

บทที่ 1

บทนำ



### 1.1 ความเป็นมาของปัญหา

ปัจจุบันได้มีการนำคอมพิวเตอร์มาใช้งานภายในประเทศไทยกันอย่างกว้างขวาง ตั้งแต่การใช้งานส่วนบุคคลไปจนถึงหน่วยงานขนาดใหญ่ สื่อหรือข้อมูลที่ใช้กันไว้ในเครื่องคอมพิวเตอร์นั้น แต่เดิมจะถูกจัดวางให้ใช้เฉพาะในแบบ(mode)ตัวอักษร (text) เป็นส่วนมาก ส่วนในแบบกราฟิก (graphic) จะมีการใช้งานไม่มากนัก และมักเป็นงานที่ไม่เน้นด้านการแสดงข้อความต่างๆ เนื่องจากประสิทธิภาพของเครื่องที่ยังไม่ดีพอ ซึ่งทำให้การใช้งานมีข้อจำกัด เช่น การเปลี่ยนแปลงขนาดของตัวอักษร การผสมอักษรกับรูปภาพ การหมุน(rotate) หรือการไ้(shear) ตัวอักษร เป็นต้น

อย่างไรก็ดี ในปัจจุบันเครื่องคอมพิวเตอร์ในระดับไมโครคอมพิวเตอร์ รวมถึงระดับสถานีงาน (work station) ได้มีประสิทธิภาพเพิ่มขึ้นเป็นอย่างมาก ทั้งในด้านความเร็วและปริมาณหน่วยความจำหลัก แนวโน้มการใช้งานและพัฒนาโปรแกรมจึงเริ่มเปลี่ยนเป็นการทำงานในแบบกราฟิก ซึ่งมีรูปแบบการแสดงผลที่สวยงามและยืดหยุ่นกว่า เช่น การนำมาใช้งานเกี่ยวกับงานพิมพ์ตั้งโต๊ะ (desktop publishing) ซึ่งเป็นการผสมผสานระหว่างข้อความและรูปภาพในแบบกราฟิก จึงเริ่มเป็นที่นิยมในปัจจุบัน

นิยาม ก

เส้นบาง (Thin line) คือเส้นซึ่งมีความยาวประกอบขึ้นจากจุดจำนวนหนึ่งและมีความหนาเพียงจุดเดียว

นิยาม ข

รูปจำกัด (simple picture) ในที่นี้จะหมายถึงรูปสองมิติซึ่งประกอบขึ้นจากเส้นบาง ซึ่งอาจเป็นเส้นตรงหรือเส้นโค้ง โดยไม่นับรวมถึงรูปที่เป็นพื้นที่ (area) และจะไม่มีการระบายสี หรือการแบ่งระดับสีเทา (gray scale)

## นิยาม ค

เส้นขอบ (outline) ของรูปใดๆ จะกำหนดโดยจุดซึ่งอยู่ระหว่างพื้นที่ภายในและภายนอกของรูปนั้น

การสร้างแบบของตัวอักษร หรือเรียกในทางคอมพิวเตอร์ว่าเป็นการสร้างฟอนต์ (font) นั้น เราอาจแบ่งได้เป็น 2 วิธี (MARC BERGER 1986) คือ การสร้างในแบบเวกเตอร์ เรียกว่า สโตรกฟอนต์ (stroke font) ซึ่งกำหนดตัวอักษรจากชุดของเส้นตรงซึ่งอยู่ในรูปแบบสมการทางคณิตศาสตร์ และแบบราสเตอร์เซลล์ (raster cell) หรือที่เรียกกันว่าฟอนต์จุดภาพ (bitmap font) ซึ่งกำหนดตัวอักษรขึ้นจากตารางของจุด

ในการสร้างฟอนต์ในแบบกราฟิก ปัจจุบันมักใช้แบบจุดภาพกันเป็นส่วนมาก เนื่องจากสามารถสร้างได้ง่ายกว่าและมีความเร็วในการแสดงผลที่ดีกว่า แต่ปัญหาที่พบในการสร้างฟอนต์ชนิดนี้ก็คือ การขยายฟอนต์ทำได้แต่การขยายให้ใหญ่ขึ้นและเป็นจำนวนเท่าของต้นแบบเท่านั้น นอกจากนี้ฟอนต์ที่ขยายขึ้นจะมีรายละเอียดที่หยาบ อีกทั้งการหมุนหรือการโย้ฟอนต์ก็ทำได้ยากและไม่มีประสิทธิภาพ ในขณะที่ฟอนต์แบบเวกเตอร์จะมีความยืดหยุ่นมากกว่า เนื่องจากฟอนต์แบบนี้จะสามารถขยายเป็นขนาดใดๆ ก็ได้ และสามารถใช้กับอุปกรณ์แสดงผลใดๆ (จอภาพหรือเครื่องพิมพ์) และที่ความละเอียดต่างๆ (เช่น 120 จุดต่อนิ้วในเครื่องพิมพ์แบบเข็มกระทบ หรือ 300 จุดต่อนิ้วในเครื่องพิมพ์แบบเลเซอร์) ซึ่งมีเพียงเวกเตอร์ฟอนต์เท่านั้นที่สามารถแสดงผลในแบบ "สิ่งที่เห็นคือสิ่งที่ได้รับ (What You See Is What You Get)" อย่างสมบูรณ์ (CHARLES PETZOLD 1990) ซึ่งมีความจำเป็นสำหรับงานประเภทโรงพิมพ์ตั้งโต๊ะ (Desktop publishing) เป็นอย่างมาก

งานวิจัยเกี่ยวกับเวกเตอร์ฟอนต์ที่สำคัญงานหนึ่ง คือการพัฒนาเครื่องมือในการสร้างเวกเตอร์ฟอนต์ ที่เรียกว่า METAFONT ซึ่งพัฒนาโดยทีมงานของ ศาสตราจารย์ Donald E. Knuth ที่ Stanford University และมีการนำมาใช้เพื่อสร้างฟอนต์ให้แสดงผลภาษาไทยในแบบเวกเตอร์ โดยทีมงานของ ดร.ชิตชนก เหลือสินทรัพย์ที่ University of Southwestern Louisiana (ชิตชนก เหลือสินทรัพย์ 2530) โดยที่ระบบ METAFONT เป็นซอฟต์แวร์ซึ่งช่วยในการสร้างแบบตัวอักษรในงานเรียงพิมพ์ ซึ่งจะต้องบรรยายโครงสร้างของตัวพิมพ์ โดยอาศัยสมการคณิตศาสตร์และโครงสร้างทางภาษาแบบภาษาทางโปรแกรมทั้งหลาย ซึ่งจะมีปัญหาคือ ถ้ามีตัวพิมพ์ลักษณะแปลกอยู่ตัวหนึ่ง ทำอย่างไรจึงจะหาสมการคณิตศาสตร์ เพื่อจะบรรยายลักษณะของเส้นโค้งให้มีความใกล้เคียงกับตัวพิมพ์ตัวนั้นมากที่สุด และเราจะแบ่งและแยกส่วนตัวพิมพ์นั้นเป็นกี่ส่วนจึงเหมาะสม (ชิตชนก เหลือสินทรัพย์ 2530) ซึ่งจะต้องใช้เวลาและความรู้ในการสร้างพอสมควร อีกทั้งเป็นการใช้งานที่เน้นทางด้านงานเรียงพิมพ์และแสดงผลตัวอักษร ไม่ได้เป็น

การนำมาใช้แสดงผลทางจอภาพเป็นรูปสัญลักษณ์ต่างๆ

งานวิจัยนี้ จะเป็นการศึกษาถึงวิธีที่เหมาะสมสำหรับการแทนรูปสัญลักษณ์ต่างๆ ด้วยพอนต์ในแบบเวกเตอร์ ซึ่งอาจเป็นตัวอักษรใดๆ หรือรูปต่างๆ ก็ได้ โดยที่รูปนั้นจะต้องเป็นรูปจำกัดและจะแทนที่รูปนั้นในแบบเวกเตอร์ อนึ่ง ในการแสดงผลจะเป็นการแสดงผลออกทางจอภาพเท่านั้น ซึ่งจากการที่สามารถแทนรูปจำกัดด้วยเวกเตอร์พอนต์ได้ การนำมาแทนตัวอักษรภาษาไทยก็เป็นสิ่งที่กระทำได้เช่นกัน และเนื่องจากภาษาไทยมีลักษณะเฉพาะตัวที่แตกต่างจากภาษาอื่น คือ ตัวอักษรในภาษาไทยมักจะมีลักษณะโค้งมนเป็นส่วนมาก เป็นเหตุให้การสร้างเวกเตอร์พอนต์จากเส้นตรง จะมีความไม่เหมาะสมทั้งด้านความสวยงามและประสิทธิภาพ จึงสมควรจะมีการศึกษาและพัฒนาวิธีในการสร้างพอนต์ตัวอักษรให้เหมาะสมกับภาษาไทยด้วย เช่น การใช้เส้นโค้ง (curves) ต่างๆในการแทนพอนต์ นอกจากนี้ ในการออกแบบและสร้างเวกเตอร์พอนต์เพื่อใช้งาน ผู้สร้างมักจะต้องมีความรู้ทางคณิตศาสตร์ เพื่อให้สามารถระบุรูปร่างของพอนต์ได้อย่างถูกต้องและสวยงาม ซึ่งทำให้การออกแบบพอนต์ใหม่ๆ เป็นไปอย่างลำบากและล่าช้า ในที่นี้จะใช้แนวทางการออกแบบพอนต์จากตารางแบบจุดภาพ เช่นเดียวกับการสร้างพอนต์จุดภาพ ซึ่งผู้สร้างพอนต์ไม่จำเป็นต้องมีความรู้ทางคณิตศาสตร์มากนัก ดังนั้นในงานวิจัยนี้จึงต้องมีการศึกษาวิธีการในการวิเคราะห์จุดภาพที่ปรากฏ เพื่อนำมาแทนด้วยรูปแบบของเวกเตอร์พอนต์โดยอัตโนมัติ

ทฤษฎีที่เกี่ยวข้องกับงานวิจัยชิ้นนี้ มีดังต่อไปนี้

1.1.1 ลักษณะการสร้างภาพโดยทั่วไป ศิลปะเกี่ยวกับการวาดภาพจะเน้นถึงวิธีการนำเส้นและสีต่างๆ มาประกอบกันเป็นรูป ลักษณะของเส้นที่ใช้อยู่ อาจแบ่งได้เป็นสองประเภทใหญ่ๆ ทั้งนี้โดยไม่คำนึงถึงความเข้มหรือการต่อเนื่องและขนาดตอนของเส้นที่ใช้ คือ เส้นตรง (straight lines) และเส้นโค้ง (Curved lines) (สัญธาน ชีรนรวิชย์ 2526) ซึ่งในการสร้างพอนต์ เราอาจพิจารณาได้ว่าพอนต์แต่ละตัวก็คือภาพนั่นเอง โดยในที่นี้เส้นที่ใช้ในการสร้างภาพจะต้องอยู่ในรูปแบบของเส้นบาง เพื่อความสะดวกในการวิเคราะห์ ซึ่งในกรณีที่เป็นภาพที่มีความหนา เราจะทำการวิเคราะห์ให้อยู่ในแบบเส้นขอบ (outline) ก่อนที่จะนำมาแทนเป็นเวกเตอร์พอนต์ต่อไป

1.1.2 เวกเตอร์พอนต์ ในการสร้างเวกเตอร์พอนต์ เราอาจจำแนกการแทนที่พอนต์แต่ละตัวในแบบเวกเตอร์ได้เป็นสามวิธี คือ

1.1.2.1 การแทนที่พอนต์โดยประกอบส่วนต่างๆ ขึ้นจากสมการทางคณิตศาสตร์ เช่น ประกอบขึ้นจาก ฟังก์ชัน  $\sin$ ,  $\cos$ , วงกลม เส้นตรง ตลอดจนสมการของรูปต่างๆ ซึ่งมีข้อเสียคือ ในการออกแบบพอนต์แต่ละตัวมีความยุ่งยากเป็นอย่างมาก เพราะผู้ออกแบบต้องแยกพอนต์แต่ละตัวเป็นส่วนๆ และหาสมการที่เหมาะสมเพื่อนำมาแทนที่พอนต์นั้นๆ

1.1.2.2 การแทนที่พอนต์ด้วยเส้นตรงย่อยๆ คือการนำเส้นตรงเล็ก ๆ จำนวนหนึ่ง ซึ่งมีขนาดและมุมต่างๆกันมาประกอบขึ้นเป็นพอนต์แต่ละตัว เช่น การสร้างสโตรคพอนต์ของเทอร์โบซี ซึ่งจะสร้างตัวอักษรแต่ละตัวด้วยการวาดชุดของเส้นตรง (line sequence) (BEN EZZELL 1989) ซึ่งวิธีนี้มีข้อเสียคือ การแทนที่ส่วนที่มีความโค้งมน จะปรากฏเป็นเหลี่ยมมุมได้เมื่อมีการขยายพอนต์ให้ใหญ่ขึ้น

1.1.2.3 การแทนที่พอนต์ด้วยเส้นโค้ง (curve) เป็นวิธีที่นิยมมาใช้กันอยู่ในปัจจุบัน เนื่องจากมีข้อดีคือสามารถแทนพอนต์ในแบบต่างๆ ได้เหมือนจริง อีกทั้งผู้ออกแบบพอนต์ไม่จำเป็นต้องมีความรู้ทางคณิตศาสตร์ในการสร้างพอนต์ต่างๆ ขึ้นมา ส่วนข้อเสียคือ การเสียเวลาในการจัดเส้นโค้ง (curve fitting)

1.1.3 เส้นโค้งเบซิเยร์(Bezier curves) เป็นเส้นโค้งแบบพาราเมตริก (parametric)แบบหนึ่งซึ่งสามารถกำหนดขึ้นมาโดยใช้จุดจำนวนหนึ่ง ทำให้สามารถเก็บข้อกำหนดของรูปร่างตัวอักษร(character shape) ได้อย่างกะทัดรัดและมีประสิทธิภาพ ซึ่งเส้นโค้งชนิดนี้ได้ถูกสร้างขึ้นโดยนักคณิตศาสตร์ชาวฝรั่งเศสในทศวรรษที่ 1970 เพื่อใช้ในการอธิบายรูปร่างแบบสามมิติของพื้นผิวรถ ส่วนในแบบของสองมิติ เส้นโค้งเบซิเยร์ได้ถูกใช้โดยผู้ผลิตอุปกรณ์เรียงพิมพ์แบบดิจิทัลเมื่อไม่นานมานี้ โดยนำมาแทนที่การใช้เส้นตรง วงกลมและส่วนโค้ง(arc) ในการแทนรูปร่างของตัวเรียงพิมพ์ เพราะคุณสมบัติของความเข้าใจง่ายและสามารถคำนวณได้อย่างมีประสิทธิภาพ โดยที่ Adobe เป็นผู้ผลิตรายแรกที่นำเส้นโค้งเบซิเยร์นี้มาใช้ในภาษาอธิบายหน้ากระดาษแบบโพสต์สคริปต์ (Postscript page description language) (Todd King 1990) และหลังจากนั้น ผู้ผลิตรายอื่นๆ ก็ได้ยึดแนวทางนี้เช่นกัน

1.1.4 เส้นโค้งสไปไลน์ (Spline curves) เป็นเส้นโค้งอีกแบบหนึ่ง ซึ่งมีคุณสมบัติที่แตกต่างจากเส้นโค้งเบซิเยร์ คือ การแก้ไขส่วนใดส่วนหนึ่งของเส้นโค้ง จะไม่มีผลกระทบต่อส่วนอื่นของเส้นโค้งนั้น และเส้นโค้งนี้ยังสามารถจำแนกเป็นเส้นโค้งในแบบต่างๆ ได้อีก เช่น คิวบิกสไปไลน์(Cubic splines), ไบคิวบิกสไปไลน์(Bicubic splines), บี-สไปไลน์(B-splines) เป็นต้น ซึ่งเส้นโค้งเหล่านี้สามารถนำมาใช้ในการอธิบายรูปร่าง(shape)ได้ ตัวอย่างเช่น ได้มีการนำเส้นโค้งบี-สไปไลน์ มาออกแบบพอนต์ของตัวอักษรสำหรับการเรียงพิมพ์ ซึ่งควบคุมโดยคอมพิวเตอร์(Theo Pavlidis 1982)

## 1.2 วัตถุประสงค์ของการวิจัย

1.2.1 เพื่อพัฒนาขั้นตอนวิธี (algorithm) ในการแปลงรูปจำกัคใดๆ ให้อยู่ในแบบของ เส้นขอบ (outline)

1.2.2 เพื่อพัฒนาขั้นตอนวิธีในการวิเคราะห์จุดภาพ (bit image) เพื่อแปลงให้อยู่ในแบบ ของเวกเตอร์พอนต์

1.2.3 เพื่อศึกษาและพัฒนาโปรแกรมในการออกแบบและใช้งานเวกเตอร์ฟอนต์ ซึ่งมีความสะดวกต่อผู้สร้างฟอนต์ โดยที่ผู้สร้างฟอนต์ไม่จำเป็นต้องมีความรู้ทางคณิตศาสตร์ขั้นสูง

1.2.4 เพื่อพัฒนาขั้นตอนวิธีและโปรแกรมในการเปลี่ยนแปลงขนาด ตำแหน่ง และการหมุน ฟอนต์ที่สร้างขึ้น

### 1.3 ขั้นตอนการวิจัย

1.3.1 ศึกษาและพัฒนาวิธีในการแปลงรูปสัญลักษณ์ของฟอนต์ให้อยู่ในแบบของเส้นขอบ (outline)

1.3.2 ศึกษาและคัดเลือกวิธีการทางคณิตศาสตร์ที่เหมาะสม ในการนำมาแทนรูปสัญลักษณ์เพื่อ ให้อยู่ในแบบของเวกเตอร์ฟอนต์

1.3.3 พัฒนารูปแบบในการวิเคราะห์จุดภาพ (bit image) ด้วยคอมพิวเตอร์ เพื่อแทนรูปสัญลักษณ์ ด้วยสมการทางคณิตศาสตร์ที่กำหนดไว้โดยฮัตโนมิต

1.3.4 ออกแบบขั้นตอนวิธีในการสร้าง จัดเก็บ และนำเวกเตอร์ฟอนต์ไปใช้งาน

1.3.5 พัฒนาโปรแกรมในการสร้างเวกเตอร์ฟอนต์ตามขั้นตอนที่ศึกษาในข้อ 1.3.4

1.3.6 พัฒนาโมดูลพื้นฐานในการประยุกต์ใช้เวกเตอร์ฟอนต์ที่สร้างขึ้น ซึ่งโปรแกรมประยุกต์ สามารถนำไปเรียกใช้งานได้ที่ทันที

### 1.4 ขอบเขตการวิจัย

1.4.1 พัฒนาขึ้นด้วยภาษาซีบนไมโครคอมพิวเตอร์ IBM PC

1.4.2 การแสดงผลจะเป็นการแสดงผลออกทางหน่วยจอกกราฟิกของไมโครคอมพิวเตอร์ คือ Hercules, EGA และ VGA

1.4.3 การสร้างฟอนต์จะใช้แนวทางการสร้างตารางจุดภาพ เช่นเดียวกับการสร้างฟอนต์จุดภาพ

1.4.4 ฟอนต์ตัวอักษรและรูปสัญลักษณ์พิเศษต่างๆ ที่สร้างขึ้น ในงานวิจัยนี้ต้องเป็นรูปจำกัด เท่านั้น นั่นคือ จะประกอบขึ้นจากเส้นตรงและเส้นโค้งในแบบ 2 มิติ โดยไม่นับ รวมถึงการระบายสี หรือการไล่ระดับสีเทา

1.4.5 การแสดงผลฟอนต์แต่ละตัว จะสามารถเลือกแสดงสีได้หนึ่งสีต่อหนึ่งตัวเท่านั้น

1.4.6 ในการวิจัยและพัฒนาโปรแกรมจะไม่เน้นถึงความเร็วในการแสดงผลฟอนต์

## 1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 ช่วยให้ผู้สามารถแสดงรูปสัญลักษณ์ ตลอดจนตัวอักษรภาษาต่างๆ เช่น ภาษาไทย ออกทาง หน่วยจอภาพได้อย่างสวยงามและมีประสิทธิภาพ

1.5.2 เป็นแนวทางในการพัฒนาการสร้างเวกเตอร์ฟอนต์สำหรับอุปกรณ์แสดงผลอื่นๆ

1.5.3 ส่งเสริมให้มีการพัฒนาโปรแกรมประยุกต์ ซึ่งใช้รูปและภาษาไทยในการแสดงผลและติดต่อกับผู้ใช้งาน

1.5.4 เป็นแนวทางในการศึกษาการแทนตัวอักษรและรูปซึ่งซับซ้อน เช่น รูปซึ่งมีการระบายสี หรือการไล่ระดับสีเทา ตลอดจนรูปสามมิติ ให้อยู่ในแบบเวกเตอร์

1.5.5 เป็นเครื่องมือ (tool) ในการสร้างภาพสำหรับอุปกรณ์ช่วยสอน (Computer Assisted Instruction) โดยการแทนภาพหรือส่วนของภาพด้วยรูปสัญลักษณ์ของเวกเตอร์ฟอนต์ ซึ่งจะสามารถ เคลื่อนย้าย เปลี่ยนแปลงขนาด และหมุนภาพได้โดยสะดวก



ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย