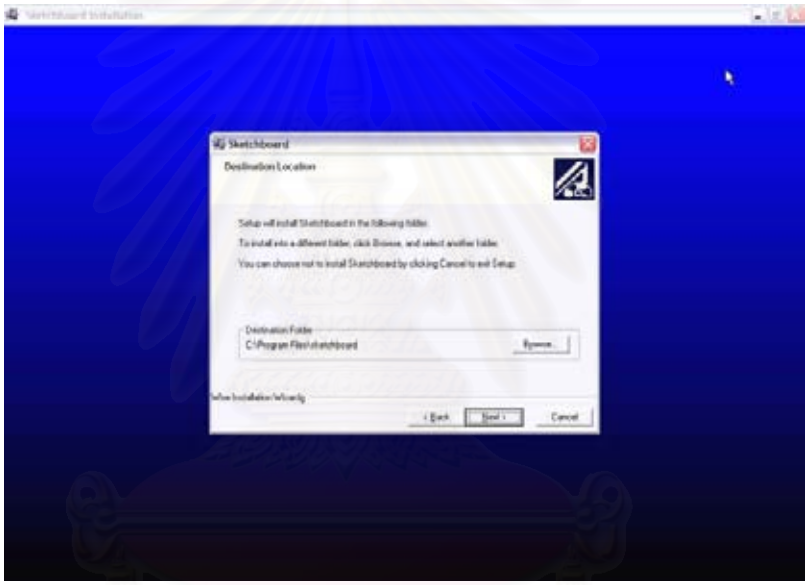
[15] Microsoft Corporation

[16] Silicon Graphic Inc. http://www.opengl.org

งานวิจัยนี้ได้ทดสอบและเลือกใช้การ์ดแสดงผลของบริษัท ATI รุ่น Radeon 9000 Pro DDR 64 MB[17] และบริษัท Nvidia รุ่น GForce 2 Go 100 DDR 32 MB [18]

4.2 การติดตั้งโปรแกรม ในการติดตั้งโปรแกรมสามารถติดตั้งโดยการเลือกแฟ้มในการติดตั้งชื่อแฟ้ม sketchboard_x.x.xx.exe เพื่อสั่งให้โปรแกรมทำการติดตั้งเองโดยอัตโนมัติ หลังจากโปรแกรมเริ่มทำการติดตั้งจะมีหน้าจอแสดงการติดตั้ง ให้ตอบตกลงเพื่อให้โปรแกรมติดตั้งตามลำดับขั้นตอนต่อไป

รูปภาพ 4.3 ภาพแสดงแฟ้มสำหรับติดตั้งโปรแกรม โดยมีตัวเลขบอกเวอร์ชันของโปรแกรมต่อท้าย



รูปภาพ 4.4 ภาพแสดงหน้าจอการติดตั้งโปรแกรม สามารถเลือกแฟ้มปลายทางที่จะทำการติดตั้งได้

หลังจากติดตั้งแล้วโปรแกรมจะสร้าง icon สำหรับเปิดโปรแกรมชื่อ SketchBoard ให้เลือกเพื่อเปิดโปรแกรมสำหรับทำงานในขั้นต่อไป

รูปภาพ 4.5 ภาพแสดง icon สำหรับเปิดโปรแกรม

[17] ATI Corporation http://www.ati.com

[18] NVIDIA http://www.nvidia.com

4.3 ปุ่มและคำสั่งในการทำงาน หลังจากเริ่มเปิดโปรแกรมเพื่อทำงานแล้ว โปรแกรมจะแสดงหน้าจอขึ้นมาโดยประกอบด้วยองค์ประกอบดังต่อไปนี้

4.3.1 ส่วนเมนูหลัก

4.3.2 ส่วนปุ่มในการทำงาน

4.3.3 ส่วนพื้นที่วาดภาพ

4.3.4 ส่วนแสดงข้อมูลประกอบการทำงาน

4.3.5 ส่วนรายงานผล

รูปภาพ 4.6 ภาพแสดงหน้าจอการทำงานของโปรแกรม

4.3.1 **ส่วนเมนูหลัก** ส่วนเมนูหลักประกอบด้วยเมนูดังต่อไปนี้

- File สำหรับการจัดการแฟ้มข้อมูล เปิด บันทึก และออกจากโปรแกรม
- View สำหรับการปรับมุมมองให้แสดงภาพแบบปรับแต่งขอบภาพ AntiAlias ให้มีความนุ่มนวลมากขึ้น หรือปิดการทำงานการปรับแต่งขอบภาพ
- Setting สำหรับปิดเปิดระบบการรับรู้ภาพวาดลายเส้น ปิดเปิดการ snap และปิดเปิด grid
- Help สำหรับแสดงข้อมูลผู้ออกแบบและพัฒนาโปรแกรม

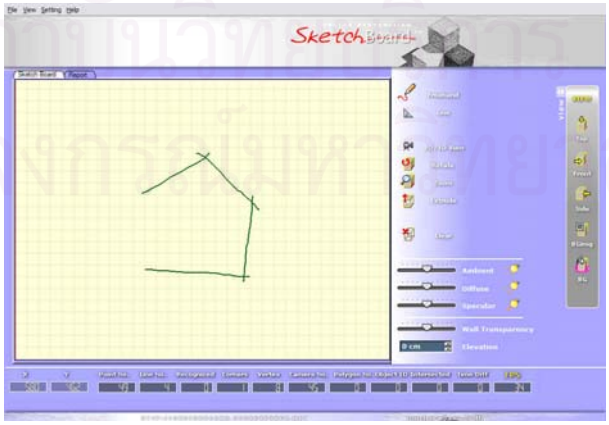
รูปภาพ 4.7 ภาพแสดงเมนูหลักของโปรแกรม

4.3.2 **ส่วนปุ่มในการทำงาน** มีการสร้างปุ่มเพื่อรองรับการทำงานหลักของโปรแกรมโดยประกอบด้วยปุ่มดังต่อไปนี้

- Freehand สำหรับทำงานด้วยการวาดภาพลายเส้นแบบอิสระโดยจะมีการวิเคราะห์ด้วยระบบรับรู้ลายเส้นโดยอัตโนมัติ
- Line สำหรับทำงานด้วยการวาดภาพลายเส้นแบบเรขาคณิต โดยโปรแกรมจะวาดเส้นเพื่อสร้างภาพเรขาคณิตและภาพหลายเหลี่ยม (Polygon) อีกทั้งยังสามารถเลือกขนาดของหัวปากกาได้ว่าจะให้มีการวาดเส้นขนาด 0.1 หรือ 0.5 มม.
- 2D/3D View คือปุ่มสำหรับเปลี่ยนมุมมองจาก พื้นที่การทำงานแบบ 2 มิติ สลับกับแบบ 3 มิติ
- Zoom เป็นปุ่มที่ใช้ดึงภาพเข้าและออก สำหรับการทำงานแบบ 3 มิติเท่านั้น
- Clear คือปุ่มล้างการทำงานของโปรแกรมทั้งหมดในพื้นที่การวาดภาพ
- Ambient Light คือปุ่มปรับระดับแสงแบบ Ambient ในพื้นที่การทำงานแบบ 3 มิติ เป็นแสงที่มาจากทุกทิศทางไม่ว่าแหล่งกำเนิดแสงจะอยู่ตำแหน่งใดก็ตาม
- Diffuse Light คือปุ่มปรับระดับแสงแบบ Diffuse ในพื้นที่การทำงานแบบ 3 มิติ เป็นแสงที่มาจากทิศทางของแหล่งกำเนิดและเมื่อกระทบกับพื้นผิวใดๆ แล้วจะสะท้อนออกไปทุกทิศทาง
- Specular Light คือปุ่มปรับระดับแสงแบบ Specular Light ในพื้นที่การทำงานแบบ 3 มิติ เป็นแสงที่มีทิศทางจากแหล่งกำเนิดและเมื่อกระทบกับพื้นผิวใดๆ แล้วจะสะท้อนออกไปในทิศทางเดียว
- Elevation คือปุ่มปรับระดับในแกน z ของพื้นที่การวาดภาพเพื่อให้สามารถวาดภาพลายเส้นในระดับ 3 มิติได้

รูปภาพ 4.8 ภาพแสดงปุ่มหลักในการใช้โปรแกรม

4.3.3 **ส่วนพื้นที่วาดภาพ** เป็นส่วนสำคัญที่ผู้ใช้งานจะใช้ปากกาอิเล็กทรอนิกส์ในการวาดภาพลายเส้น และจะเป็นส่วนรับข้อมูลของโปรแกรมที่จะนำไปสู่การรับรู้ภาพวาดลายเส้นต่อไป และพื้นที่นี้ยังเป็นส่วนแสดงผลในรูปแบบ 3 มิติ เพื่อให้ผู้ใช้สามารถปรับแต่งผลงานให้มีขนาดและระยะที่ต้องการได้ ในการออกแบบโปรแกรมจะเน้นการทำงานบนพื้นที่นี้เป็นหลัก เพื่อให้ผู้ใช้สามารถทำงานวาดภาพได้อย่างต่อเนื่องโดยไม่ต้องสลับไปเลือกคำสั่งอื่นๆ จำนวนมากในการวาดภาพลายเส้น



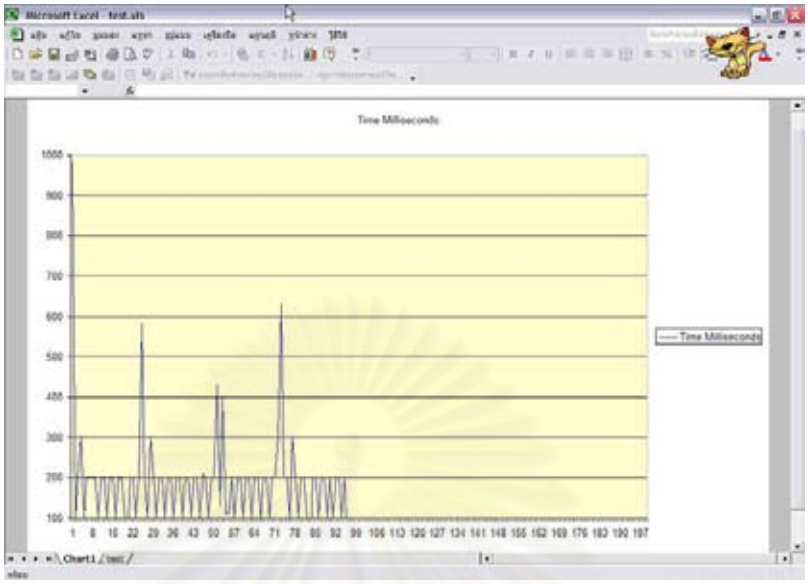
รูปภาพ 4.9 ภาพแสดงการวาดภาพลายเส้นบนพื้นที่สำหรับวาดภาพ

4.3.4 **ส่วนแสดงข้อมูลประกอบการทำงาน** ในการวาดภาพลายเส้นที่มีการรับรู้ภาพวาดลายเส้น จำเป็นจะต้องมีส่วนแสดงข้อมูลประกอบการทำงานเพื่อแสดงข้อมูลที่โปรแกรมรับจากผู้ใช้และข้อมูลที่โปรแกรมรับรู้จากภาพวาดลายเส้น เพื่อให้ผู้ใช้สามารถตรวจสอบความถูกต้องในการทำงานของโปรแกรมได้ตลอดเวลา

รูปภาพ 4.10 ภาพแสดงข้อมูลประกอบการทำงาน

4.3.5 **ส่วนรายงานผล** คือส่วนแสดงผลการทำงานของโปรแกรมเพื่อนำข้อมูลที่ได้จากการวาดภาพลายเส้นไปวิเคราะห์ต่อหรือนำไปเป็นข้อมูลประกอบการประเมินผลการทำงานของโปรแกรม โดยข้อมูลที่แสดงผลประกอบด้วยข้อมูลจำนวนจุดที่สร้างขึ้น และข้อมูลความแตกต่างของเวลาในการวาดจุด มีหน่วยเป็น 1/1000 วินาที ซึ่งข้อมูลที่ได้สามารถบันทึกเป็นแฟ้มข้อมูล แล้วนำไปเปิดโดยโปรแกรม Microsoft Excel ได้ เพื่อวิเคราะห์ด้วยกราฟหรือวิเคราะห์ด้วยวิธีการอื่นๆ ได้โดยง่าย

รูปภาพ 4.11 ภาพแสดงส่วนรายงานผล



รูปภาพ 4.12 ภาพแสดงการนำรายงานไปวิเคราะห์ด้วยโปรแกรม Microsoft Excel

4.4 **การใช้งานโปรแกรมโดยทั่วไป** ในการใช้งานโปรแกรมสามารถลำดับขั้นตอนได้ดังนี้

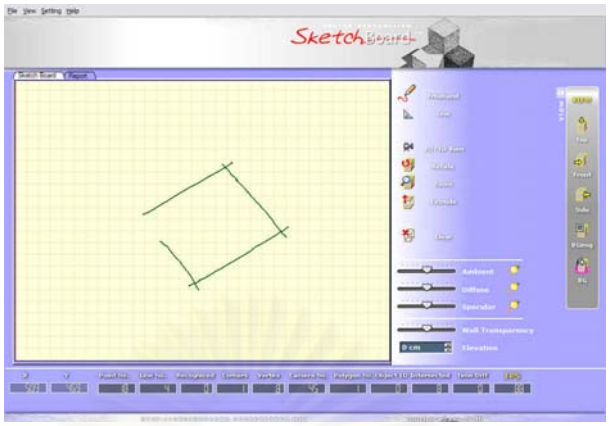
4.4.1 **การวาดภาพ** การใช้งานสามารถเริ่มทำงานด้วยการวาดภาพลายเส้นด้วยปากกาอิเล็กทรอนิกส์ โดยไม่ต้องกดปุ่มใดๆ ซึ่งสามารถวาดภาพได้สองลักษณะคือ

- การวาดเส้นแบบอิสระ
- การวาดเส้นแบบเรขาคณิต

- *การวาดเส้นแบบอิสระ* คือการวาดภาพลายเส้นที่ไม่มีการควบคุมวิธีการวาดเส้นโดยผู้ใช้สามารถลากเส้นได้อย่างอิสระเพื่อสร้างภาพงานออกแบบเขียนแบบได้ทันที ในงานวิจัยนี้ได้พัฒนาโปรแกรมให้สามารถรับรู้ลักษณะการวาดเส้นได้ 2 ลักษณะที่เหมาะกับงานวาดเส้นทางสถาปัตยกรรม คือ

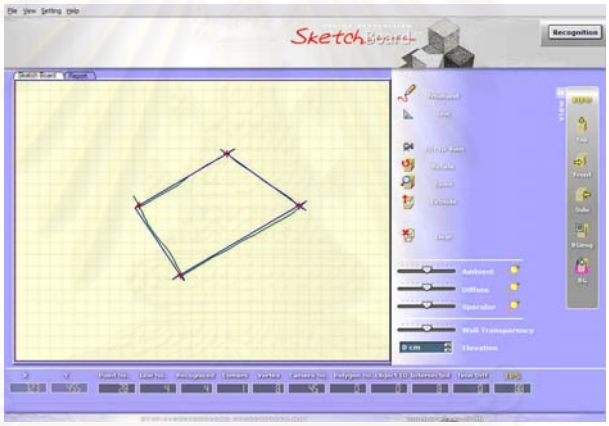
- การวาดเส้นอิสระแบบตัดมุม
- การวาดเส้นอิสระแบบต่อเนื่อง

การวาดเส้นอิสระแบบตัดมุม คือลักษณะการวาดภาพลายเส้นแบบอิสระที่ผู้วาดมักจะลากเส้นมาตัดกันที่ขอบของภาพเพื่อสร้างขอบเขตของภาพวาดลายเส้น

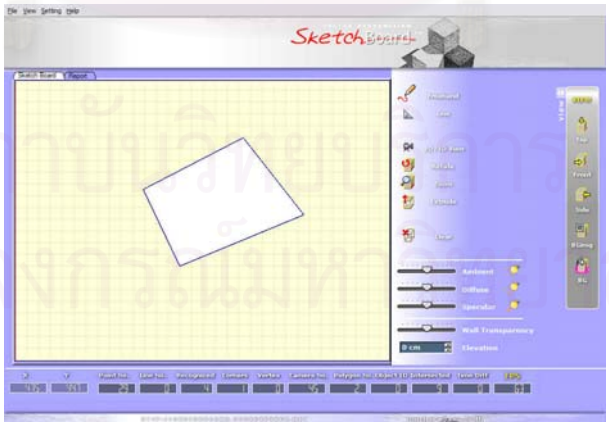


รูปภาพ 4.13 ภาพแสดงการวาดภาพแบบเรขาคณิต

การวาดเส้นอิสระแบบต่อเนื่อง คือการวาดภาพลายเส้นต่อเนื่องโดยไม่มีการยกปากกา สามารถรับรู้ได้ด้วยวิธีการรับรู้ความแตกต่างของเวลา (Time Difference)



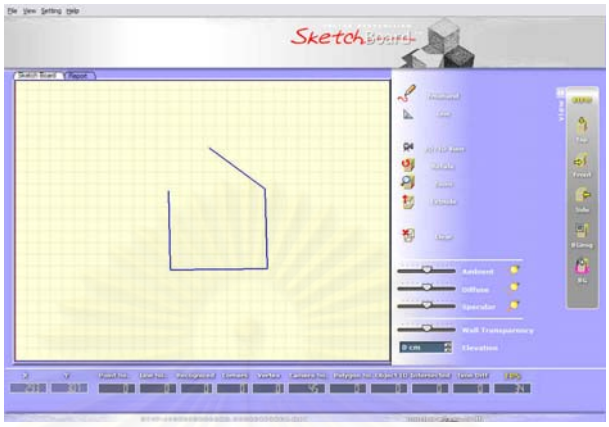
(a)



(b)

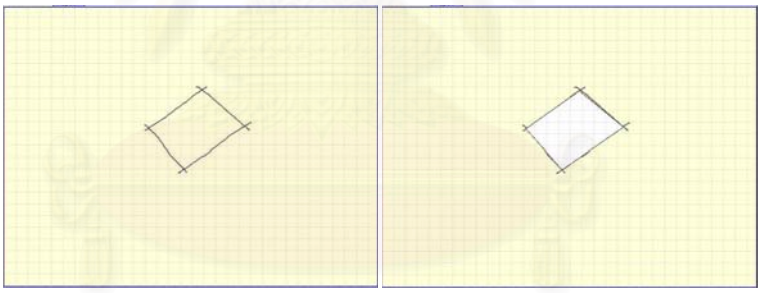
รูปภาพ 4.14 ภาพแสดงการวาดภาพลายเส้นอิสระแบบต่อเนื่อง (a) และผลจากการรับรู้เส้น (b)

- *การวาดเส้นแบบเรขาคณิต* คือการวาดเส้นที่มีการควบคุมวิธีการวาดเส้นและลักษณะของเส้น โดยงานวิจัยนี้กำหนดให้เป็นการวาดเส้นตรง

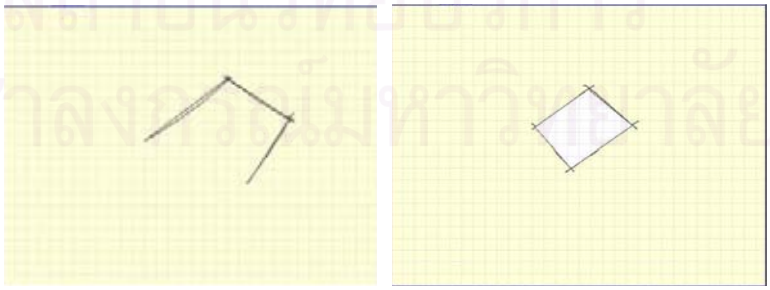


รูปภาพ 4.15 ภาพแสดงการวาดเส้นแบบเรขาคณิต

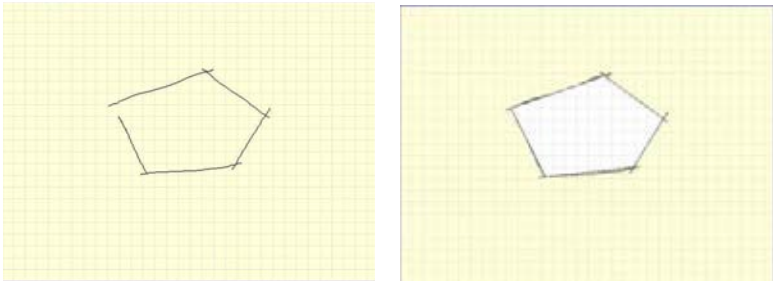
4.4.2 **การรับรูปภาพวาดลายเส้น** หลังจากวาดเส้นจนเสร็จแล้วโปรแกรมจะนำข้อมูลจากภาพวาดมารับรู้ด้วยขั้นตอนวิธีทางคอมพิวเตอร์เพื่อแปลงภาพวาดลายเส้นให้เป็นเส้นเรขาคณิตทางคอมพิวเตอร์



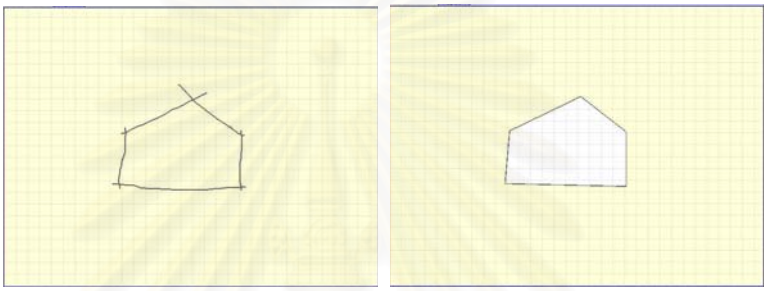
รูปภาพ 4.16 การรับรู้ภาพวาดลายเส้นอิสระแบบตัดมุม



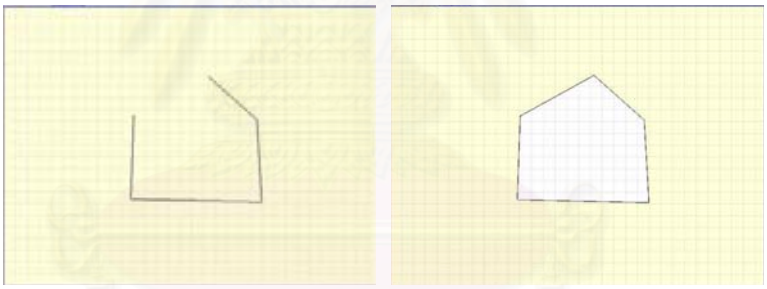
รูปภาพ 4.17 การรับรู้ภาพวาดลายเส้นอิสระแบบเส้นต่อเนื่อง



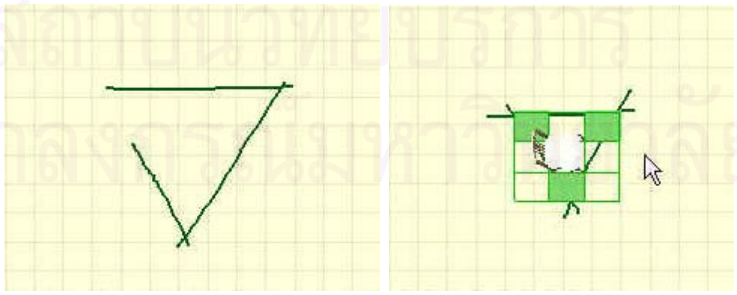
รูปภาพ 4.18 การรับรู้ภาพวาดลายเส้นอิสระโดยรับรู้ระยะเว้นว่าง



รูปภาพ 4.19 การรับรู้ภาพวาดลายเส้นอิสระโดยรับรู้ระยะยื่น



รูปภาพ 4.20 รับรู้ภาพวาดลายเส้นแบบเรขาคณิต

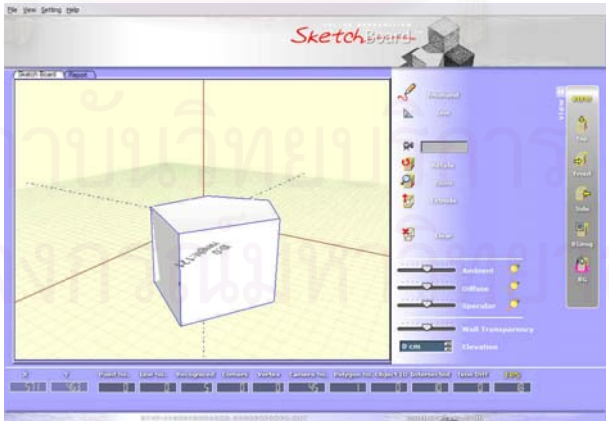


รูปภาพ 4.21 รับรู้รูปแบบที่เหมือนกันของภาพวาดลายเส้น (Sketches Pattern Recognition)

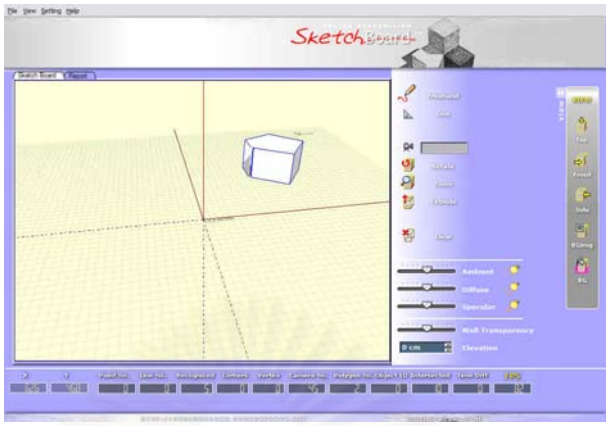
4.4.3 การเปลี่ยนมุมมองและการสร้างภาพ 3 มิติ หลังจากสร้างภาพเรขาคณิตจากภาพวาดลายเส้นแล้วผู้ใช้สามารถปรับเปลี่ยนมุมมองในการทำงานจาก 2 มิติ ให้เป็น 3 มิติได้ เพื่อให้สามารถสร้างภาพ 3 มิติจากภาพวาดลายเส้น เพื่อช่วยในการออกแบบเขียนแบบได้

รูปภาพ 4.22 ภาพแสดงการเปลี่ยนมุมมอง 3 มิติ

ในการสร้างภาพ 3 มิติ ให้กดปุ่ม Extrude แล้วจึงเลือกภาพเรขาคณิตหรือ Polygon ที่ต้องการเพื่อสร้างความสูงใน แกน z ให้กับวัตถุได้

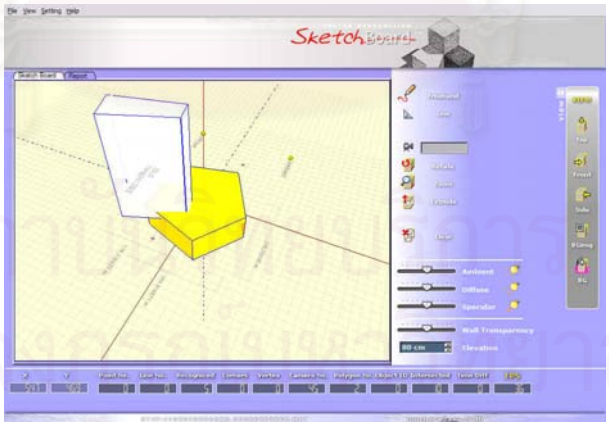


รูปภาพ 4.23 ภาพแสดงการสร้างความสูงในแกน z ให้กับวัตถุ

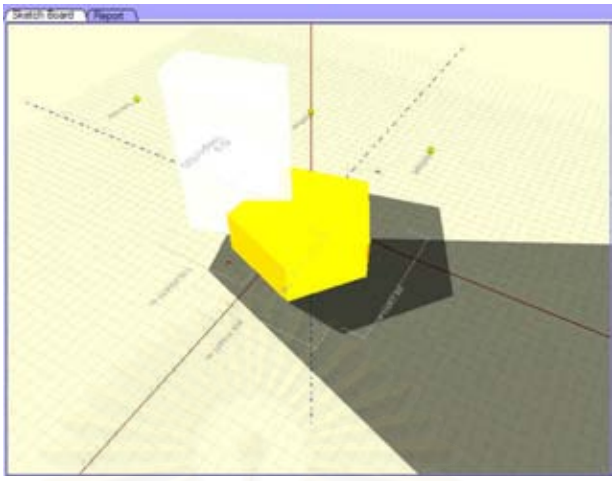


รูปภาพ 4.24 ภาพแสดงการปรับมุมมองภาพโดยการลากปากกาในพื้นที่ทำงานเพื่อหมุนกล้อง

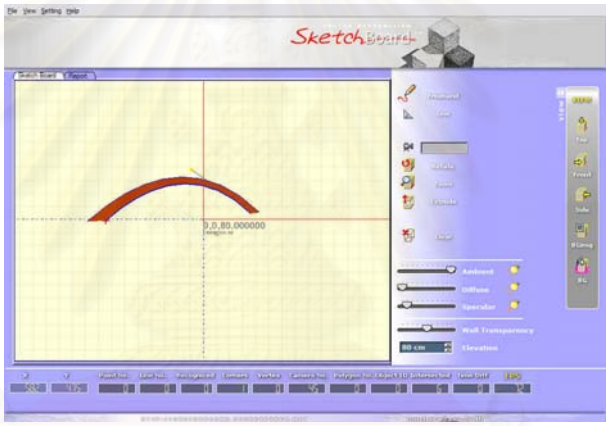
รูปภาพ 4.25 ภาพแสดงการบอกระยะของวัตถุที่สร้างขึ้น



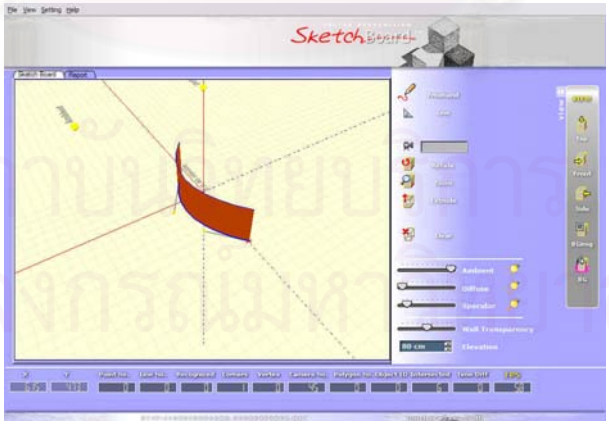
รูปภาพ 4.26 ภาพแสดงการสร้างวัตถุที่ระดับความสูงในแกน z ที่ผู้ใช้เป็นผู้กำหนด



รูปภาพ 4.27 ภาพแสดงการใส่เงาให้กับวัตถุทั้งหมดในภาพเพื่อสร้างความสมจริงให้กับวัตถุ

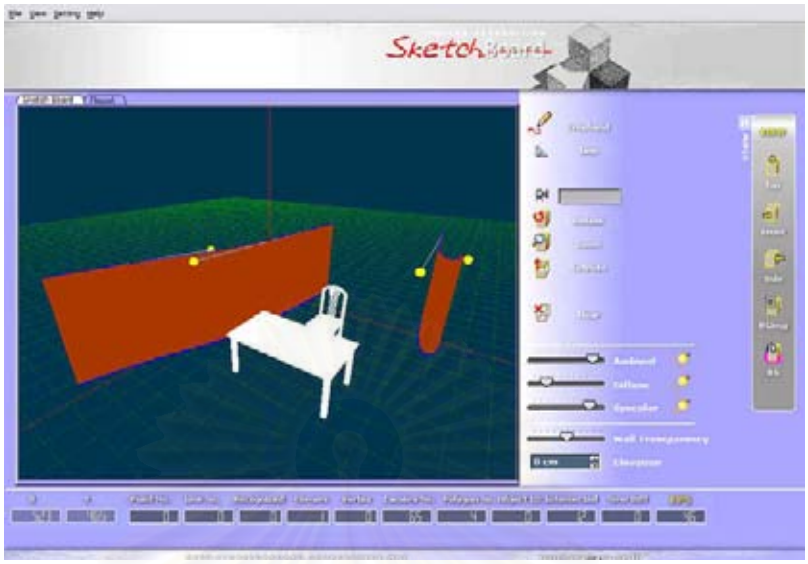


(a)



(b)

รูปภาพ 4.28 ภาพแสดงการสร้างเส้นโค้ง (a) และแผ่นระนาบโค้ง 3 มิติ (b)

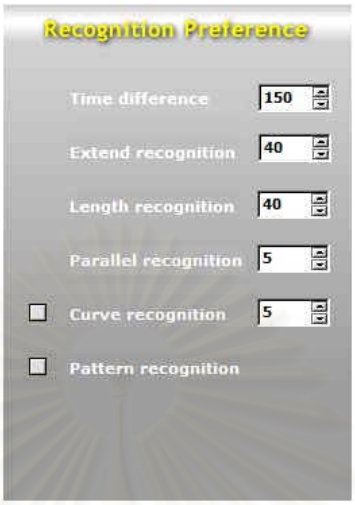


รูปภาพ 4.29 ภาพแสดงการจัดหุ่นจำลองชุดเก้าอี้กับผนังโค้งจากการรับรู้ภาพวาดลายเส้น

4.5 **การปรับค่าในการใช้งาน** ในการใช้งานโปรแกรมจำเป็นจะต้องใช้กำหนดตัวแปรหลายประเภท ดังที่ได้วิเคราะห์ไว้ในบทที่ 3 ซึ่งตัวแปรบางอย่างมีความเกี่ยวข้องกับความถนัดในการทำงานของแต่ละบุคคลด้วยโดยเฉพาะการทำงานของสถาปนิกที่มีลักษณะความถนัดเฉพาะบุคคลสูง จึงมีการทำส่วนปรับแต่งข้อมูลสำหรับการทำงานของโปรแกรมขึ้นเพื่อให้ผู้ใช้แต่ละคนสามารถปรับแต่งตัวแปรที่เหมาะสมกับตัวเองได้ โดยประกอบด้วยตัวแปรที่ปรับได้ดังต่อไปนี้

- Time Difference คือตัวแปรในการรับรู้ความแตกต่างของเวลาระหว่างจุด ค่าเริ่มต้นเท่ากับ 150
- Extend Recognition คือตัวแปรในการรับรู้ระยะเว้นว่างของเส้น ค่าเริ่มต้นเท่ากับ 40
- Length Recognition คือตัวแปรในการรับรู้ระยะยื่นของเส้น ค่าเริ่มต้นเท่ากับ 40
- Parallel Recognition คือตัวแปรในการรับรู้ความขนานกันของเส้น ค่าเริ่มต้นเท่ากับ 5
- Curve Recognition คือการปิดเปิดระบบรับรู้เส้นโค้ง
- Pattern Recognition คือการปิดเปิดระบบรับรู้รูปแบบที่เหมือนกันของภาพวาดลายเส้น

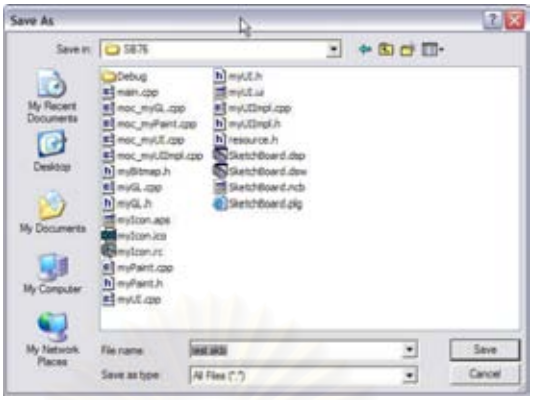
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



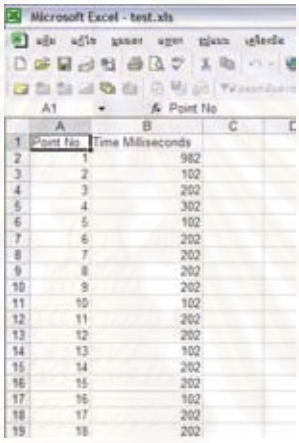
รูปภาพ 4.30 ภาพแสดงส่วนปรับแต่งการทำงานของโปรแกรม

4.6 **การจัดทำรายงาน** หลังจากการวาดภาพลายเส้นแล้วจะต้องมีการประเมินความถูกต้องของโปรแกรม จึงพัฒนาระบบรายงานเพื่อนำข้อมูลที่ได้ไปวิเคราะห์ต่อด้วยโปรแกรม Microsoft Excel เพื่อวิเคราะห์และจัดทำเป็นรายงานการประเมินผลโปรแกรมต่อไป

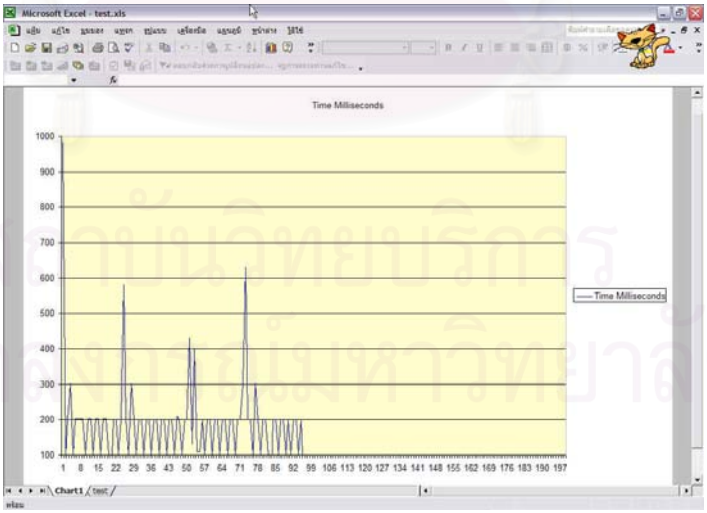
รูปภาพ 4.31 ภาพแสดงระบบรายงานผลการทำงานของโปรแกรม



รูปภาพ 4.32 ภาพแสดงการบันทึกข้อมูลจากรายงานเป็นแฟ้มข้อมูล *.skb



รูปภาพ 4.33 ภาพแสดงการเปิดข้อมูลที่บันทึกไว้ด้วยโปรแกรม Microsoft Excel



รูปภาพ 4.34 ภาพแสดงการวิเคราะห์ข้อมูลวาดภาพลายเส้นด้วยกราฟจากโปรแกรม Microsoft Excel จะเห็นการใช้เวลาสูงกว่าปกติ 4 จุด ซึ่งเป็นจุดมุมของการวาดภาพสี่เหลี่ยม

บทที่ 5

การประเมินผลการวิจัย

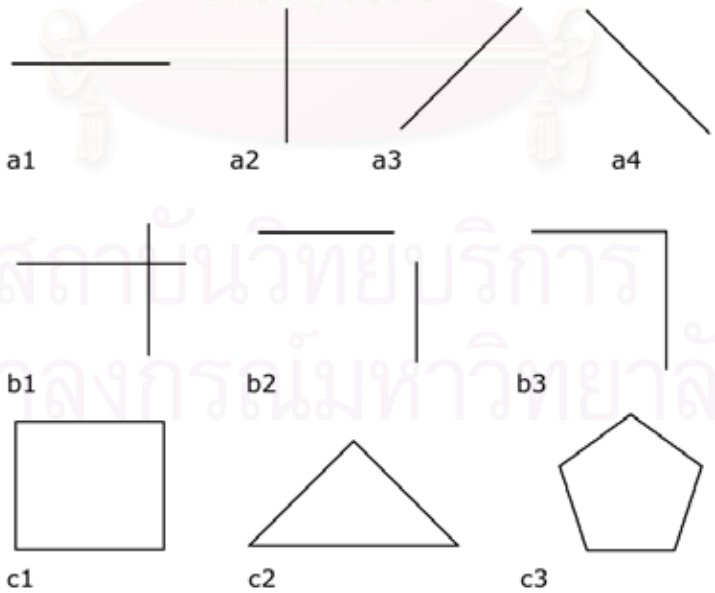
จากการพัฒนาโปรแกรมรับรู้ภาพวาดลายเส้นโดยใช้ขั้นตอนวิธีดังที่ได้กล่าวมาแล้ว จะต้องมีการทดสอบขั้นตอนวิธีรับรู้แต่ละแบบเพื่อหาความถูกต้องในการรับรู้ภาพวาดลายเส้น โดยการวาดภาพลายเส้นตามตัวอย่างที่กำหนดให้ด้วยปากกาอิเล็กทรอนิกส์ วาดตามตัวอย่างละ 10 ครั้งจากทั้งหมด 10 ตัวอย่าง โดยที่ตัวอย่างที่ใช้จะเป็นเส้นเรขาคณิตเบื้องต้น เช่น เส้นตั้ง เส้นนอน เส้นเอียง รูปหลายเหลี่ยม เพราะเป็นเส้นที่ใช้มากที่สุดในงานสถาปัตยกรรม (Frank Ching, 1985) จากการทดสอบสามารถประเมินและสรุปผลได้ตามหัวข้อดังนี้

5.1 การรับรู้ภาพวาดลายเส้น

5.2 การสร้างรูปทรงสามมิติเบื้องต้น

5.1 การรับรู้ภาพวาดลายเส้น

จากการประเมินเพื่อวัดความถูกต้องในการรับรู้ภาพวาดลายเส้น โดยกำหนดรูปแบบในการวาดเส้นด้วยรูปทรงเรขาคณิตเบื้องต้น เพื่อเป็นต้นแบบในการวาดทั้งหมด 10 แบบดังต่อไปนี้



รูปภาพ 5.1 ต้นแบบที่ใช้ในการทดสอบการรับรู้วาดภาพลายเส้น

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

จากการทดสอบโดยการวาดภาพลายเส้นตามต้นแบบใน รูปภาพ 5.1 แบบละ 10 ครั้ง สามารถการประเมินผลความถูกต้องในการรับรู้ได้ดังต่อไปนี้

รูปแบบภาพวาด	รับรู้ถูกต้อง	รับรู้ผิด	เปอร์เซ็นต์ความถูกต้อง
a1	10	0	100%
a2	10	0	100%
a3	10	0	100%
a4	10	0	100%
b1	7	3	70%
b2	9	1	90%
b3	8	2	80%

ตาราง 5.1 แสดงผลการทดสอบการรับรู้ภาพวาดลายเส้น a1-b3

นอกจากนี้ยังมีการทดสอบการรับรูปภาพวาดลายเส้น c1-c3 ตามรูปภาพ 5.1 ได้ผลการทดสอบดังต่อไปนี้

รูปแบบภาพวาด	ตัวแปร Time Difference	รับรู้ถูกต้อง	รับรู้ผิด	เปอร์เซ็นต์ความถูกต้อง
c1	100	5	5	50%
c1	150	3	7	30%
c1	200	1	9	10%
c2	100	3	7	30%
c2	150	2	8	20%
c2	200	1	9	10%
c3	100	2	8	20%
c3	150	7	3	70%
c3	200	0	10	0%

ตาราง 5.2 แสดงผลการทดสอบการรับรู้ภาพวาดลายเส้น c1-c3 โดยมีการปรับค่าตัวแปร Time Difference เพื่อทดสอบความเหมาะสมของตัวแปร

จากการทดสอบด้วยการวาดภาพลายเส้น สรุปได้ว่าโปรแกรมสามารถรับรู้ภาพวาดลายเส้นได้อย่างถูกต้องโดยสามารถรับรู้ลักษณะของภาพวาดลายเส้น 5 ลักษณะคือ

5.1.1 เส้นต่อเนื่องในแนวนอนแนวตั้งและแนวทแยง

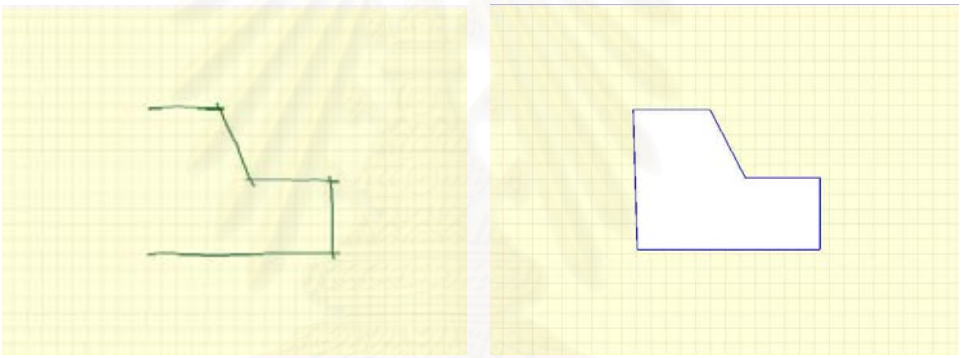
5.1.2 เส้นตัดกัน

5.1.3 ระยะเว้นว่างของเส้น

5.1.4 ระยะยื่นของเส้น

5.1.5 รูปหลายเหลี่ยม

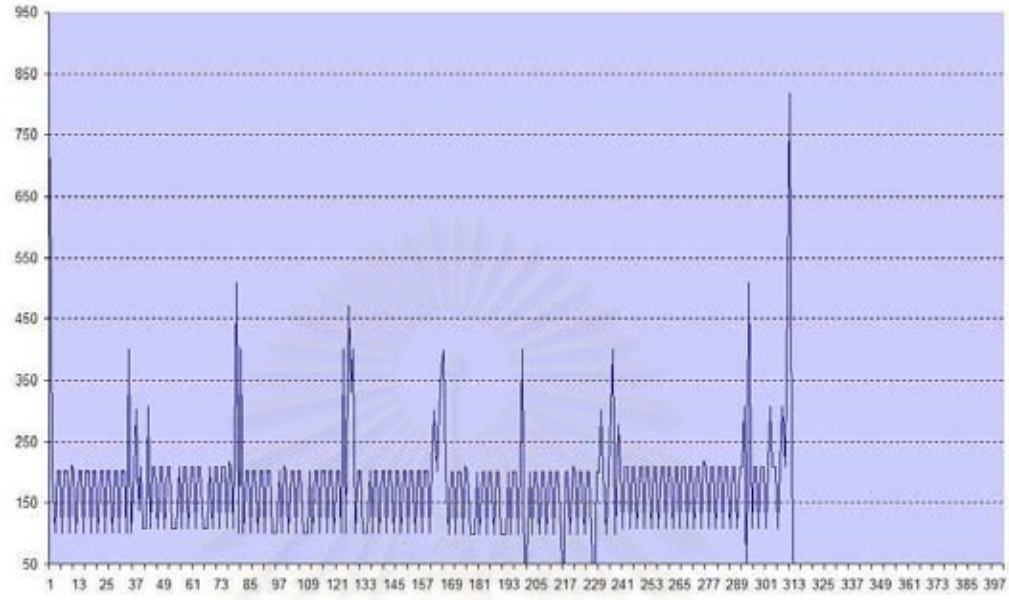
นอกจากนี้ยังมีการทดสอบเพิ่มเติมเพื่อหาความถูกต้องในการรับรู้ภาพวาดลายเส้นที่ซับซ้อนมากขึ้นโดยการใช้ตัวแปร Time Difference เป็นตัววัดค่าจุดมุมของภาพวาดลายเส้น โดยได้ผลการทดสอบดังต่อไปนี้



รูปภาพ 5.2 แสดงการทดสอบการรับรู้ภาพวาดลายเส้นด้วยตัวแปร Time Difference

รูปภาพ 5.3 แสดงความถูกต้องในการรับรู้ ซึ่งโปรแกรมสามารถแปลงภาพวาดลายเส้นให้เป็นภาพ3 มิติได้ทันที

และเมื่อนำข้อมูลจากการวาดภาพ 5.2 มาพิจารณา จะสามารถสร้างเป็นกราฟได้ดังนี้

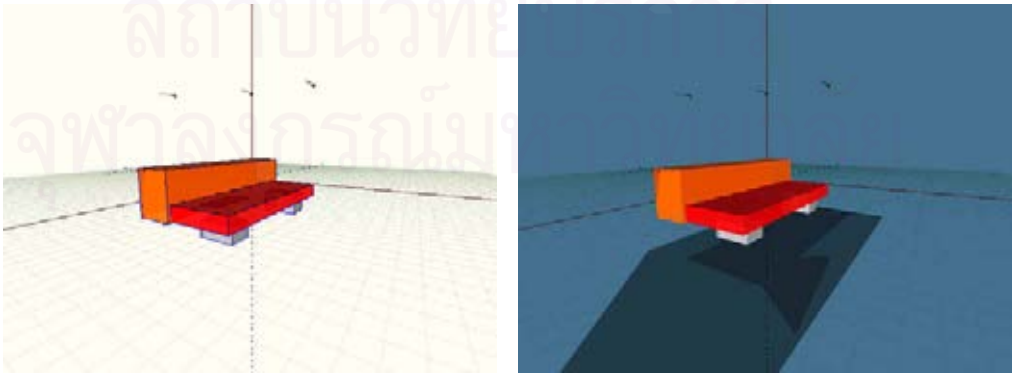


รูปภาพ 5.4 แสดงความแตกต่างของเวลาในการวาดภาพลายเส้น

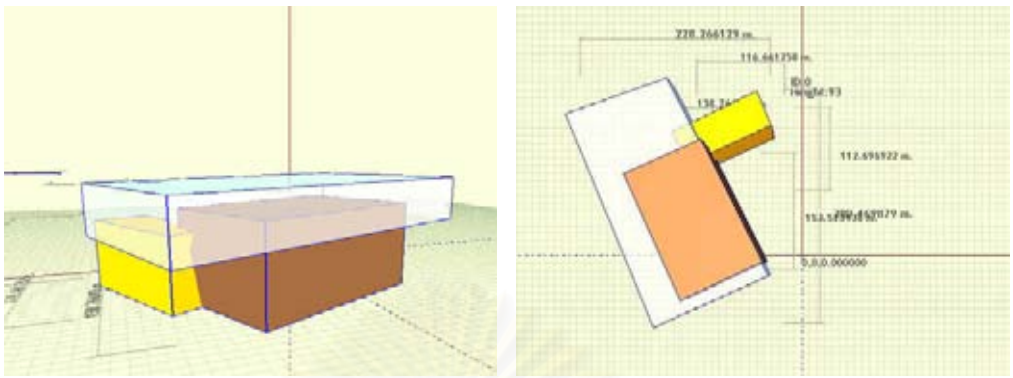
จะเห็นได้ว่าจุดที่เป็นมุมของภาพวาดลายเส้นจะมีค่า Time Difference สูงมากกว่าค่าเฉลี่ย โดยมีความแตกต่างของเวลาเทียบกับค่าเฉลี่ยมากกว่า 100 milliseconds ขึ้นไปฉะนั้นจึงสามารถสรุปค่าตัวแปร Time Difference ที่เหมาะสมได้เท่ากับ 100 milliseconds

5.2 การสร้างรูปทรง 3 มิติเบื้องต้น

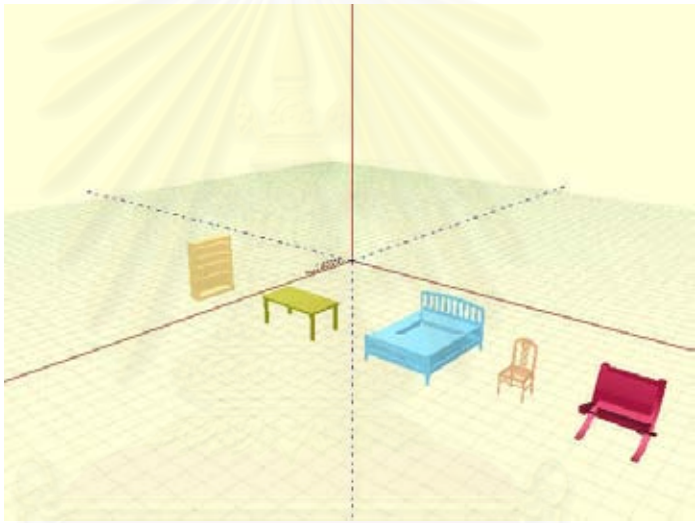
โปรแกรมสามารถนำไปใช้ช่วยสร้างรูปทรง 3 มิติเบื้องต้นได้ โดยการวาดภาพลายเส้นเพื่อสร้างรูปทรง 3 มิติแต่ละชิ้น แล้วนำมาประกอบกันเป็นงานออกแบบได้



รูปภาพ 5.5 ภาพแสดงการประยุกต์ใช้โปรแกรมเพื่อช่วยในการสร้างรูปทรง 3 มิติเบื้องต้น



รูปภาพ 5.6 ภาพแสดงตัวอย่างการสร้างรูปทรง 3 มิติเบื้องต้นเพื่อช่วยในการออกแบบ



รูปภาพ 5.7 ภาพแสดงตัวอย่างการสร้างหุ่นจำลองจากระบบรับรู้รูปแบบที่เหมือนกันของภาพวาดลายเส้น (Sketches Pattern Recognition)

รูปภาพ 5.8 ภาพแสดงตัวอย่างการประยุกต์ใช้ในการออกแบบ

บทที่ 6

สรุปผลการวิจัยและข้อเสนอแนะ

การวาดภาพลายเส้นเป็นวิธีการที่ช่วยในการบันทึกความคิดในการออกแบบของสถาปนิกและเป็นวิธีการทำงานที่สถาปนิกทุกคนมีทักษะพื้นฐานอยู่แล้ว การพัฒนาโปรแกรมรับรู้ภาพวาดลายเส้นจึงเป็นการช่วยเหลือการทำงานของสถาปนิกในยุคที่มีการนำเอาคอมพิวเตอร์มาใช้งานกันอย่างแพร่หลายในสำนักงานสถาปนิก โดยจะช่วยลดช่องว่างในการทำงานระหว่างการเริ่มออกแบบด้วยภาพวาดลายเส้นลงบนกระดาษ กับการพัฒนาแบบจริงด้วยโปรแกรมช่วยเขียนแบบ (CAD) บนคอมพิวเตอร์ ทำให้มีขั้นตอนการทำงานลดลง และยังช่วยให้สถาปนิกสามารถใช้ทักษะการวาดเส้นที่มีอยู่เดิมมาใช้กับโปรแกรมรับรู้ภาพวาดลายเส้นได้ทันที ทำให้ง่ายต่อการเริ่มทำงานโดยไม่ต้องเรียนรู้คำสั่งและขั้นตอนการทำงานจำนวนมากเหมือนโปรแกรม CAD ในการวาดเส้น และจากการพัฒนาโปรแกรมสามารถสรุปผลได้ดังนี้

6.1 สรุปผลการวิจัย

จากการพัฒนาโปรแกรมรับรู้ภาพวาดลายเส้นตามที่ได้ประเมินผลไว้แล้วในบทที่ 5 ซึ่งพบว่าโปรแกรมนี้สามารถช่วยให้สถาปนิกสร้างรูปทรง 3 มิติเบื้องต้นได้จากภาพวาดลายเส้นได้ดี และยังช่วยให้สถาปนิกนำรูปทรง 3 มิติเบื้องต้นมาประกอบกันเป็นงานออกแบบรูปทรงอื่นๆ ได้อีกด้วย โดยนอกจากการรับรูปภาพวาดลายเส้นแล้ว รูปทรงที่สร้างขึ้นยังสามารถปรับแต่งขนาด ความสูง สี ความโปร่งใสได้ทุกระดับ นอกจากนี้ยังสามารถปรับแต่งระยะ grid และ snap สำหรับการวาดภาพด้วยเส้นเรขาคณิต และมีฟังก์ชันในการสร้างผนังห้องเพื่อนำมาประกอบกับรูปทรงที่สร้างขึ้นอีกด้วย เพื่อช่วยให้สามารถนำโปรแกรมไปประยุกต์ใช้กับงานออกแบบได้ง่ายขึ้น และสุดท้ายคือระบบการรายงานผลที่จะช่วยให้ผู้พัฒนาโปรแกรมต่อไปในอนาคต สามารถนำไปใช้วัดพฤติกรรมการวาดภาพลายเส้นของผู้ใช้และใช้วัดความถูกต้องในการทำงานของโปรแกรมได้

เนื่องจากงานวิจัยนี้เป็นการวิจัยเพื่อการสร้างรูปทรง 3 มิติเบื้องต้น โปรแกรมจึงสามารถสร้างภาพเรขาคณิตและแปลงให้เป็นรูปทรง 3 มิติเบื้องต้น แต่ถ้าหากต้องการสร้างรูปทรงที่มีความซับซ้อนและต้องการปรับแต่งรูปทรงจำนวนมาก ในขั้นนี้โปรแกรมยังไม่สามารถสร้างผลลัพธ์ที่น่าพอใจและต้องใช้เวลาในการสร้างรูปทรงที่ซับซ้อนนานมาก และอาจเกิดข้อผิดพลาดจากการทำงานสูง ในอนาคตจึงต้องมีการปรับปรุงโปรแกรมให้มีประสิทธิภาพมากยิ่งขึ้นเพื่อรองรับการทำงานที่ซับซ้อนให้ดียิ่งขึ้น

6.2 ปัญหาและอุปสรรค

ในการวิจัยและพัฒนาโปรแกรมจะพบปัญหาและอุปสรรคหลายด้านทั้งในด้านอุปกรณ์การทำงานและด้านขั้นตอนวิธีที่นำมาพัฒนาโปรแกรม สามารถแบ่งประเภทของปัญหาที่พบได้ดังนี้

6.2.1 ปัญหาด้าน Hardware

- *อุปกรณ์ Input ข้อมูล* ในการพัฒนาโปรแกรมได้เลือกใช้ปากกาอิเล็กทรอนิกส์มาเป็นเครื่องมือสำหรับการ Input ข้อมูล โดยปากกาอิเล็กทรอนิกส์จะทำหน้าที่ส่งสัญญาณจุดที่เกิดจากการวาดภาพลายเส้นเข้าไปในคอมพิวเตอร์ จากนั้นโปรแกรมจะทำหน้าที่รับข้อมูลที่ได้รับมาเพื่อเข้าสู่กระบวนการรับรู้อีกส่วนหนึ่ง

ซึ่งในการส่งสัญญาณจากปากกาอิเล็กทรอนิกส์เข้ามานั้นจะมีความละเอียดของข้อมูลแตกต่างกันตามประสิทธิภาพของปากกาอิเล็กทรอนิกส์ที่เลือกใช้ ซึ่งในการทดลองนี้มีการเลือกใช้อุปกรณ์ที่มีประสิทธิภาพต่ำเกินไปจึงทำให้ได้ผลลัพธ์จากการทำงานที่มีข้อผิดพลาดตามไปด้วย เพราะการรับรู้ภาพวาดลายเส้นจะใช้เวลาในระดับ 1/1000 วินาที เป็นตัววัด ฉะนั้นหากอุปกรณ์ที่ใช้มีประสิทธิภาพต่ำ ก็จะทำให้ข้อมูลที่ได้รับมีความผิดพลาดแฝงอยู่ด้วยและจะนำไปสู่การรับรู้ภาพวาดลายเส้นที่ผิดพลาดในที่สุด

- *ปัญหาด้านความเร็วในการแสดงผล* ความเร็วในการแสดงผลคือจำนวนภาพต่อวินาทีที่เครื่องคอมพิวเตอร์สร้างขึ้นมาแสดงผลทางหน้าจอคอมพิวเตอร์ (Frame Rates) ซึ่งความเร็วนี้จะเกิดจากความเร็วของการ์ดแสดงผลกราฟิกเป็นหลัก (Graphic Card) เพราะในการพัฒนาโปรแกรมมีการใช้คลังกราฟิก (Graphic Library) OpenGL version 1.2[19] ขึ้นไป ซึ่งจะต้องใช้ร่วมกับการ์ดแสดงผลที่ออกแบบมาให้ทำงานร่วมกันโดยเฉพาะ ฉะนั้นในการใช้งานโปรแกรมนี้จึงต้องการ์ดแสดงผลที่เหมาะสม เพื่อสร้างความเร็วในการแสดงผลที่เหมาะกับการทำงานของโปรแกรม (จากการทดสอบพบว่าต้องใช้ความเร็วในการทำงานประมาณ 60 Frames per second ขึ้นไป) แต่ถ้าหากมีความเร็วในการแสดงผลน้อยเกินไป จะทำให้มีการรับรู้จำนวนจุดจากการวาดภาพลายเส้นน้อยกว่าปกติและเกิดข้อผิดพลาดในการรับรู้ได้

[19] Siligon Graphic Inc. http://www.opengl.org

6.2.2 ปัญหาด้าน Software

- *ปัญหาความถูกต้องในการรับรู้ภาพวาดลายเส้น* ในการรับรู้ภาพวาดลายเส้นมีการใช้ขั้นตอนวิธีในการรับรู้หลายวิธี บางวิธีจะมีลักษณะในการรับรู้คล้ายกับการรับรู้ตัวอักษรของอุปกรณ์บันทึกข้อมูลส่วนตัว (PDA) ซึ่งพบว่ายังไม่มีอุปกรณ์หรือขั้นตอนวิธีใดสามารถรับรู้ภาพวาดลายเส้นได้ถูกต้องในระดับเดียวกับการรับรู้ของมนุษย์[20] จึงทำให้เกิดข้อจำกัดในการใช้งานโปรแกรมที่พัฒนาขึ้น โดยโปรแกรมต้องการการฝึกฝนของผู้ใช้ให้เกิดความคุ้นเคยในระยะเวลาหนึ่ง เพื่อให้สามารถวาดภาพลายเส้นที่มีความผิดพลาดไม่เกินระดับที่โปรแกรมสามารถรับรู้ได้

- *ปัญหาการวาดภาพลายเส้นในมุมมอง 3 มิติ* เนื่องจากการพัฒนาโปรแกรมใช้คลังกราฟิก OpenGL version 1.2 [21] นอกเหนือจากการวาดเส้นในมุมมอง 2 มิติซึ่งรับค่าโดยตรงจากอุปกรณ์ Input แล้ว หากมีการเปลี่ยนมุมมอง 3 มิติจะทำให้การรับค่าพิกัดจากอุปกรณ์ Input ในมุมมอง 3 มิติจะเกิดความสัมพันธ์กับตำแหน่งของมุมกล้องในขณะนั้น โดยใช้ฟังก์ชันในการแปลงของ OpenGL แปลงพิกัดของอุปกรณ์ Input ให้อยู่ในระบบทัศนียภาพ 3 มิติขณะนั้น ทำให้ค่าที่ได้จากอุปกรณ์วาดภาพลายเส้นที่ส่งเข้ามาจะถูกแปลงให้อยู่ในพิกัด 3 มิติของ OpenGL อยู่ตลอดเวลา

ฉะนั้นเมื่อมีการแปลงข้อมูลอยู่ตลอดเวลาแล้ว การวาดเส้นอิสระจึงจะเกิดความผิดพลาดสูง ในการพัฒนาโปรแกรมจึงยังไม่สามารถวาดภาพลายเส้นแบบอิสระในมุมมอง 3 มิติได้แต่สามารถวาดเส้นแบบเรขาคณิตด้วยคำสั่ง Line ในมุมมอง 3 มิติแทน เพื่อความถูกต้องของการรับรู้ภาพวาดลายเส้น

- *ปัญหาการเลือกใช้ขั้นตอนวิธีในการรับรู้ภาพวาดลายเส้น* ในการเลือกใช้ขั้นตอนวิธีในการรับรู้ภาพวาดลายเส้นซึ่งมีหลายวิธีดังที่ได้กล่าวมาแล้วในบทที่ 4 โดยแต่ละวิธีก็มีลักษณะเด่นและคุณสมบัติในการรับรู้ภาพวาดลายเส้นแตกต่างกันออกไป ดังนั้นในการเลือกขั้นตอนวิธีมาใช้กับงานวิจัยนี้จึงเกิดปัญหาเรื่องความเหมาะสมของขั้นตอนวิธีขึ้น เนื่องจากบางวิธีเช่น การรับรู้จุดมุมของภาพ Time Difference ใช้เวลาเป็นตัว

---

[20] R. J. Shillman (1974), Character recognition based on phenomenological attributes, theory and methods. Ph. D. Thesis, Massachusetts Institute of Technology.

[21] Silicon Graphic Inc. http://www.opengl.org

วัดเพื่อรับรู้ภาพวาดลายเส้น แต่วิธีการหาระยะยื่นและจุดตัดจะใช้คณิตศาสตร์เป็นตัววัด จึงทำให้เกิดการรับรู้ภาพวาดลายเส้นที่แตกต่างกันและทำให้ไม่สามารถนำวิธีการทั้งหมดมารวมกันเพื่อสร้างขั้นตอนที่ใช้ร่วมกันได้เนื่องจากความแตกต่างกันในเรื่องตัวแปรที่ใช้วัดตามที่ได้กล่าวข้างต้น การพัฒนาโปรแกรมจึงต้องกำหนดให้มีการใช้ขั้นตอนวิธีรับรู้แต่ละวิธีกับลักษณะการวาดเส้นอย่างใดอย่างหนึ่ง เช่น Time Difference กับลักษณะการวาดเส้นแบบต่อเนื่องโดยไม่ยกปากกา หรือ วิธีการหาระยะยื่นใช้กับการวาดเส้นแบบยกปากกาวาดที่ละเส้น เป็นต้น

- *ปัญหาด้านการออกแบบส่วนโต้ตอบกับผู้ใช้ (User Interface)* ในการออกแบบส่วนโต้ตอบกับผู้ใช้จากการศึกษางานวิจัยเรื่อง Free Form User Interface (Takeo Igarashi, 1999) ซึ่งชี้ให้เห็นว่าการพัฒนาส่วนโต้ตอบผู้ใช้ของโปรแกรมทางด้านการวาดภาพลายเส้น จะต้องตอบสนองต่อการทำงานซึ่งเกี่ยวกับความคิดสร้างสรรค์ เป็นความคิดที่จะเกิดขึ้นอย่างรวดเร็ว จึงไม่ต้องการความซับซ้อนของขั้นตอนการทำงาน เช่น การทำงานที่ต้องกดปุ่มจำนวนมาก หรือต้องพิมพ์คำสั่งหลายขั้นตอน จึงเลือกออกแบบส่วนโต้ตอบกับผู้ใช้ตามแนวคิดดังกล่าว แต่ทำให้เกิดปัญหาตามมาคือในการประยุกต์ใช้งานนั้นผู้ใช้มักจะคาดหวังเครื่องมือช่วยเหลือในการทำงานให้เกิดความรวดเร็วเสมอ ฉะนั้นหากไม่มีปุ่มคำสั่งที่ช่วยเหลือในการทำงานให้รวดเร็วขึ้นก็จะทำให้โปรแกรมไม่สามารถนำไปประยุกต์ใช้กับการทำงานจริงได้ฉะนั้นจึงเกิดข้อขัดแย้งขึ้นระหว่างแนวคิดในการออกแบบส่วนโต้ตอบกับผู้ใช้ที่เหมาะสมกับการวาดภาพลายเส้น และการนำโปรแกรมมาประยุกต์เพื่อช่วยเหลือในการทำงาน

6.3 ข้อเสนอแนะ

6.3.1 การเลือกอุปกรณ์ Hardware

ในการพัฒนาโปรแกรมขั้นต่อไปน่าจะเป็นการพัฒนาโปรแกรมเพื่อใช้กับอุปกรณ์ที่มีข้อจำกัดด้านอุปกรณ์ Input น้อย มีความแม่นยำสูง เช่น Tablet PC หรือ PDA หากสามารถพัฒนาโปรแกรมให้ทำงานบน Tablet PC ได้น่าจะทำให้โปรแกรมมีประโยชน์ต่อการทำงานของสถาปนิกต่อไปในอนาคตเพราะอุปกรณ์ดังกล่าวเป็นอุปกรณ์ชนิดใหม่ที่เน้นการทำงานด้วยการใช้ปากกาอิเล็กทรอนิกส์ วาดภาพลงบนหน้าจอคอมพิวเตอร์โดยตรง จึงน่าจะทำให้มีความความผิดพลาดในการทำงานน้อยและน่า

จะทำให้สถาปนิกสามารถพกพาอุปกรณ์ชนิดนี้พร้อมโปรแกรมที่พัฒนาขึ้นไปตามสถานที่ต่างๆ เพื่อช่วยในการรับรู้ภาพวาดลายเส้นได้

รูปภาพ 6.1 ภาพแสดงตัวอย่างอุปกรณ์ Tablet PC (แหล่งที่มา: http://www.toshiba.com)

6.3.2 การพัฒนาโปรแกรมในอนาคต

- การพัฒนาส่วนเพิ่มเติมของโปรแกรม

เนื่องจากโปรแกรมนี้มีการพัฒนาด้วยหลักการ Object Oriented Programming ทำให้สามารถนำเอาโปรแกรมที่พัฒนาเพิ่มเติมมาเพิ่มเข้าไปได้ทันที โดยสร้างชิ้นส่วน (Module) ใหม่มาประกอบเข้ากับโปรแกรมเดิม ซึ่งในการพัฒนาโปรแกรมในขั้นนี้ได้สร้าง Module การวาดเส้นโค้ง การสร้างเงา การเขียนตัวอักษร แยกเป็นส่วนๆ ไว้และสามารถเพิ่มเติมได้ในอนาคต จึงน่าจะมีการพัฒนาโปรแกรมเพิ่มเติมเป็นชิ้นส่วนอื่นๆ เช่น การวาดวงกลม การสร้างรูปตัดจากภาพวาดลายเส้น หรือการเพิ่มขั้นตอนวิธีการรับรู้ใหม่เข้าไปก็น่าจะทำได้

- *การปรับการรับรู้ด้วยการวัดความเร็วของปากกา* จากการทดสอบและประเมินผลโปรแกรมทำให้มองเห็นข้อจำกัดของการใช้เวลาเป็นตัววัดในการรับรู้จุดมุมของภาพ เพราะตัวแปรเวลาดังกล่าวจะใช้ได้กับความเร็วในการวาดเส้นระดับหนึ่ง หากผู้ใช้วาดเส้นที่ความเร็วหลายระดับแตกต่างกันไประหว่างการวาดภาพทั้งหมด โปรแกรมจะรับรู้ผิดพลาดเพราะไม่สามารถใช้ตัวแปรจับเวลาเดียวกันกับทุกความเร็วได้ เพราะผู้ใช้แต่ละคนจะมีความเร็วในการวาดต่างกัน หากนำตัวแปรความเร็วมาช่วยในการวัดการวาดภาพลายเส้นแล้วก็น่าจะทำให้โปรแกรมสามารถปรับเปลี่ยนตัวแปรเวลาที่

ใช้วัดได้ด้วยตัวเอง เช่น หากผู้ใช้วาดภาพลายเส้นด้วยความเร็วระดับหนึ่ง โปรแกรมจะทำการวัดความเร็วเพื่อหาค่าตัวแปรการจับเวลาที่เหมาะสมด้วยตัวเอง และเมื่อมีความเร็วเพิ่มขึ้นก็จะใช้ตัวแปรการจับเวลาอีกค่าหนึ่ง เป็นต้น ทั้งนี้น่าจะทำให้โปรแกรมมีความสามารถในการรับรู้เพิ่มขึ้นและเป็นการพัฒนาโปรแกรมให้สามารถช่วยเหลือการทำงานของสถาปนิกได้ดีขึ้นอีกด้วย

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

ภาษาไทย

กิตติมา มะแก้ว. การแปลงข้อมูลภาพแบบราสเตอร์เป็นแบบเว็กเตอร์สำหรับรูปภาพกราฟิกเบื้องต้น. กรุงเทพฯ: ภาควิชาวิทยาศาสตร์คอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, พ.ศ. 2538.

ธิดาสิริ ภัทรากาญจน์. Design plus Digital: สถาปัตยกรรมกับการออกแบบสร้างสรรค์อย่างดิจิทัล. กรุงเทพฯ: โรงพิมพ์สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ, พ.ศ. 2546.

ธรรมศักดิ์ โกศลกิจจา. การสร้างภาพเคลื่อนไหวสองมิติโดยวิธีภาพหลัก. กรุงเทพฯ: ภาควิชาวิทยาศาสตร์ คอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, พ.ศ. 2532.

ปุณณรัตน์ พิชญไพบูลย์. Computer Graphics: สำหรับนักออกแบบ. กรุงเทพฯ: สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย, พ.ศ. 2542.

เลอสม สถาปิตานนท์. การออกแบบคืออะไร. พิมพ์ครั้งที่ 1 กรุงเทพฯ: 49 กราฟฟิค & พับบลิเคชั่นส์, พ.ศ. 2537.

วิญญู วานิชศิริโรจน์. IT Management in Architectural Firm เอกสารในการสัมมนาเรื่อง Computer Graphics in Architectural Design Process เสนอที่ ศูนย์นิทรรศการและการประชุมไบเทค 21 กรกฎาคม 2544.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาษาอังกฤษ

Antony Radford, Garry Stevens. CAAD Made Easy. USA: McGraw Hill, 1987.

Armin Hofman. Graphic Design Manual / Principles and Practice. Zurich: Reinhold, 1965.

Dae-Hyun Kim, Paul Michalik, Stephan Hofer. Quick Sketch[Online]. Available from: http://rabbit.prakinf.tu-ilmenau.de/qsketch.html [2002, August 22]

Eric Saund, Thomas P. Moran. A Perceptually-Supproted Sketch Editor[Online]. Available from: http://www.cs.virginia.edu/~acc2a/techie/notes/hci/SaundMoran.htm [2002, Aug 22]

Fatos Bengi Durgun, Bülent Özgüc. Architectural Sketch Recognition. USA: Architectectural Science Review, 1990.

Frank Ching. Architectural Graphics. New York: Van Nostrand Reinhold, 1985.

Francis D.K. Ching. Drawing: A Creative Process. New York: Van Nostrand Reinhold, 1990.

Francis D.K. Ching. Design Drawing. New York: Van Nostrand Reinhold, 1998.

Gerard Cesar Gabriel, Mary Lou Maher. Does Computer Mediation Affect Design Representation? Sydney: University of Sydney, 1998.

James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes. Computer Graphics principles & practice, second edition. India: Pearson Education, 2003.

Jin Xiangyu, Liu Wenyin, Sun Jianyong, Zhengxing Sun. On-Line Graphics Recognition. Nanjing: Department of Computer Science, NanJing University, 2002.

John Rieman. Testing and Revising JSketch: A Drawing Tool for Informal Graphics. Pittsburgh: ACM Conference On Human Factors in Computing Systems, 1999.

Mark D. Gross, Ellen Yi-Luen Do. The Electronic Cocktail Napkin Project[Online]. Available from: http://depts.washington.edu/napkin/ [2002, Aug 22]

Mark Von Wodtke. Design with digital tools. USA: McGraw – Hill, 2000.

Michael Trinder. The Computer's Role in Sketch Design: A Transparent Sketching Medium[Online]. Available from: http://www.arct.cam.ac.uk/research/pubs/pdfs/trinder99a.pdf [2002, Aug 22]

Paul Laseau. Graphic Thinking For Architects And Designers. Second Edition New York: Van Nostrand Reinhold, 1989.

Peter Agar, Kevin Novins. ACM Proceedings:Computer graphics and interactive techniques in Austalasia and South East Asia: Polygon Recognition in Sketch-Based Interfaces with Immediate and Continuous Feedback. New York: ACM Press, 2003. p. 147-150.

Robert C. Zeleznik, Kenneth Herndon, John F. Hughes. SKETCH: An interface for sketching 3D scenes. Computer Graphics(SIGGRAPH'96 Proceedings). USA: ACM SIGGRAPH, 1994.

Robert W. Gill. Manual of Rendering with Pen and Ink. New York: Van Nostrand Reinhold, 1973.

Ron Kasprisin, James Pettinari. Visual Thinking for Architects and Designers. New York: Van Nostrand Reinhold, 1995.

Takeo Igarashi. A Suggestive Interfaces for 3D Drawing. USA: Computer Science Department, Brown University, 2001.

Takeo Igarashi. Freeform User Interfaces for Graphical Computing. Tokyo: Graduate School of Information Engineering, The University of Tokyo, 1999.

Tom Porter. Graphic Design Techniques for Architectural Drawing. Hong Kong: Hamlyn, 1990.

Tsuyoshi Sasada. Computer Graphics and Design: Presentation, Design Development and Conception. CAADRIA'99[Online]. Available from: http://http://itc.fgg.uni-lj.si/data/cumincad/robots/4827.htm [2002, Aug 22]

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

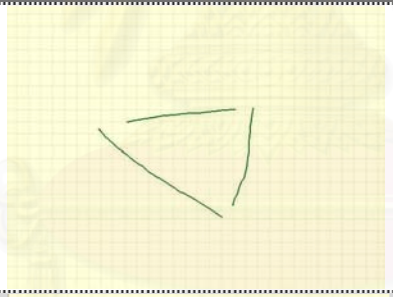
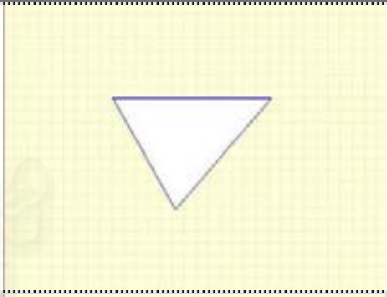
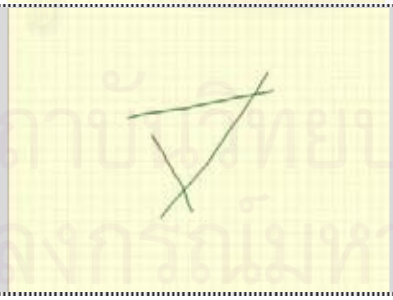
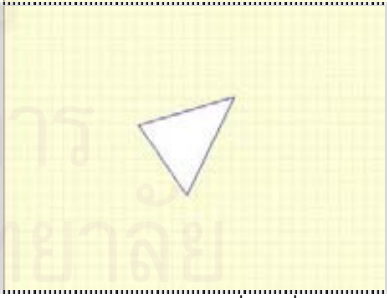
ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

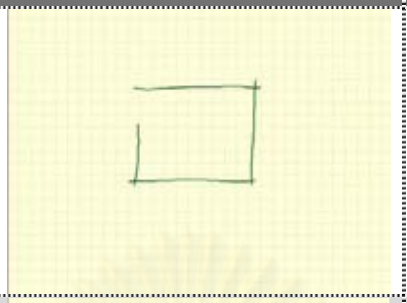
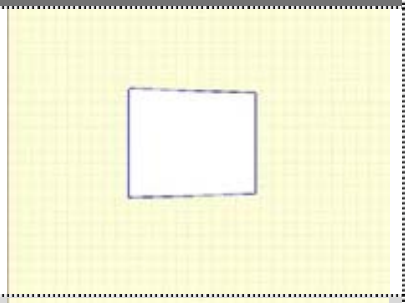
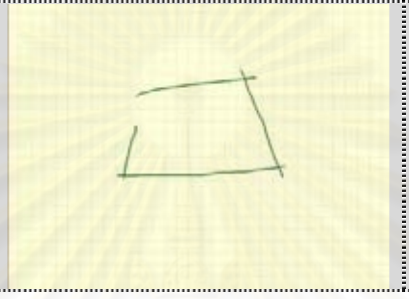
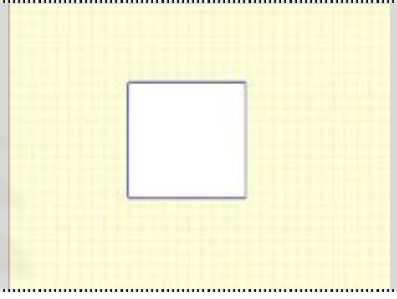
1. การทดลองหาผลจากการปรับค่าตัวแปรในการรับรู้ภาพวาดลายเส้น

นอกเหนือจากการทดสอบในบทที่ 5 แล้วโปรแกรมนี้ยังสามารถปรับแต่งค่าตัวแปรเพื่อสร้างผลลัพธ์ที่แตกต่างกันไปได้ โดยการปรับแต่งค่าตัวแปรแต่ละชนิดจะทำให้เกิดการรับรู้แต่ละวิธีแตกต่างกันไปดังต่อไปนี้ (ยกเว้นการปรับแต่งตัวแปร Time Difference ซึ่งมีการทดสอบความถูกต้องอยู่ในบทที่ 5 แล้ว และตัวแปร Pattern Recognition ซึ่งไม่สามารถปรับแต่งได้)

- *Extend Recognition* คือตัวแปรในการรับรู้ระยะเว้นว่างของเส้น ค่าเริ่มต้นเท่ากับ 40
- *Length Recognition* คือตัวแปรในการรับรู้ระยะยื่นของเส้น ค่าเริ่มต้นเท่ากับ 40
- *Parallel Recognition* คือตัวแปรในการรับรู้ความขนานกันของเส้น ค่าเริ่มต้นเท่ากับ 5
- *Curve Recognition* คือการปิดเปิดระบบรับรู้เส้นโค้ง ค่าเริ่มต้นเท่ากับ 5

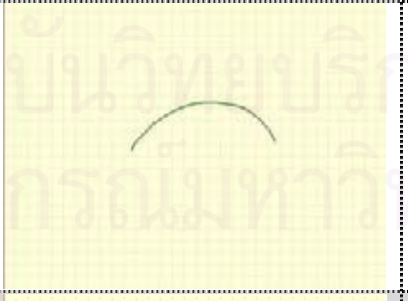
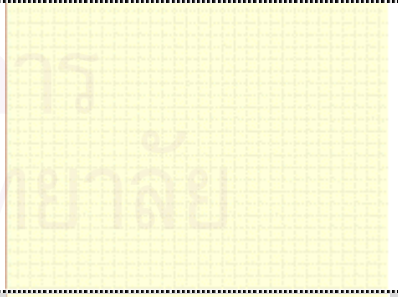
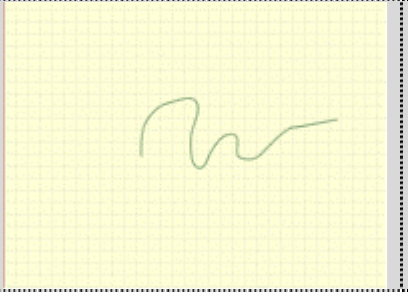
ชื่อตัวแปร	ค่าเริ่มต้น	ผลการทดสอบ	ค่าปรับแต่ง	ผลการทดสอบ
Extend Recognition	40		100	
Length Recognition	40		100	

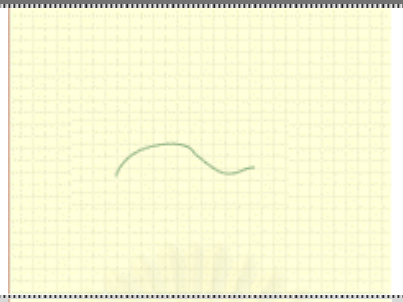
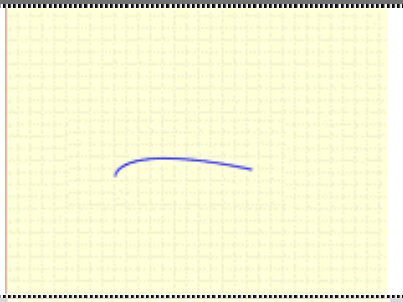
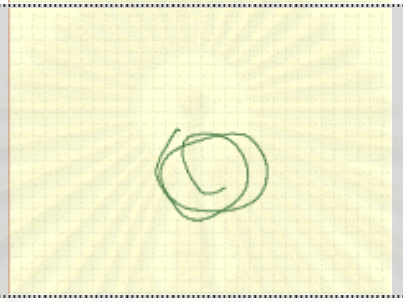
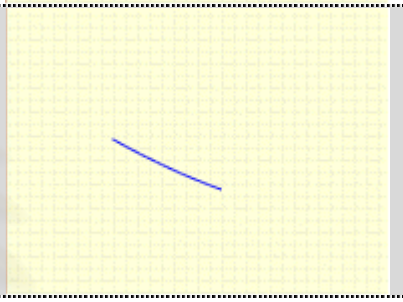
จากการทดสอบจะเห็นว่าตัวแปร Extend Recognition และ Length Recognition ที่ค่าเริ่มต้นจะไม่สามารถรับรู้ภาพวาดลายเส้น ที่มีระยะห่างของจุดปลายตามเงื่อนไขการรับรู้เกินกว่าค่าที่กำหนด (กำหนดค่าเริ่มต้นให้เป็น 40) และเมื่อมีการปรับแต่งค่าตัวแปร (ปรับให้เป็น 100) ก็จะสามารถรับรู้ภาพวาดลายเส้นได้ถูกต้อง

ชื่อตัวแปร	ค่าตัวแปร	ภาพวาดลายเส้น	ผลการรับรู้
Parallel Recognition	5		
Parallel Recognition	30		

ในการปรับแต่งตัวแปร Parallel Recognition จะเห็นความแตกต่างในการรับรู้ได้อย่างชัดเจนโดยกำหนดค่าเริ่มต้นไว้ที่ 5 เมื่อวาดภาพลายเส้นที่มีความเอียงของเส้นมากกว่า 5 ผลการรับรู้จะได้ภาพสี่เหลี่ยมที่มีความเอียงตามความชันของเส้นที่วาดซึ่งไม่เป็นเส้นตั้งฉาก

แต่เมื่อมีการปรับแต่งตัวแปรเป็น 30 ก็จะทำให้การวาดภาพลายเส้นที่มีความลาดเอียงมาก จะถูกรับรู้ให้เส้นขนานกับแกน X และแกน Y กลายเป็นเส้นตั้งฉากและเส้นนอนได้ ซึ่งจะทำให้สามารถวาดรูปสี่เหลี่ยมมุมฉากได้โดยสะดวก

ชื่อตัวแปร	ค่าตัวแปร	ภาพวาดลายเส้น	ผลการรับรู้
Curve Recognition	5		
Curve Recognition	5		

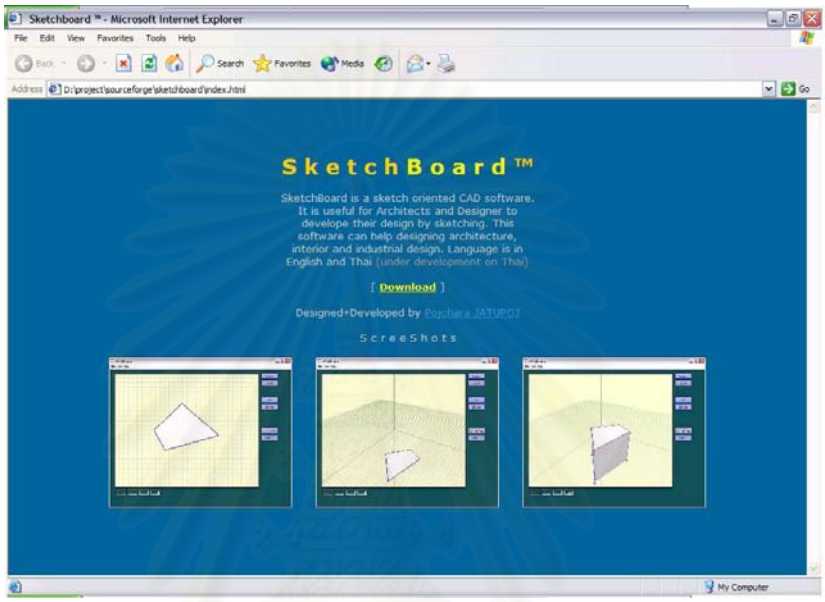
ชื่อตัวแปร	ค่าตัวแปร	ภาพวาดลายเส้น	ผลการรับรู้
Curve Recognition	0		
Curve Recognition	30		

จากผลการทดสอบจะเห็นว่าค่าตัวแปร Curve Recognition หากมีค่าน้อย ก็จะสามารถรับรู้ได้ถูกต้อง แต่หากมีค่ามากก็จะสามารถรับรู้ Curve ได้ถูกต้องน้อยลง จากการกำหนดค่าเริ่มต้นที่ 5 จะเห็นว่าโปรแกรมไม่สามารถรับรู้เส้นโค้งที่มีความชันแตกต่างกันคงที่ แต่จะรับรู้เส้นโค้งที่มีความชันแตกต่างกันสูง และเมื่อปรับแต่งตัวแปรให้เป็น 0 จะเห็นว่าโปรแกรมสามารถรับรู้และสร้างเส้นโค้งได้ใกล้เคียงกับการวาดภาพลายเส้นมากที่สุด และเมื่อปรับแต่งตัวแปรให้เป็น 30 จะเห็นว่าโปรแกรมจะรับรู้เฉพาะภาพวาดลายเส้นที่มีความชันแตกต่างกันสูงมาก เช่นเส้นโค้งที่ลากเป็นวงกลม เป็นต้น

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

2. การเผยแพร่งานวิจัยและโปรแกรมผ่านทางเว็บไซต์ Sourceforge.net

ในการพัฒนาโปรแกรมได้มีการเผยแพร่ข้อมูลของขั้นตอนวิธีพัฒนาโปรแกรม ผ่านทางเว็บไซต์ http://sketchboard.sourceforge.net ซึ่งเป็นกลางการเผยแพร่ข้อมูลการพัฒนาโปรแกรมแบบเปิด (Open Source) ทั้งนี้เพื่อสนับสนุนให้มีการเผยแพร่ความรู้สู่สาธารณะ และเปิดเผยขั้นตอนวิธีในการพัฒนาโปรแกรมรับรู้ภาพวาดลายเส้นให้แก่ผู้ที่สนใจนำไปพัฒนาต่อได้อย่างอิสระ



ผู้ที่สนใจดาวโหลดโปรแกรมสามารถดาวโหลดได้โดยการกดปุ่ม Download แล้วเลือก โปรแกรมชื่อ Sketchboard_exe_1.36.zip จากนั้นให้ Extract แฟ้มเพื่อเปิดโปรแกรม Sketchboard

หากมีข้อสงสัยเกี่ยวกับการพัฒนาโปรแกรมสามารถติดต่อได้ที่ E-Mail : pojchara@asa.or.th

ประวัติผู้เขียนวิทยานิพนธ์

ชื่อ – สกุล นายพชระ จาฏุพจน์

วัน เดือน ปีเกิด 22 เมษายน พ.ศ.2519

ที่อยู่ สวนต้นไม้เสาวลักษณ์ ถ.พโลรังฤทธิ์ ตำบลในเมือง อ.เมือง จ.อุบลราชธานี

ประวัติการศึกษา

2540 สถาปัตยกรรมศาสตรบัณฑิต ภาควิชาสถาปัตยกรรม คณะสถาปัตยกรรมศาสตร์ มหาวิทยาลัยขอนแก่น จ.ขอนแก่น

2546 สถาปัตยกรรมศาสตรมหาบัณฑิต กลุ่มวิชา คอมพิวเตอร์ในการออกแบบสถาปัตยกรรม สาขาวิชาสถาปัตยกรรม คณะสถาปัตยกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ประวัติการทำงาน

2540-2542 เจ้าหน้าที่สารสนเทศ สมาคมสถาปนิกสยามฯ กรุงเทพมหานคร

2542-2544 Web Designer บริษัท Samart Internet จำกัด กรุงเทพมหานคร

2544-2545 Senior Web Designer บริษัท Netfusion Loxbit จำกัด กรุงเทพมหานคร

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รางวัลที่ได้รับ

2544 Best General Business Website, Thailand Web Award 2002, http://www.siamfibrecement.com

2544 รางวัลชมเชย ASA Experimental Design “Visionary of Future” Bangkok, Thailand จากสมาคมสถาปนิกสยามฯ