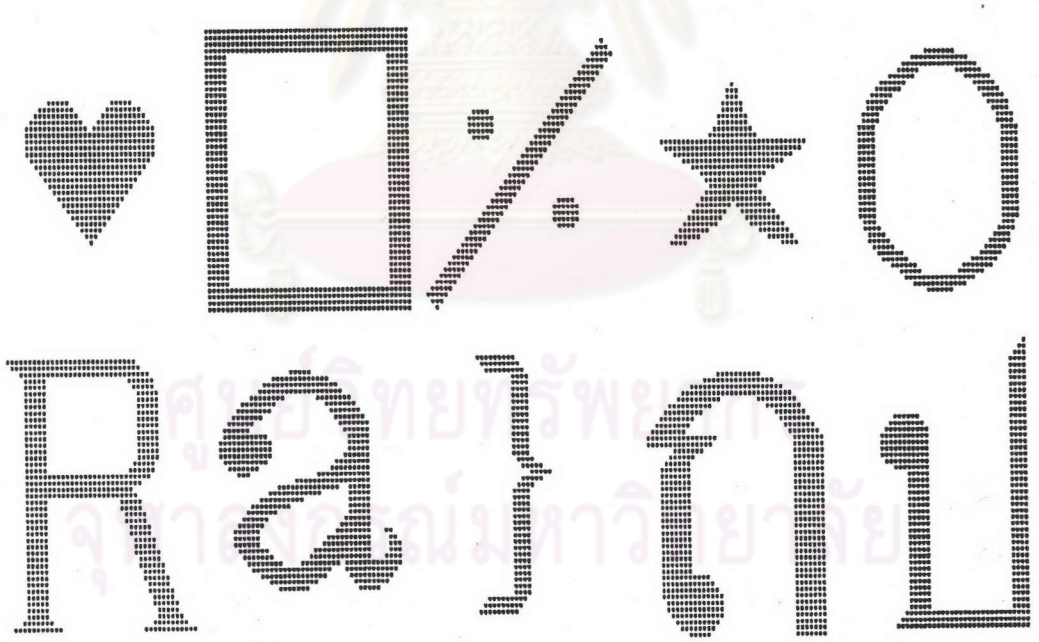
บทที่ 4

สรุปผลการวิจัยและข้อเสนอแนะ

4.1 สรุปผลการวิจัย

การวิจัยในหัวข้อเรื่อง การพัฒนาการแทนรูปสัญลักษณ์ของฟอนต์ด้วยวิธีเวกเตอร์บนหน่วยจอกราฟฟิกนี้ ได้ทำการสร้างฟอนต์แบบจุดภาพเพื่อเป็นตัวอย่างในงานวิจัยจำนวน 10 ฟอนต์ ซึ่งจะประกอบด้วยฟอนต์รูปภาพ ฟอนต์ตัวอักษรทั้งภาษาอังกฤษและภาษาไทย เพื่อเปรียบเทียบผลการแปลงฟอนต์ทั้งในแง่ความเร็วและความถูกต้องสวยงาม ซึ่งผู้สนใจจะสามารถนำไปสร้างฟอนต์ในรูปแบบที่ต้องการเพิ่มเติมได้ต่อไป

ฟอนต์ตัวอย่างที่นำมาใช้ในงานวิจัยนี้ แสดงได้ดังรูปที่ 4.1



รูปที่ 4.1 อักขระที่ใช้ในงานวิจัย

4.1.1 ผลการแปลงรูปจำกัดใด ๆ ให้อยู่ในรูปแบบของเส้นขอบ (outline)

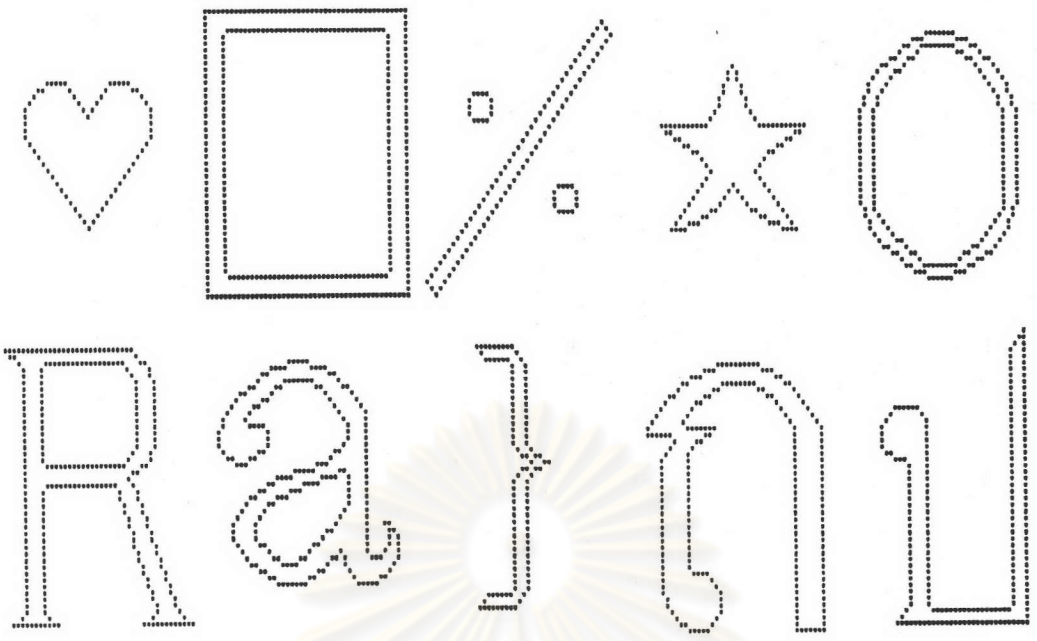
ในงานวิจัยนี้ เราสามารถแปลงรูปจำกัดซึ่งมีส่วนทึบ ที่สร้างขึ้นเป็นฟอนต์แบบจุดภาพ ให้เหลือเพียงเส้นขอบได้โดยไม่มีข้อผิดพลาดใด ๆ ยกเว้นในกรณีมุมฉาก เทคนิควิธีที่ใช้อาจจะทำให้เปลี่ยนเป็นมุมมนขึ้น แต่เนื่องจากงานวิจัยนี้เน้นในรูปแบบของเส้นโค้ง จึงไม่เป็นการยากลำบากในการแสดงมุมมนนั้นให้สวยงามเมื่ออยู่ในรูปแบบเวกเตอร์ฟอนต์ และในกรณีที่เราต้องการให้มุมนั้นยังคงเป็นมุมฉาก เราก็สามารถปรับเปลี่ยนรูปได้ในขั้นตอนหลัง

อนึ่ง ในส่วนของความเร็วในการแปลงฟอนต์แบบจุดภาพให้เหลือเพียงเส้นขอบ จากฟอนต์ตัวอย่างทั้ง 10 ฟอนต์ สามารถแปลงได้ภายในเวลาน้อยกว่า 1 วินาทีทั้งสิ้น จึงสรุปได้ว่า ความเร็วในการแปลงอยู่ในเกณฑ์ที่ได้รับผลลัพธ์ทันที ไม่ว่าจะเป็นรูปที่ซับซ้อนเท่าใดก็ตาม ตารางที่ 4.1 จะแสดงถึงเวลาที่ใช้ในการแปลงฟอนต์แบบจุดภาพให้อยู่ในรูปแบบของฟอนต์แบบเส้นขอบ และรูปที่ 4.2 จะแสดงอักขระที่ใช้ในงานวิจัยเมื่อผ่านขั้นตอนการแปลงให้อยู่ในรูปแบบเส้นขอบแล้ว

ฟอนต์	เวลาที่ใช้ในการแปลง (วินาที)
หัวใจ	0.160
สี่เหลี่ยม	0.170
เปอร์เซนต์	0.110
ดาว	0.110
อักษร O	0.110
อักษร R	0.110
อักษร a	0.160
ปีกกา	0.110
อักษร ถ	0.110
อักษร ป	0.110

ตารางที่ 4.1 แสดงเวลาที่ใช้ในการแปลงฟอนต์ตัวอย่างแบบจุดภาพเป็นฟอนต์แบบจุดภาพเส้นขอบ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.2 แสดงอักขระที่ใช้ในงานวิจัยเมื่ออยู่ในรูปแบบเส้นขอบ (outline)

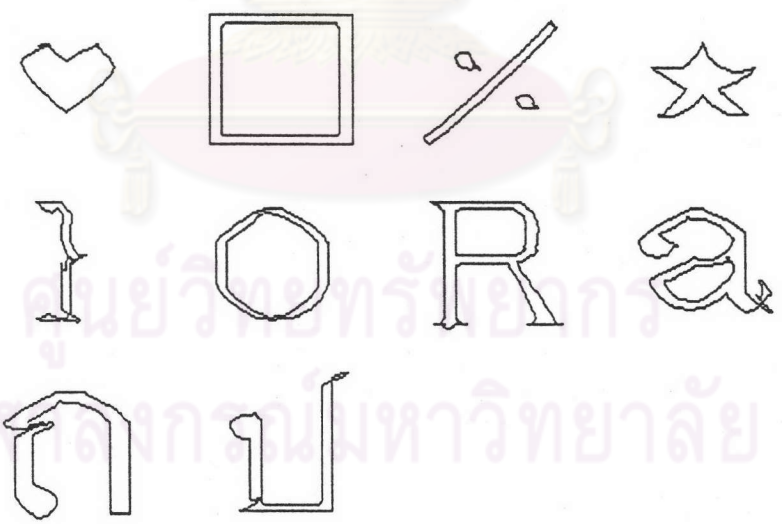
4.1.2 <u>ผลการแปลงฟอนต์แบบจุดภาพเส้นขอบให้เป็นเวกเตอร์ฟอนต์</u>

ในงานวิจัยครั้งนี้ เราสามารถวิเคราะห์ฟอนต์แบบจุดภาพเส้นขอบ ให้อยู่ในรูปแบบของเวกเตอร์ฟอนต์ได้สำเร็จ ทั้ง 10 รูปแบบฟอนต์ที่นำมาทดสอบ อันแสดงให้เห็นว่าโปรแกรมสามารถวิเคราะห์ได้ทั้งเส้นตรงและเส้นโค้ง ตลอดจนการประกอบกันของเส้นพื้นฐานต่าง ๆ ขึ้นเป็นตัวอักษร ตารางที่ 4.2 จะแสดงถึงเวลาที่ใช้ในการแปลงฟอนต์ทั้ง 10 ตัว

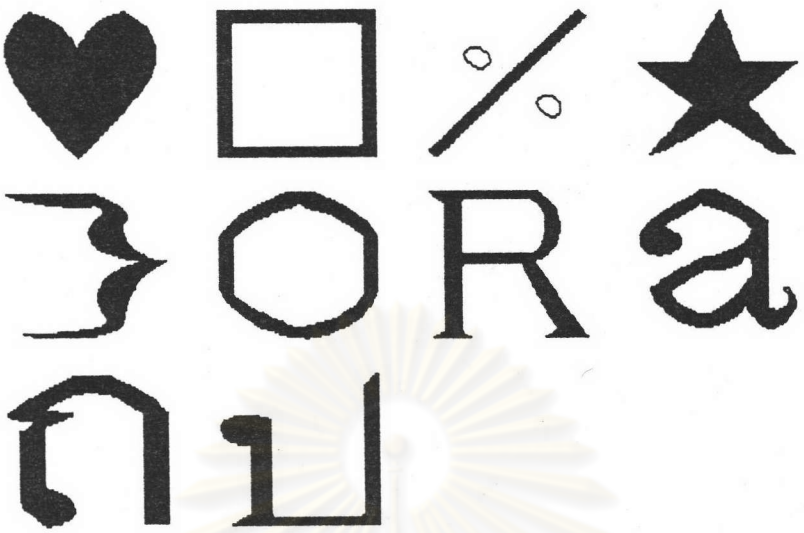
อย่างไรก็ดี การแปลงฟอนต์ในงานวิจัยครั้งนี้ ก็ยังไม่อาจทำการแปลงฟอนต์ได้สมบูรณ์ตามต้นแบบ 100 เปอร์เซนต์ โดยที่ ในการแปลงจะทำได้ถูกต้องประมาณ 90 เปอร์เซนต์ ซึ่งเราจะเห็นได้ชัดเมื่อทำการขยายเวกเตอร์ฟอนต์ให้มีขนาดใหญ่กว่าขนาดต้นแบบ เนื่องมาจากขนาดของฟอนต์แบบจุดภาพที่เป็นต้นแบบมีขนาดเพียง 50x50 เท่านั้น ในขณะที่หลังจากโปรแกรมทำการแปลงเป็นเวกเตอร์ฟอนต์แล้ว ได้ทำการเปลี่ยนขนาดพิกัดให้อยู่ในขนาด 1000x1000 ดังนั้น ความผิดเพี้ยนจึงมากขึ้น ซึ่งเราสามารถแก้ไขได้โดยการใช้ฟอนต์แบบจุดภาพที่นำมาเป็นต้นแบบที่มีขนาดใหญ่ขึ้น โดยอาจมีขนาดถึง 1000x1000 ก็จะเป็นขนาดที่ดีที่สุด อันจะทำให้ถึงแม้ว่ายังมีความผิดเพี้ยนที่ต้องแก้ไขโดยผู้สร้างฟอนต์ แต่ก็จะมีการแก้ไขน้อยที่สุด รูปที่ 4.3 จะแสดงให้เห็นถึงเวกเตอร์ฟอนต์ที่ได้จากการแปลงฟอนต์แบบจุดภาพเส้นขอบโดยโปรแกรม และรูปที่ 4.4 แสดงให้เห็นถึงเวกเตอร์ฟอนต์หลังจากที่ผู้สร้างฟอนต์ได้แก้ไขแล้ว

ฟอนต์	เวลาที่ใช้ในการแปลง (วินาที)
หัวใจ	54.37
สี่เหลี่ยม	0.11
เปอร์เซนต์	111.33
ดาว	62.17
อักษร O	161.04
อักษร R	80.47
อักษร a	261.50
ปีกกา	130.67
อักษร ถ	90.03
อักษร ป	12.20

ตารางที่ 4.2 แสดงเวลาที่ใช้ในการแปลงฟอนต์ตัวอย่างแบบจุดภาพเส้นขอบเป็นฟอนต์แบบเวกเตอร์



รูปที่ 4.3 แสดงฟอนต์ตัวอย่างเมื่อถูกแปลงให้อยู่ในรูปแบบเวกเตอร์โดยโปรแกรม



รูปที่ 4.4 แสดงเวกเตอร์ฟอนต์ตัวอย่างเมื่อได้รับการแก้ไขโดยผู้สร้างฟอนต์แล้ว

4.1.3 ผลการสร้างและใช้งานเวกเตอร์ฟอนต์

จากการที่ได้ทดลองทำการสร้างฟอนต์ตัวอย่างในงานวิจัยนี้ ได้พบว่าการสร้างโดยตารางจุดภาพ แม้ว่าจะต้องใช้เวลาในการสร้างพอสมควร แต่ก็เป็นวิธีที่สามารถทำได้ไม่ยากนัก และผู้สร้างไม่จำเป็นต้องมีความรู้ทางคณิตศาสตร์หรือคอมพิวเตอร์ ทำให้สะดวกต่อผู้ใช้ขั้นสุดท้าย (end user) ในการที่จะสร้างฟอนต์รูปภาพของตนเองเพื่อใช้งานเป็นอย่างมาก อีกทั้งเป็นการลดภาระของผู้พัฒนาโปรแกรมในการสร้างฟอนต์ โดยสามารถกำหนดให้บุคคลใด ๆ ซึ่งอาจไม่มีความรู้ทางคอมพิวเตอร์เป็นผู้สร้างฟอนต์แบบจุดภาพเข้าไปในคอมพิวเตอร์ จากนั้นจึงใช้โปรแกรมแปลงเป็นเวกเตอร์ฟอนต์และนำมาใช้งานได้ทันที

นอกจากนี้ เวกเตอร์ฟอนต์ที่ได้จากงานวิจัยนี้ สามารถทำการหมุน ย้ายตำแหน่ง ตลอดจนขยายหรือลดขนาดได้อย่างอิสระ ตลอดจนสามารถรักษาความสวยงามแม้ฟอนต์นั้นจะประกอบด้วยเส้นโค้งจำนวนมากก็ตาม ซึ่งทำให้สามารถนำไปประยุกต์ใช้ในงานต่าง ๆ อย่างมีประสิทธิผล

อนึ่ง ในงานวิจัยนี้ การแปลงรูปของเวกเตอร์ฟอนต์ ไม่ว่าจะเป็นการหมุนหรือเสื่อนเวกเตอร์ฟอนต์ก็ตาม เวลาที่ใช้ในการแสดงผลเวกเตอร์ฟอนต์จะใช้เวลาใกล้เคียงกันเสมอ เนื่องจากก่อนที่จะแสดงผลเวกเตอร์ฟอนต์จะมีการคำนวณจุดควบคุมใหม่ทุกครั้ง เพื่อความสะดวกในการเรียกใช้งาน ซึ่งในส่วนนี้ผู้ที่ต้องการประสิทธิภาพความเร็วในการแสดงผล อาจเพิ่มเติมในส่วนของการแสดงผล โดยทำการคำนวณใหม่เฉพาะในส่วนที่จำเป็น

ฟอนต์	เวลาที่ใช้ในการแสดงผล (วินาที)			
	รูปปกติ	หมุน 45 องศา	เสื่อน(60, 10)	ขยาย 2 เท่า
หัวใจ	0.330	0.330	0.320	0.330
สี่เหลี่ยม	0.610	0.660	0.610	0.600
เปอร์เซนต์	0.660	0.760	0.710	0.660
ดาว	0.550	0.550	0.550	0.610
อักษร O	0.550	0.610	0.550	0.600
อักษร R	0.600	0.710	0.660	0.600
อักษร a	1.040	1.160	1.040	1.100
ปีกกา	1.270	1.370	1.270	1.320
อักษร ถ	0.870	0.930	0.880	0.880
อักษร ป	0.660	0.720	0.610	0.660

ตารางที่ 4.3 แสดงเวลาที่ใช้ในการแสดงผลเวกเตอร์ฟอนต์ในรูปแบบต่าง ๆ

ตารางที่ 4.3 จะแสดงถึงเวลาที่ใช้ในการแสดงผลเวกเตอร์ฟอนต์ตัวอย่างทั้ง 10 ฟอนต์ ซึ่งจะเห็นได้ว่าความเร็วในการวาดฟอนต์จะขึ้นกับความซับซ้อนของรูปที่ทำการวาด โดยเฉพาะรูปที่มีเส้นโค้งมาก ก็จะยิ่งใช้เวลามากขึ้น ในขณะที่การแปลงรูป (transform) มีผลต่อเวลาในการแสดงผลน้อยมาก ยิ่งไปกว่านั้นการแปลงรูปอาจมีผลให้การแสดงผลฟอนต์ช้าลงหรือเร็วขึ้นก็ได้

4.2 ข้อจำกัดของการวิจัย

ในงานวิจัยนี้ได้มุ่งถึงความสะดวกในการตรวจสอบ และแก้ไขผลการวิจัย ตลอดจนความสะดวกของผู้สนใจในการศึกษาต่อไปเป็นสำคัญ จึงไม่ได้เน้นถึงเทคนิควิธีในการโปรแกรมให้มีประสิทธิภาพสูงสุด ซึ่งสามารถสรุปข้อจำกัดที่สามารถปรับปรุงได้ต่อไป คือ

ก. ในส่วนของขนาดข้อมูล จะเห็นได้จากรูปแบบการจัดเก็บแฟ้มข้อมูลซึ่งจัดเก็บเป็นรหัสแอสกีในรูปตัวอักษร ซึ่งสามารถตรวจแก้ได้โดยโปรแกรมบรรณาธิกรณ์ (editor) ทั่วไป ในการนำไปใช้งานจริง สำหรับแฟ้มข้อมูลเวกเตอร์ฟอนต์ เราสามารถแก้ไขให้อยู่ในรูปแบบตัวเลขจำนวนจริง (floating point) และจัดเก็บในแฟ้มฐานสอง (binary file) เพื่อเพิ่มประสิทธิภาพในด้านความเร็วและลดขนาดของแฟ้มเวกเตอร์ฟอนต์

ข. ในส่วนของความเร็ว เนื่องจากในงานวิจัยนี้ ได้พยายามรักษาโปรแกรมให้คล้ายคลึงกับสูตรต่าง ๆ ตามทฤษฎี เพื่อให้สะดวกต่อการศึกษาค้นคว้าและอ้างอิงต่อไป จึงไม่ได้ทำการแปลงให้มีประสิทธิภาพสูงสุด (optimize) ดังนั้น หากต้องการความเร็วในการแสดงผลเวกเตอร์ฟอนต์เพิ่มขึ้น เราสามารถทำการแก้ไขตัวโปรแกรมเพิ่มเติมได้ นอกจากนี้ ยังอาจใช้ภาษาแอสแซมบลีเข้าช่วย เพื่อให้เกิดประสิทธิภาพสูงสุดในส่วนที่มีการคำนวณจำนวนมากด้วย

ค. ในส่วนของการสร้างฟอนต์แบบจุดภาพ ได้จำกัดขนาดไว้ที่ 50x50 เนื่องมาจากข้อจำกัดในเรื่องความละเอียดของจอภาพ (640x350) แต่ในส่วนของเทคนิควิธีและตัวโปรแกรมไม่ได้มีข้อจำกัดของขนาดในการแปลง ดังนั้นเราสามารถเพิ่มเติมในส่วนของการเลื่อนจอภาพ (scroll) เพื่อให้สามารถสร้างฟอนต์แบบจุดภาพที่มีขนาดใหญ่กว่านี้ได้ นอกจากนี้ ยังอาจเพิ่มเติมในส่วนของการรับรูปภาพจากเครื่องอ่านภาพ (scanner) และทำการแปลงให้อยู่ในรูปแบบที่โปรแกรมสามารถนำมาใช้งานได้ อันจะทำให้สะดวกและรวดเร็วต่อการใช้งาน

ง. ในส่วนของการระบายสีฟอนต์ ได้ใช้คำสั่งในการระบายพื้น (flood fill) ของเทอร์โบซี ซึ่งมีข้อจำกัดที่จะหยุดการระบายเมื่อไม่พบเส้นทางระบายต่อ ทำให้อาจเกิดการระบายไม่ทั่วฟอนต์ที่ต้องการเมื่อฟอนต์มีขนาดเล็ก หรือเมื่อมีรูปอยู่ในตำแหน่งเดียวกันก่อนแล้ว ปัญหานี้อาจแก้ได้โดยทำการศึกษาเทคนิควิธีในการระบายพื้นที่เอง อย่างไรก็ดีในการระบายสีพื้นก็จะทำให้เกิดข้อจำกัดว่าฟอนต์รูปภาพที่ใช้งานจะต้องเป็นพื้นที่ปิดเท่านั้น

4.3 ข้อเสนอแนะในการวิจัยขั้นต่อไป

4.3.1 ปรับปรุงขั้นตอนในการตรวจสอบความถูกต้องของจุดควบคุมที่หาได้ ตลอดจนขั้นตอนในการตัดแบ่งเส้นโค้งย่อยให้เล็กลงในกรณีที่ไม่สามารถหาจุดควบคุมได้

4.3.2 ทำการวิจัยการใช้เส้นโค้งแบบเบซิเยร์ ดีกรี 3 หรือการใช้เส้นโค้งแบบบี-สไปลน์ (B-Spline) ตลอดจนเส้นโค้งแบบอื่น ๆ เพื่อศึกษาข้อดีข้อเสียในการแทนที่เวกเตอร์ฟอนต์ด้วยเส้นโค้งแบบต่าง ๆ

4.3.3 ศึกษาวิธีในการแปลงภาพที่ได้จากเครื่องอ่านภาพหรืออุปกรณ์ในการนำเข้าข้อมูลอื่น ๆ ให้อยู่ในรูปแบบเวกเตอร์

4.3.4 ปรับปรุงประสิทธิภาพในการแปลงฟอนต์แบบจุดภาพ จากรูปจำกัดให้สามารถวิเคราะห์ รูปซึ่งประกอบด้วยสีต่าง ๆ หรือการไล่ระดับสีเทา รูปซึ่งมีความทึบ ตลอดจนรูป 3 มิติ

4.3.5 ใช้ฟอนต์แบบเวกเตอร์ที่สร้างขึ้น นำไปประยุกต์ในงานวิจัยการสร้างฟอนต์ภาษาไทย ตลอดจนภาษาอื่น ๆ ซึ่งอาจเพิ่มเติมในส่วนของการจัดระดับอักษร การคำนวณเพื่อสร้างรูปแบบตัวอักษรต่าง ๆ เช่น ตัวหนา ตัวเอียง เป็นต้น

4.3.6 ทำการวิจัยในการสร้างภาพเคลื่อนไหว ตลอดจนทำการสร้างโปรแกรมคอมพิวเตอร์ช่วยการสอน (CAI) โดยใช้วิธีเวกเตอร์ต่อไป

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย