

ระบบจินตทัศน์อัลกอริธึม ภายใต้สภาพปฏิบัติการไมโครซอฟต์วินโดวส์ *

นายสมชาย ประสิทธิ์จูตระกูล
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

นายวิทยา วัชรวิทยากุล
สาขาคอมพิวเตอร์
คณะสถิติประยุกต์
สถาบันพัฒนาบริหารศาสตร์

An Algorithm Visualization System under Microsoft Windows Operating Environment

Somchai Prasitjutrakul
Dept. of Computer Engineering
Faculty of Engineering
Chulalongkorn University

Wittaya Watcharawittaykul
Division of Computer Science
School of Applied Statistics
National Institute of Development Administration

บทคัดย่อ

การจินตทัศน์อัลกอริธึมเป็นกรรมวิธีหนึ่งในการศึกษาทำความเข้าใจการทำงานของอัลกอริธึม ด้วยการ
ใช้ภาพ และการเปลี่ยนแปลงของภาพ เป็นสื่อในการแสดงถึงขั้นตอนการทำงานของอัลกอริธึม บทความนี้
นำเสนอภาพรวมของโครงสร้างหลักของระบบจินตทัศน์อัลกอริธึมที่ทำงานภายใต้สภาพปฏิบัติการไมโครซอฟต์วิน
โดวส์ ระบบจินตทัศน์นี้อาศัยคุณสมบัติที่สำคัญของวินโดวส์หลายประการมาใช้ เพื่อให้การออกแบบเป็นไปอย่าง
มีโครงสร้าง เป็นระบบ และใช้งานง่าย อันได้แก่การใช้ระบบการประสานกับผู้ใช้งานแบบกราฟิกส์ การจินตทัศน์
อัลกอริธึมแบบหลายภารกิจ การแสดงมุมมองการนำเสนอการจินตทัศน์แบบหลายมุมมองโดยใช้การประสาน
แบบหลายเอกสาร และการใช้คลังโปรแกรมเชื่อมโยงแบบพลวัตเพื่อเก็บฟังก์ชันต่างๆของระบบ

Abstract

Algorithm Visualization is a means to study the behavior of how algorithms work by
using graphical views and animations of an algorithm in action. This paper presents the
overview of the main structure of an algorithm visualization system running under Microsoft
Windows operating environment. The system uses many features provided by Windows so
that the design will be structured, systematic, and easy to use. These features are the use of
graphical user interface (GUI) for the ease of use, multitasking for multiple algorithm
visualizations, multiple document interface (MDI) for multiple views of presentations, and
dynamic link library (DLL) for storing all the system functions.

* งานวิจัยนี้ได้รับทุนอุดหนุนการวิจัยจากศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ

1. บทนำ

อัลกอริทึม (algorithm) ซึ่งถูกบัญญัติในศัพท์คอมพิวเตอร์ฉบับราชบัณฑิตยสถาน ปีพ.ศ. 2535 ว่า ขั้นตอนวิธี นั้นหมายความว่าขั้นตอนวิธีในการแก้ไขปัญหา ในปัจจุบันจะเป็นที่เข้าใจโดยทั่วไปว่าอัลกอริทึมคือขั้นตอนวิธีในการแก้ไขปัญหาโดยใช้เครื่องคอมพิวเตอร์ ตัวอย่างเช่น อัลกอริทึมในการหาค่าหาร่วมมากที่สุด (greatest common divisor) ของเลขจำนวนเต็มสองจำนวนของยูคลิด (Euclid) อัลกอริทึมในการหารากที่สองของค่า x ที่เป็นเลขบวกใดๆ โดยใช้วิธีของนิวตัน-ราฟสัน (Newton-Raphson) หรืออัลกอริทึมในการเรียงลำดับข้อมูลแบบผสาน เป็นต้น โดยทั่วไปแล้ว ปัญหาหนึ่งๆย่อมต้องมีขั้นตอนวิธีในการแก้ปัญหานั้นได้หลายวิธี การตัดสินใจเลือกใช้ อัลกอริทึมใดแก้ปัญหานั้น คงต้องขึ้นกับประสิทธิภาพในการทำงานของอัลกอริทึม ลักษณะของข้อมูล พร้อมทั้งสภาวะและเงื่อนไขต่างๆของปัญหาที่สนใจ ดังนั้นผู้ที่ทำการเลือกใช้ขั้นตอนวิธีในการแก้ปัญหานั้น จำเป็นอย่างยิ่งที่จะต้องมีความเข้าใจลักษณะและพฤติกรรมการทำงานของอัลกอริทึมต่างๆเหล่านั้น

การจินตทัศน์อัลกอริทึม (algorithm visualization) เป็นลักษณะวิธีหนึ่งในการแสดงขั้นตอนการทำงานของอัลกอริทึมด้วยภาพ พร้อมทั้งการเปลี่ยนแปลงหรือการเคลื่อนไหวของภาพ เพื่อเป็นสื่อในการแสดงถึงพฤติกรรมของอัลกอริทึมที่สนใจศึกษา โดยทั่วไปจะแสดงการทำงานของอัลกอริทึมด้วยมุมมองต่างๆ เพื่อให้ผู้ศึกษาเห็นถึงคุณสมบัติเด่นๆของการทำงาน ซึ่งในบางครั้งอาจยากต่อการเข้าใจ หรืออาจไม่เคยตระหนักถึง มุมมองการทำงานต่างๆเหล่านี้จะเปลี่ยนแปลงไปพร้อมๆกันระหว่างการทำงานของอัลกอริทึม นอกจากนี้ยังสามารถแสดงการทำงานของหลายๆอัลกอริทึมที่ใช้แก้ปัญหานั้น เพื่อการเปรียบเทียบพฤติกรรม หรือประสิทธิภาพของอัลกอริทึมเหล่านั้น โดยระบบจะทำการประสานจังหวะ (synchronization) การทำงานและการแสดงผล เพื่อให้การเปรียบเทียบการทำงานเป็นไปอย่างถูกต้อง

การประยุกต์ใช้งานระบบจินตทัศน์อัลกอริทึมที่เห็นเด่นชัดคือ การนำระบบไปใช้ประกอบการเรียนการสอนวิชาที่เกี่ยวข้องกับอัลกอริทึม ผู้สอนสามารถสาธิตระบบจินตทัศน์อัลกอริทึมประกอบคำบรรยาย อีกทั้งผู้เรียนสามารถทดลองใช้ระบบเพื่อศึกษาอัลกอริทึมด้วยตนเองในห้องปฏิบัติการ โดยสามารถทดลองเปลี่ยนแปลงมุมมอง เปลี่ยนแปลงข้อมูลนำเข้า หรือเปลี่ยนแปลงความเร็วในการแสดงผล เพื่อสังเกตพฤติกรรมการทำงานของอัลกอริทึม ประโยชน์อีกประการหนึ่งคือการนำระบบนี้มาประยุกต์ใช้เป็นเครื่องมือในการวิจัยทางด้านการวิเคราะห์และออกแบบอัลกอริทึม ผู้ใช้สามารถศึกษาผลกระทบของตัวแปรต่างๆ หรือลักษณะข้อมูลนำเข้าของอัลกอริทึม ที่มีผลต่อประสิทธิภาพในการทำงาน ซึ่งอาจนำไปสู่การปรับปรุงอัลกอริทึมให้มีประสิทธิภาพดียิ่งขึ้น นอกจากนี้ ยังมีผู้นำระบบจินตทัศน์มาใช้เป็นเครื่องมือในการผลิตภาพที่แทนโครงสร้างข้อมูล เพื่องานพิมพ์ ภาพต่างๆเหล่านี้ถูกผลิตออกมาเองโดยตรงจากตัวอัลกอริทึม ดังนั้นจึงมีความถูกต้องทั้งในแง่ของตำแหน่งและสัดส่วนของวัตถุต่างๆในภาพ ตัวอย่างเช่นรูปภาพในหนังสือของ Sedgewick [1] ทุกรูปถูกผลิตจากระบบจินตทัศน์

บทความนี้นำเสนอภาพรวมของระบบจินตทัศน์อัลกอริทึมที่คณะวิจัยกำลังดำเนินการออกแบบและพัฒนา ภายใต้สภาพปฏิบัติการไมโครซอฟต์วินโดวส์ (Microsoft Windows) โดยจะเริ่มนำเสนองานวิจัยที่เกี่ยวข้องในหัวข้อที่ 2 จากนั้นนำเสนอโครงสร้างหลักของระบบจินตทัศน์อัลกอริทึมที่คณะวิจัยเสนอในหัวข้อที่ 3 หัวข้อที่ 4 อธิบายถึงการออกแบบส่วนจำเพาะต่างๆด้วยลักษณะสมบัติที่วินโดวส์มี ให้เหมาะสม ตามด้วยบทสรุปของบทความนี้ในหัวข้อที่ Error! Bookmark not defined.

2. งานวิจัยที่เกี่ยวข้อง

ในหัวข้อนี้จะได้กล่าวถึงความเป็นมาและงานวิจัยที่เกี่ยวข้องกับการแสดงการทำงานของอัลกอริทึม ซึ่งประกอบด้วยลักษณะงานสามรูปแบบ คือการบันทึกภาพยนตร์ที่แสดงการทำงานของอัลกอริทึม การแสดงค่าของข้อมูลหรือภาพของโครงสร้างข้อมูลระหว่างการทำงาน และระบบจินตทัศน์อัลกอริทึม ซึ่งมีรายละเอียดในแต่ละรูปแบบดังนี้

2.1 ภาพยนตร์แสดงอัลกอริทึม วิธีนี้คือการบันทึกภาพยนตร์ที่แสดงลักษณะการทำงานของอัลกอริทึมที่สนใจ โดยอาจอาศัยเทคนิคการจัดทำภาพยนตร์มาประกอบการแสดงพฤติกรรมการทำงาน ตัวอย่างเช่น ภาพยนตร์จาก Bell Labs ที่แสดงถึงการทำงานของ list processing [2] ภาพยนตร์ของ Booth ที่แสดงอัลกอริทึมต่างๆบนโครงสร้างข้อมูลแบบ PQ-Trees [3] ภาพยนตร์เรื่อง *Sorting Out Sorting* ของ Baecker [4] แสดงการทำงานของการทำงานของเรียงลำดับข้อมูล

2.2 การแสดงโครงสร้างข้อมูล เป็นวิธีการแสดงการเปลี่ยนแปลงของโครงสร้างข้อมูลระหว่างการทำงาน เช่นการแสดงค่าของฟังก์ชันระหว่างการทำงานของวิธีนิวตัน-ราฟสัน ระหว่างการหาจุดต่ำสุดของฟังก์ชัน ตัวอย่างของระบบพัฒนาโปรแกรมซึ่งมีการแสดงภาพของโครงสร้างข้อมูล ระหว่างการทำงาน มีอาทิเช่น Incense [5] , GDBX [6], PROVIDE [7] เป็นต้น การแสดงค่าของข้อมูล หรือภาพของโครงสร้างข้อมูล พร้อมทั้งการเปลี่ยนค่าหรือภาพระหว่างการทำงานเหล่านี้ มีความเหมาะสมมากกับการแก้จุดบกพร่องระหว่างการพัฒนาโปรแกรม

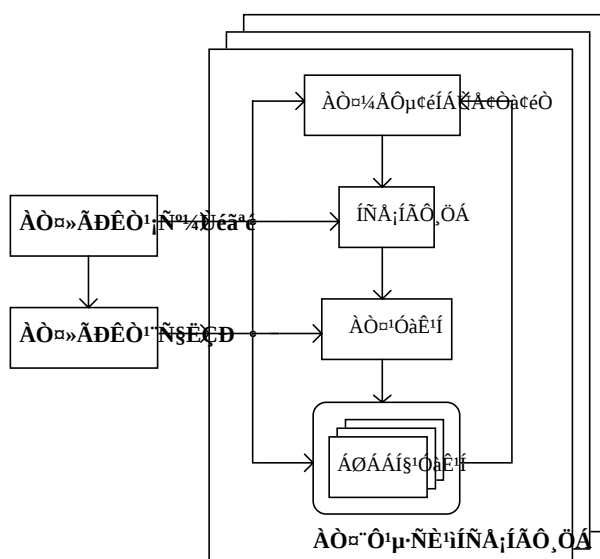
2.3 ระบบจินตทัศน์อัลกอริทึม ระบบจินตทัศน์อัลกอริทึมระบบแรกถูกพัฒนาขึ้นที่มหาวิทยาลัยโตรอนโตเมื่อปี 1975 [8] เนื่องจากราคาอุปกรณ์ทางคอมพิวเตอร์ยังสูงมาก ระบบนี้จึงถูกออกแบบมาเพื่อการบันทึกภาพยนตร์เป็นหลัก การทำงานจึงไม่ใช่เป็นการติดต่อโต้ตอบกับผู้ใช้โดยตรง ต่อมาได้มีการพัฒนาระบบจินตทัศน์อัลกอริทึมขึ้นอีกหลายระบบ ในระบบของ Yarwood [9] ผู้เขียนโปรแกรมจะจัดเตรียมแฟ้มชุดคำสั่งพิเศษแบบ JCL ที่บ่งบอกถึงการแสดงภาพซึ่งแทนข้อมูลในโปรแกรม ข้อมูลจะถูกแสดงเมื่อมีการเปลี่ยนเฉพาะในส่วนของโปรแกรมที่ได้ระบุไว้ ในระบบของ de Boer [10] นั้นมีคุณสมบัติเพิ่มเติมจากระบบของ Yarwood ตรงที่ ผู้เขียนโปรแกรมสามารถควบคุมการแสดงผลของข้อมูล โดยเขียนคำสั่งพิเศษเพิ่มเติมลงในโปรแกรม PL/I เลย สำหรับระบบของ Brown [11] ที่ชื่อว่า BALSA เป็นระบบที่รับการกล่าวถึงมาก เนื่องจากมีการติดต่อโต้ตอบกับผู้ใช้โดยตรงแบบ GUI (Graphics User Interface) โดยระบบ BALSA รุ่นที่ 2 นั้นได้ถูกพัฒนาให้ทำงานกับเครื่อง Macintosh

ในแง่ของผู้ศึกษาการทำงานของอัลกอริทึม ภาพยนตร์ที่บันทึกการทำงานของอัลกอริทึมแสดงการทำงานตามที่ผู้สร้างเตรียมไว้เท่านั้น ผู้ใช้ไม่สามารถเปลี่ยนแปลงได้ ในกรณีของระบบพัฒนาโปรแกรมที่แสดงข้อมูลขณะทำงานได้นั้น โดยทั่วไปก็ไม่สามารถสร้างกฎเกณฑ์เงื่อนไขในการแสดง อีกทั้งโดยส่วนใหญ่ไม่สามารถแสดงความสัมพันธ์ของข้อมูลและสถานะการทำงานได้ นอกจากนี้ภาพการเปลี่ยนแปลงโครงสร้างข้อมูล ก็ไม่สามารถแสดงให้เห็นถึงการทำงานของอัลกอริทึมได้อย่างเด่นชัด ดังนั้นเพื่อให้เข้าใจการทำงานของอัลกอริทึมอย่างได้ผล จะต้องใช้ระบบจินตทัศน์อัลกอริทึมที่โต้ตอบกับผู้ใช้โดยตรง ผู้สร้างการจินตทัศน์เป็นระบุดูสนใจเพื่อให้ผู้ใช้สังเกตเห็นเหตุการณ์ที่น่าสนใจเหล่านั้น โดยจุดสนใจจะต้องมีไม่มากเกินไป และไม่น้อยเกินไป นั่นคือมีเพียงพอต่อการเข้าถึงแก่นสารของการทำงานของอัลกอริทึมที่กำลังศึกษา

3. โครงสร้างหลักของระบบจินตทัศน์อัลกอริธึม

การจินตทัศน์อัลกอริธึมจะได้ผลก็ต่อเมื่อมีระบบที่เอื้ออำนวยการสร้างและการใช้ ในแง่ของผู้สร้างการจินตทัศน์ของอัลกอริธึม¹ ระบบจะต้องเอื้ออำนวยคำสั่งในการแสดงภาพ การเคลื่อนไหวภาพ การจัดการเกี่ยวกับการแบ่งมุมมองบนจอภาพ การรับข้อมูล การโต้ตอบกับผู้ใช้ กลไกการประสานจังหวะการทำงานระหว่างอัลกอริธึมและการเปลี่ยนแปลงภาพในมุมมองต่างๆ ทั้งนี้เพื่ออำนวยให้การสร้างการจินตทัศน์เป็นไปอย่างสะดวกและรวดเร็ว โดยผู้สร้างจะได้มุ่งเน้นแต่เฉพาะงานในการนำเสนอการทำงานของอัลกอริธึมว่าจะนำเสนออย่างไรจึงจะทำให้ผู้ใช้เข้าใจถึงแก่นของการทำงานของอัลกอริธึมที่กำลังศึกษา

ส่วนในแง่ของผู้ใช้นั้น ปัจจัยที่สำคัญที่สุดคือความสะดวกในการใช้งาน ระบบจินตทัศน์อัลกอริธึม จะต้องมียุทธศาสตร์การติดต่อประสานกับผู้ใช้เป็นเอกภาพ ตรงตามสามัญสำนึก การเปิดโอกาสให้ผู้ใช้ขัดจังหวะเร่ง หน่วง หยุด หรือเลิก การทำงาน และอื่นๆ ระหว่างการจินตทัศน์ได้ตามต้องการ



รูปที่ 1. โครงสร้างหลักของระบบจินตทัศน์อัลกอริธึม

โครงสร้างของระบบจินตทัศน์อัลกอริธึมที่กำลังพัฒนาในงานวิจัยนี้ ประกอบด้วยส่วนจำเพาะ (module) ต่างๆ ดังแสดงในรูปที่ 1 ส่วนจำเพาะต่างๆนี้จะมีการส่งถ่ายข้อความหรือสัญญาณซึ่งกันและกัน ส่วนจำเพาะที่สำคัญส่วนแรกของระบบคือ ภาควิจิตทัศน์อัลกอริธึม ส่วนจำเพาะนี้แบ่งย่อยเป็นส่วนจำเพาะย่อยต่างๆคือ ภาควิจิตทัศน์อัลกอริธึม ซึ่งมีการรับข้อมูลที่สร้างจากภาคผลิตข้อมูลเข้า โดยจะแสดงภาพการทำงานของอัลกอริธึมผ่านทางภาคนำเสนอ ซึ่งสามารถแสดงภาพการทำงานในมุมมองนำเสนอต่างๆกันตามความต้องการของผู้ใช้

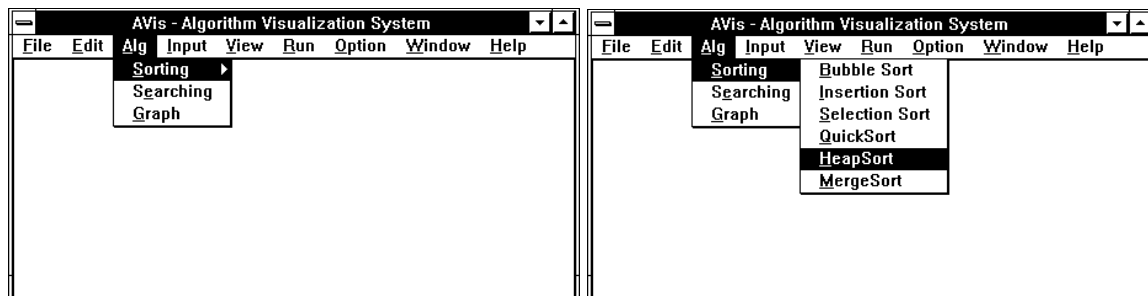
¹ มีบุคคลอยู่สองประเภทที่มีบทบาทในระบบจินตทัศน์อัลกอริธึม ประเภทแรกคือ ผู้สร้าง ซึ่งคือผู้พัฒนาการจินตทัศน์ โดยเขียนอัลกอริธึมที่สนใจ ใส่จุดสนใจ ณ ตำแหน่งต่างๆในอัลกอริธึมตามต้องการ พร้อมทั้งใช้ส่วนจำเพาะต่างๆ และคำสั่งคำสั่งต่างๆของระบบ ประกอบกันเป็นการจินตทัศน์สำหรับอัลกอริธึมนั้นๆ อีกบุคคลหนึ่งคือ ผู้ใช้ ซึ่งใช้ระบบที่ได้รวบรวมอัลกอริธึมต่างๆที่ผู้สร้างหลายคนสร้างไว้

เนื่องจากผู้ใช้อาจต้องการเปรียบเทียบการทำงานระหว่างอัลกอริธึมต่างๆ ที่แก้ไขปัญหเดียวกันพร้อมๆ กัน ดังนั้นจึงจำเป็นต้องมี *ภาคประสานจังหวะ* การทำงาน และการแสดงผลภาพ เพื่อให้การจินตทัศน์สื่อความหมายเชิงเปรียบเทียบในแง่ของความเร็วในการทำงานที่ถูกต้องเป็นไปตามจังหวะ หรือความเร็วที่ถูกกำหนด

ส่วนจำเพาะของระบบจินตทัศน์อีกส่วนหนึ่งคือ *ภาคประสานกับผู้ใช้* ส่วนจำเพาะนี้ทำให้ระบบจินตทัศน์ทำงานตามที่ผู้ใช้ต้องการ ตัวอย่างเช่นผู้ใช้สามารถเริ่ม เลิก หยุดชั่วคราว เร่ง หรือ หน่วงการทำงานของอัลกอริธึมตามต้องการ หรือผู้ใช้สามารถ เพิ่ม เลิก หรือ เปลี่ยนมุมมองการนำเสนอการทำงานได้ เป็นต้น

3.1 อัลกอริธึม

อัลกอริธึมในที่นี้คืออัลกอริธึมซึ่งผู้สร้างการจินตทัศน์ต้องการให้ผู้ใช้ศึกษา ตัวอย่างเช่นในรูปที่ 2 แสดงการเลือกอัลกอริธึมที่ต้องการจินตทัศน์การทำงาน ในขั้นแรกผู้ใช้ต้องเลือกหมวดของอัลกอริธึมที่ต้องการศึกษา จากนั้นจึงเลือกให้ละเอียดลงไปอีกว่าต้องการจินตทัศน์อัลกอริธึมใด ในที่นี้เลือก HeapSort



รูปที่ 2. ตัวอย่างการเลือกอัลกอริธึมการเรียงลำดับข้อมูลแบบ HeapSort

นอกจากการเขียนตัวอัลกอริธึมแล้ว หน้าที่สำคัญของผู้สร้างการจินตทัศน์อัลกอริธึมอีกประการหนึ่งคือการใส่จุดสนใจในอัลกอริธึม จุดสนใจแบ่งออกเป็นสองประเภทคือ จุดสนใจเพื่อรับข้อมูล และจุดสนใจเพื่อแสดงผล จุดสนใจเพื่อรับข้อมูลนั้นมีไว้เพื่อระบุว่ามีข้อมูลประเภทใดที่อัลกอริธึมต้องการ ส่วนจุดสนใจเพื่อแสดงผลมีไว้เพื่อระบุว่ามีข้อมูลที่ผู้สร้างการจินตทัศน์เห็นสมควรให้มีการแสดงผล ไม่ว่าจะเป็นภาพ ภาพเคลื่อนไหว หรือเสียง ตัวอย่างในรูปที่ 3 แสดงโปรแกรมย่อยภาษา Visual Basic ที่แทนอัลกอริธึมการเรียงลำดับข้อมูลแบบเลือก (selection sort) [1] ที่ใส่จุดสนใจแล้ว เริ่มต้นจะมีการขอรับจำนวนข้อมูลที่จะถูกเรียงลำดับ (*InputEvent_HowManyKeys*) จากนั้นจึงขอรับข้อมูลตามจำนวนเท่าที่ระบุมา (*InputEvent_ReadKey*) เมื่อรับข้อมูลครบ จะทำการส่งข้อมูลแต่ละตัวออกแสดงผลจนครบ (*OutPutEvent_NewKey*) แล้วจึงเริ่มการเรียงลำดับข้อมูลแบบเลือก จะสังเกตได้ว่า ทุกๆครั้งที่มีการสลับข้อมูล จะบอกให้มีการเปลี่ยนภาพการแสดงผล (*OutputEvent_Swap*) ควบคู่ไปด้วย แสดงว่าผู้สร้างพบว่าการสลับตำแหน่งของข้อมูลถือว่าเป็นจุดสนใจที่ต้องแสดงออกให้ผู้ใช้เห็น ดังนั้นในฐานะผู้ใช้เมื่อสั่งให้การเรียงลำดับทำงานแล้วจะเห็นการเปลี่ยนแปลงเกิดขึ้น ทุกครั้งที่ข้อมูลมีการเปลี่ยนตำแหน่ง ข้อสังเกตอีกประการหนึ่งคืออัลกอริธึมที่ผู้สร้างการจินตทัศน์เขียนขึ้นนี้ จะเป็นอิสระจากลักษณะมุมมองการนำเสนอซึ่งผู้ใช้ต้องการเห็น หรือแหล่งที่มาและลักษณะของข้อมูลขาเข้า ปัจจัยเหล่านี้จะเป็นตัวซึ่งผู้ใช้เลือกใช้เองตามที่ระบบมีไว้ให้ การที่ใส่จุดสนใจที่ไหน อย่างไรคงต้องขึ้นอยู่กับลักษณะการทำงานของอัลกอริธึมนั้นๆ

```
Sub Alg_SelectionSort_Code()  
    Dim iMaxVal, iMaxPos As Integer
```

```

Dim iData( 1 To MAXN ) As Integer
Dim N, hSync As Integer

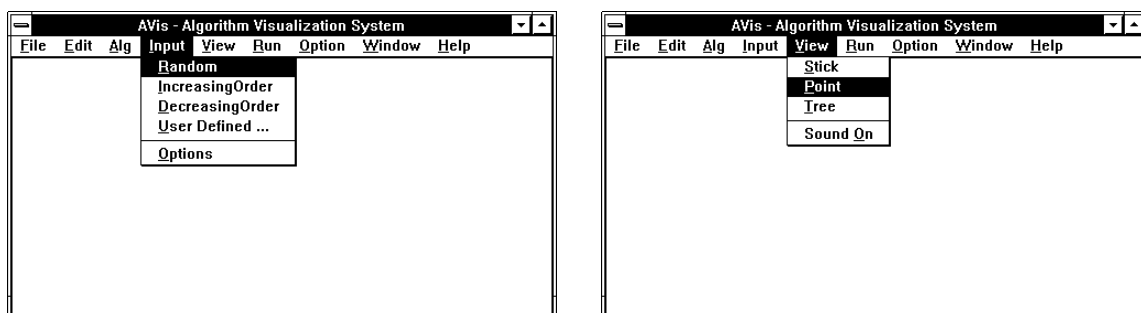
hSync = SyncOpen()
InputEvent_HowManyKeys( N )
for i% = 1 to N
    InputEvent_ReadKey( iData(i%) )
next i%
for i% = 1 to N
    OutputEvent_NewKey( i%, iData(i%) )
next i%
for i% = N to 2 step -1
    iMaxPos = i%
    for j% = 1 to i%-1
        SyncNotify( hSync )
        if ( iData(iMaxPos) < iData(j%) ) then iMaxPos = j%
    next j%
    t% = iData(i%)
    iData(i%) = iData(iMaxPos)
    iData(iMaxPos) = t%
    OutputEvent_Swap( i%, iMaxPos )
next i%
SyncClose( hSync )
End Sub

```

รูปที่ 3. ตัวอย่างอัลกอริทึมการเรียงลำดับข้อมูลแบบเลือกซึ่งใส่จุดสนใจ

3.2 ภาคผลิตข้อมูลขาเข้า

หลังจากเลือกอัลกอริทึมแล้ว ผู้ใช้ระบบจะต้องเลือกภาคผลิตข้อมูลขาเข้าเพื่อระบุถึงลักษณะของข้อมูลขาเข้าที่จะให้กับอัลกอริทึมระหว่างการทำงาน ตัวอย่างเช่นอัลกอริทึมการเรียงลำดับข้อมูล ผู้ใช้สามารถกำหนดข้อมูลที่จะถูกเรียงลำดับ (ดังแสดงในรูปที่ 4 ซ้าย) ให้มีลักษณะเป็นข้อมูลแบบสุ่ม แบบที่เรียงลำดับอยู่แล้ว แบบที่เรียงลำดับกลับกับที่ต้องการ หรือแบบที่ผู้ใช้กำหนดค่าของข้อมูลเอง เป็นต้น จากนั้นระบบจินตทัศน์อัลกอริทึม จะทำการส่งถ่ายข้อมูลกันเองกับตัวอัลกอริทึม (ตัวอย่างเช่น โดยผ่านฟังก์ชัน *InputEvent_ReadKey* ดังแสดงในโปรแกรมในรูปที่ 3)



รูปที่ 4. ตัวอย่างการเลือกภาคผลิตข้อมูลขาเข้า และการเลือกมุมมอง

การกำหนดข้อมูลขาเข้าในบางครั้งนั้น ผู้ใช้ต้องเป็นผู้กำหนดค่าของข้อมูล ซึ่งขึ้นกับลักษณะของข้อมูลอื่นๆ เช่นในปัญหาการหาเส้นทางที่สั้นที่สุดในกราฟ ผู้ใช้จะต้องระบุคู่ node ที่ต้องการหาเส้นทางสั้นสุด โดยผู้ใช้สามารถเลื่อนเมาส์ไปวางบนตำแหน่งของ node ที่ต้องการบนจอภาพ แล้วกดปุ่มของเมาส์ ให้ระบบรับรู้ node ที่ผู้

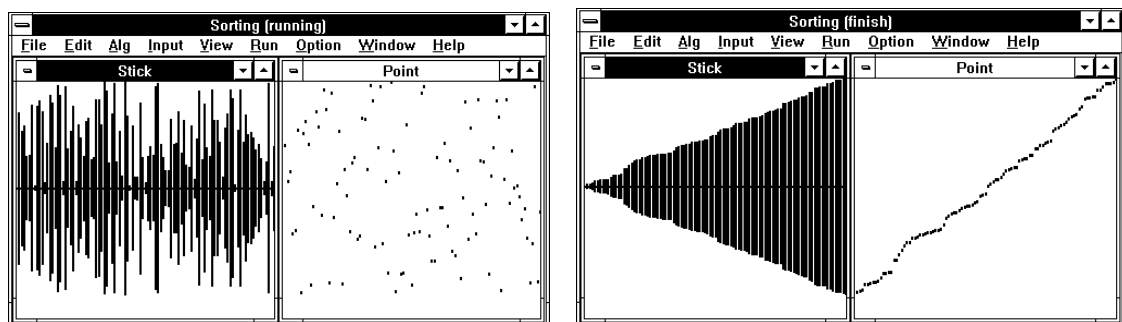
ใช้เลือก ในกรณีนี้ระบบจะทำการแปลงตำแหน่งของเมาส์บนจอภาพมาสู่หมายเลขของ node ดังนั้นภาคผลิตข้อมูลซ้ำเข้าจะต้องมีการสื่อสารติดต่อกับภาคนำเสนอด้วย (ที่จะได้กล่าวในหัวข้อย่อยถัดไป)

3.3 ภาคนำเสนอ

หลังจากผู้ใช้ต้องเลือกว่าจะจินตทัศน์อัลกอริทึมใด ต้องการให้ลักษณะของข้อมูลซ้ำเข้าเป็นอย่างไรแล้ว ผู้ใช้ยังต้องเลือกด้วยว่าจะให้ระบบจินตทัศน์แสดงการทำงานของอัลกอริทึมในลักษณะใดด้วย (ดังตัวอย่างในรูปที่ 4 ขวา) เราเรียกภาพการแสดงการทำงานของอัลกอริทึมว่า มุมมอง โดยทั่วไปการทำงานของอัลกอริทึมหนึ่งๆจะมีได้หลายมุมมอง แต่ละมุมมองจะเหมาะกับเหตุการณ์ หรือจุดที่ผู้ใช้สนใจ ตัวอย่างเช่นมุมมองการแสดงโครงสร้างข้อมูล มุมมองการแสดงการเปลี่ยนแปลงข้อมูลเชิงเวลา มุมมองการเปรียบเทียบหรือความสัมพันธ์ของข้อมูล เป็นต้น

แต่ละมุมมองที่ผู้ใช้เลือกจะปรากฏบนจอภาพในลักษณะของวินโดว ซึ่งผู้ใช้สามารถขยาย ย่อ ย้าย หรือ ยกเลิกมุมมองได้ทุกขณะ การเปลี่ยนแปลงภาพของมุมมองจะเป็นไปอย่างค่อยเป็นค่อยไป (incrementally) โดยจะต้องตอบสนองกับจุดสนใจอย่างฉับพลัน ตามที่อัลกอริทึมต้องการ โดยผ่านจุดสนใจเพื่อการแสดงผล (ตัวอย่างเช่นโดยผ่านฟังก์ชัน *OutputEvent_Swap* ดังแสดงในโปรแกรมในรูปที่ 3) นอกจากนี้เมื่อผู้ใช้อาจเลือกวัตถุหนึ่งบนมุมมอง (อาจทำได้โดยการเลื่อนเมาส์ไปบนวัตถุนั้น แล้วกดปุ่มของเมาส์) มุมมองนั้นจะต้องสามารถเปลี่ยนตำแหน่งของวัตถุนั้นบนวินโดวไปสู่ข้อมูลที่แท้จริงในอัลกอริทึม เพื่อการสื่อสารกับภาคผลิตข้อมูลซ้ำเข้า

รูปที่ 5 แสดงตัวอย่างของภาคนำเสนอการเรียงลำดับข้อมูลในสองมุมมอง คือ *Stick* และ *Point* ภาพที่ปรากฏในมุมมอง *Stick* นั้น เส้นตรงหนึ่งเส้นแทนข้อมูลหนึ่งตัว โดยที่ความยาวของเส้นจะแปรตามค่าของข้อมูล ส่วนมุมมองแบบ *Point* จะแสดงข้อมูลด้วยจุด ที่มีพิกัด (x, y) กำกับ ค่า y ของจุดจะแปรตามค่าของข้อมูล และค่า x จะแทนตำแหน่งของข้อมูล รูปที่ 5 (ก) แสดงถึงข้อมูลขณะเริ่มทำงานโดยการกำหนดให้ผลิตข้อมูลแบบสุ่มจำนวน 100 ตัว เมื่อเรียงลำดับเสร็จแล้ว (จากน้อยไปมาก) จะได้ผลในรูปที่ 5 (ข)



(ก) เมื่อเริ่มต้นทำงาน

(ข) เมื่อเรียงลำดับเสร็จ

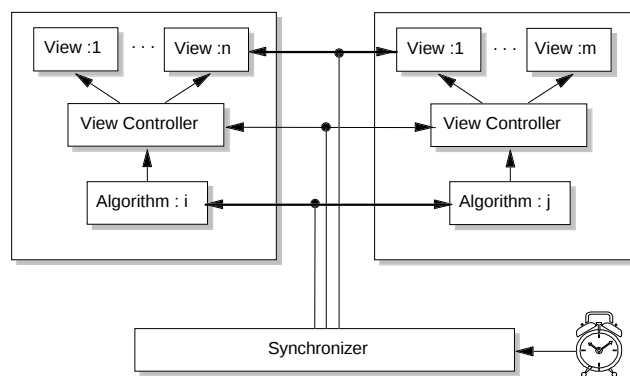
รูปที่ 5. ตัวอย่างมุมมองแสดงการทำงานของการทำงานของการเรียงลำดับข้อมูล

3.4 ภาคประสานจังหวะ

หน้าที่ของการประสานจังหวะการนำเสนอและการประสานจังหวะการทำงานของอัลกอริทึม เปรียบได้กับ ไวยากรณ์ซึ่งควบคุมการบรรเลงดนตรีของนักดนตรีหลายคน เพื่อให้การบรรเลงเข้าจังหวะกันไป การประสาน

จังหวัดมีจุดประสงค์เพื่อการเข้าจังหวัดกันตามคำสั่ง การดำเนินการพื้นฐาน (basic operation) การดำเนินการพื้นฐานนี้เป็นมาตรวัดประสิทธิภาพการทำงานของอัลกอริทึม โดยผู้สร้างการจินตทัศน์จะต้องเป็นผู้กำหนดว่าอัลกอริทึมหนึ่งๆนั้น มีคำสั่งอะไรในอัลกอริทึมนั้น ที่เป็นมาตรวัดดังกล่าว ตัวอย่างเช่นการใช้จำนวนคำสั่งการเปรียบเทียบของข้อมูล เป็นมาตรวัดประสิทธิภาพเชิงเวลาของอัลกอริทึมในการค้นหา หรือเรียงลำดับข้อมูล เป็นต้น ดังตัวอย่างในรูปที่ 3 จะเห็นได้ว่าเมื่อเริ่มทำงาน จะมีการเรียกฟังก์ชัน *SyncOpen* เพื่อขอรับบริการการประสานจังหวัด จากนั้นจะมีการเรียก *SyncNotify* ทุกๆครั้งก่อนทำการเปรียบเทียบข้อมูล (ซึ่งเป็น basic operation สำหรับ selection sort) ดังนั้นอัลกอริทึมใดที่ทำงานจนถึง basic operation ก่อนอัลกอริทึมอื่น ก็จะต้องรอจนกว่าทุกๆอัลกอริทึมจะทำถึง basic operation ทั้งนี้เพื่อให้ทำงานเข้าจังหวัดจำนวน basic operation กันไป เมื่อเสร็จสิ้นการทำงานของอัลกอริทึมใดก็จะเรียก *SyncClose* เป็นการบอกเลิกการประสานจังหวัดของอัลกอริทึมนั้น

นอกจากหน้าที่ในการประสานจังหวัดแล้ว ตัวประสานจังหวัดในระบบจินตทัศน์อัลกอริทึมจะเป็นตัวควบคุมความเร็วของการนำเสนอ และควบคุมความเร็วการทำงานของอัลกอริทึมด้วย โดยความเร็วเหล่านี้จะถูกเปลี่ยนแปลงได้ตามความต้องการของผู้ใช้ โครงสร้างความสัมพันธ์ของอัลกอริทึม ภาคควบคุมการนำเสนอ มุมมอง และภาคประสานจังหวัด ถูกแสดงในรูปที่ 6



รูปที่ 6. การประสานจังหวัดการทำงานและการนำเสนอ

4. ระบบจินตทัศน์อัลกอริทึมภายใต้สภาพปฏิบัติการไมโครซอฟต์วินโดวส์

ไมโครซอฟต์วินโดวส์ (Microsoft Windows) เป็นสภาพปฏิบัติการ (operating environment) ของเครื่องคอมพิวเตอร์ส่วนบุคคลที่ได้รับความนิยมเป็นอย่างสูง หัวข้อนี้จะได้กล่าวถึงการใช้ GUI, Multitasking, MDI, และ DLL [12] อันเป็นคุณสมบัติของวินโดวส์ที่เอื้ออำนวยในการพัฒนาส่วนจำเพาะต่างๆของระบบจินตทัศน์อัลกอริทึมดังนี้

4.1 การประสานกับผู้ใช้แบบกราฟิก (Graphical User Interface - GUI) เพื่อให้เกิดการใช้งานที่เป็นเอกภาพ และง่ายต่อความเข้าใจของผู้ใช้ เนื่องจากการกำหนดตำแหน่งและลักษณะของปุ่ม สัญลักษณ์ กล้องโต้ตอบ และเมนู ถูกกระทำตามแบบ CUA [13] อันเป็นรูปแบบการประสานกับผู้ใช้ที่ใช้ในซอฟต์แวร์ต่างๆที่ทำงานภายใต้วินโดวส์

4.2 การทำงานแบบหลายภารกิจ (Multitasking) วินโดวส์เป็นสภาพปฏิบัติการที่เอื้ออำนวยต่อการทำงานแบบหลายภารกิจ ทำให้สามารถพัฒนาการจินตทัศน์ได้มากกว่าหนึ่งอัลกอริทึม และมากกว่าหนึ่งมุมมอง ได้สะดวกมากขึ้น

4.3 การแสดงมุมมองการนำเสนอแบบหลายเอกสาร (Multiple Document Interface - MDI) MDI เป็นคุณสมบัติอันหนึ่งของวินโดวส์ที่เอื้ออำนวยต่อการแสดงและจัดการวินโดวส์ย่อยหลายวินโดวส์ที่อยู่ภายในวินโดวส์หลัก โดยจะใช้วินโดวส์ย่อยนี้เองแทนมุมมองของการจินตทัศน์ ทำให้การย่อ ขยาย ย้าย จัดวาง ตำแหน่งของมุมมองตกเป็นภาระของตัวระบบ ผู้สร้างการจินตทัศน์ไม่จำเป็นต้องจัดการเกี่ยวกับภาระดังกล่าว

4.4 การใช้คลังโปรแกรมเชื่อมโยงแบบพลวัต (Dynamic Linking Library - DLL) DLL คือคลังโปรแกรมที่เก็บไว้ในแฟ้มข้อมูลที่เป็นอิสระจากโปรแกรมที่เรียกใช้ เมื่อมีการเรียกใช้ฟังก์ชันใน DLL จึงมีการนำคลังโปรแกรม DLL เข้าสู่หน่วยความจำและเชื่อมโยงกับโปรแกรมที่เรียกใช้ หากมีหลายๆโปรแกรมเรียกใช้ฟังก์ชันเดียวกัน ฟังก์ชันที่ถูกเรียกจะใช้เนื้อที่ในหน่วยความจำเพียงชุดเดียว อีกทั้งหากมีการเปลี่ยนแปลง DLL ก็ไม่จำเป็นต้องสร้างรหัสจุดหมาย (object code) ใหม่สำหรับโปรแกรมที่เรียกใช้ บริการต่างๆที่มีในระบบจินตทัศน์อัลกอริทึมจะถูกออกแบบและพัฒนาขึ้นในรูปแบบของ DLL โดยผู้สร้างการจินตทัศน์สามารถเรียกใช้ตามกฎเกณฑ์ที่ระบบตั้งขึ้น บริการแบบ DLL ที่ระบบมีให้ใช้มีดังนี้

4.4.1 DLL สำหรับการแสดงและการเปลี่ยนแปลงภาพในมุมมอง ประกอบไปด้วยคำสั่งคำสั่งของการวาดพื้นฐาน และการสร้างวัตถุเคลื่อนไหวพร้อมการเปลี่ยนแปลงภาพและสีต่างๆ ที่เหมาะกับการนำไปสร้างมุมมองเพื่อการจินตทัศน์ โดยจะมีการเรียกใช้ฟังก์ชันทางกราฟิกส์ของวินโดวส์โดยตรง เพื่อความเร็วในการแสดงผล เช่นการวาดรูปทรงเรขาคณิตขั้นพื้นฐาน การทำ Blitting โดยใช้ฟังก์ชัน *BitBlt* *PatBlt* *StretchBlt* เป็นต้น

4.4.2 DLL สำหรับการประสานจังหวะ ประกอบไปด้วยชุดคำสั่งในการประสานจังหวะ เช่น *SyncOpen*, *SyncClose*, *SyncBegin*, *SyncEnd*, *SyncNotify*, *SyncSpeed* เป็นต้น การประสานจังหวะอาจมีการหน่วงการทำงานของบางอัลกอริทึม เพื่อให้ทำงานเข้าจังหวะกัน ดังนั้นระหว่างการหน่วงนี้ฟังก์ชันการประสานจังหวะจะต้องอนุญาตให้การประสานกับผู้ใช้เป็นไปอย่างปกติ จึงจำต้องคำนึงถึง *ข่าวสาร* (message) ที่มีการส่งถึงกันภายในวินโดวส์ด้วย

5. บทสรุป

การจินตทัศน์อัลกอริทึมเป็นกรรมวิธีหนึ่งในการศึกษาทำความเข้าใจการทำงานของอัลกอริทึม ด้วยการถ่ายภาพ และการเปลี่ยนแปลงของภาพ เป็นสื่อในการแสดงถึงขั้นตอนการทำงานของอัลกอริทึม ในบางครั้งการใช้กรรมวิธีนี้จะเปิดโลกของผู้ทำการศึกษาเข้าสู่พฤติกรรมของการทำงานของอัลกอริทึมที่ไม่เคยจินตนาการและคาดหวังมาก่อน

บทความฉบับนี้ได้นำเสนอการออกแบบและพัฒนาระบบจินตทัศน์อัลกอริทึมภายใต้สภาพปฏิบัติการไม่ใคร่ขอพตวินโดวส์ โดยได้กล่าวถึงภาพรวมของโครงสร้างหลักของระบบ โดยอาศัยคุณสมบัติสำคัญของวินโดวส์หลายประการ มาใช้เพื่อให้การออกแบบระบบเป็นไปอย่างมีโครงสร้าง เป็นระบบ และใช้งานง่าย อันได้แก่ การใช้การประสานกับผู้ใช้แบบกราฟิกส์ การจินตทัศน์อัลกอริทึมแบบหลายภารกิจ การแสดงมุมมองการนำเสนอ

การจินตทัศน์แบบหลายมุมมองโดยใช้การประสานแบบหลายเอกสาร การใช้คลังโปรแกรมเชื่อมโยงแบบพลวัต เพื่อเก็บฟังก์ชันต่างๆของระบบ จึงเห็นได้ว่าสภาพปฏิบัติการวินโดวส์เอื้ออำนวยต่อการออกแบบ การพัฒนา และการใช้งานระบบจินตทัศน์อัลกอริทึม

แผนงานการวิจัยที่เหลือจะประกอบด้วย การพัฒนาต้นแบบของส่วนต่างๆของระบบจินตทัศน์อัลกอริทึม ตามโครงร่างที่ได้นำเสนอในที่นี้ จากนั้นจะสร้างการจินตทัศน์อัลกอริทึมตัวอย่าง เช่นกลุ่มอัลกอริทึมที่เกี่ยวกับการเรียงลำดับข้อมูล กลุ่มอัลกอริทึมที่เกี่ยวกับการค้นหาข้อมูล กลุ่มอัลกอริทึมที่เกี่ยวกับปัญหาทางกราฟ เป็นต้น โดยในช่วงแรกต้นแบบนี้จะถูกทดลองใช้ประกอบการเรียนการสอน ในลักษณะของการประกอบการบรรยาย และการเข้าทำการจินตทัศน์ในห้องทดลอง

เอกสารอ้างอิง

1. Robert Sedgewick, *Algorithms in C*, Addison-Wesley, 1990
2. Kenneth C. Knowlton. *L6: Bell Telephone Laboratories Low-Level Linked List Language*, two 16mm black and white sound film, 1966
3. Kellog S. Booth, *PQ Trees*, 16mm color silent film, 12 minutes, 1975
4. Ronald M. Baecker and David Sherman, *Sorting Out Sorting*, 16mm color sound film, 30 minutes, 1981
5. Brad A. Myers, "Incense: A System for Displaying Data Structures," *Computer Graphics*, Vol 17, No.3, July 1983, 115-125
6. David B. Baskerville, "Graphic Presentation of Data Structures in the DBX Debugger," *Report No. UCB/CSD 86/260*, University of California at Berkeley, Berkeley, CA, October 1985
7. Thomas G. Moher, "PROVIDE: a Process Visualization and Debugging Environment," *Technical Report*, University of Illinois at Chicago, Chicago, IL July 1985
8. Ronald M. Baecker, "Two System Which Produce Animated Representations of the Execution of Computer Programs," *ACM SIGCSE Bullentin*, Vol. 7, No. 1, February 1975, 158-167.
9. Edward Yarwood, *Toward Program Illustration*, M.Sc. Thesis, Dept. of Computer Science, University of Toronto, Toronto, ON, 1974.
10. James M. De Boer, *A System for the Animation of Micro-PL/I Programs*, M.Sc. Thesis, Dept. of Computer Science, University of Toronto, Toronto, ON, 1974.
11. Marc H. Brown, *Algorithm Animation*, The MIT Press, Cambridge, MA, 1988.
12. Charles Petzold, *Programming in Windows 3.1 3rd Edition*, Microsoft Press, 1992
13. System Application Architecture, Common User Access: Advanced Interface Design Guide, First Edition, IBM, Document number SY0328-300-R00-1089, June 1989