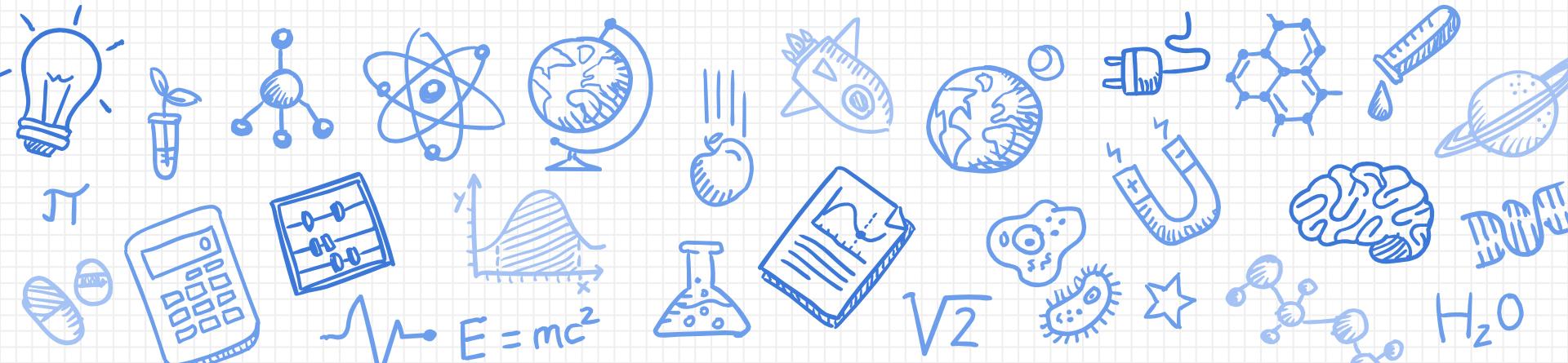
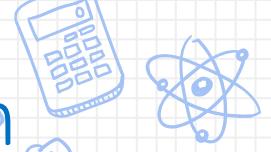


การซ่อนทีบภาพ

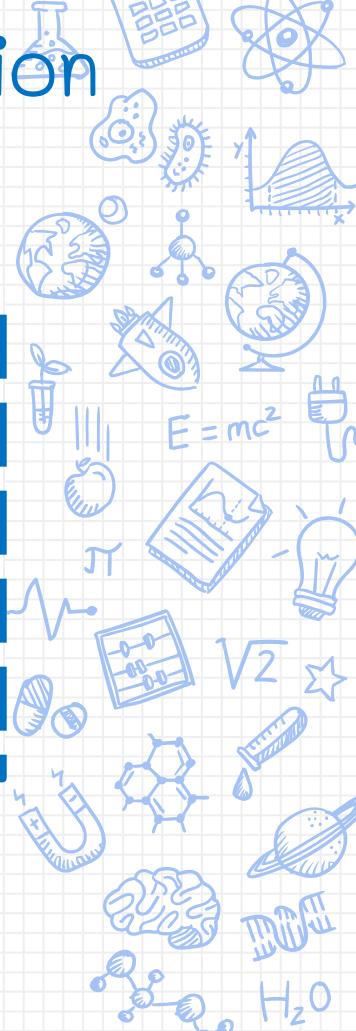
Image Registration



การซ่อนทับภาพ Image Registration



การซ่อนทับภาพ (Image Registration) เป็นกระบวนการในการจัดวางภาพตั้งแต่สอง ภาพขึ้นไปลงบนระบบเดียวกันอย่างสอดคล้อง เพื่อพิจารณารายละเอียดรวมทั้งตำแหน่งของ วัตถุบนภาพทั้งหมดได้ในขณะเดียวกัน ด้วยเหตุนี้การซ่อนทับภาพจึงถูกนำใช้อย่างแพร่หลายในการวิเคราะห์ผลภาพทางการแพทย์ เช่น การวิเคราะห์การเจริญเติบโตของเนื้อร้าย มะเร็ง เป็นต้น เป็นหนึ่งในบทนี้จะกล่าวถึงประโยชน์ของการซ่อนทับภาพและวิธีในการซ่อนทับภาพที่มีอยู่ในปัจจุบัน



ประเภทของการซ่อนทับภาพ

การซ่อนทับภาพสามารถแบ่งออกได้เป็น 3 ประเภทดังต่อไปนี้

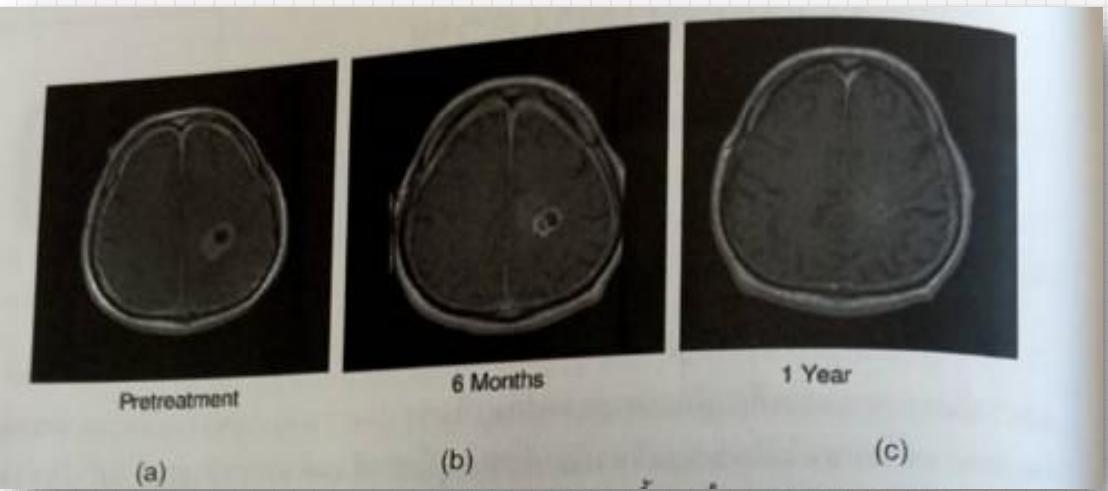
1

วัตถุนิดเดียวกันและถ่ายภาพด้วยวิธีเดียวกัน

การซ่อนทับภาพของวัตถุนิดเดียวกันและถ่ายภาพด้วยวิธีเดียวกัน

พบมากในการซ่อนทับภาพนิดเดียวกันแต่ถ่ายคนและเวลาเพื่อวิเคราะห์หาความเปลี่ยนแปลงที่เกิดขึ้นในช่วงเวลาต่างกัน

ตัวอย่าง



รูป 10.1 (a) แสดงภาพถ่าย MRI ของครีบส์ก่อนการรักษาเนื้องอกในสมอง (b) แสดงภาพถ่าย MRI หลังการรักษา 6 เดือน (c) ภาพถ่าย MRI หลังการรักษา 1 ปี

H_2O

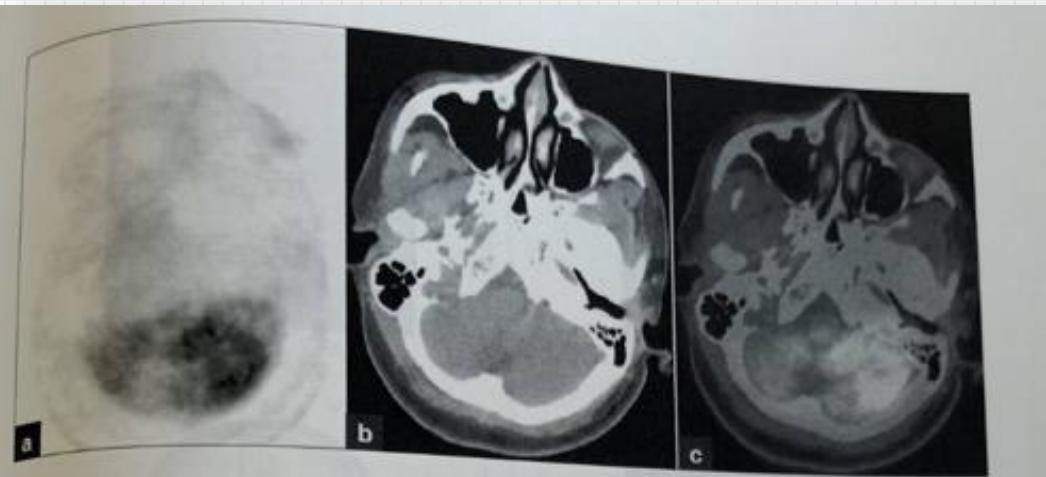
ประโยชน์ของการซ่อนทับภาพ

2

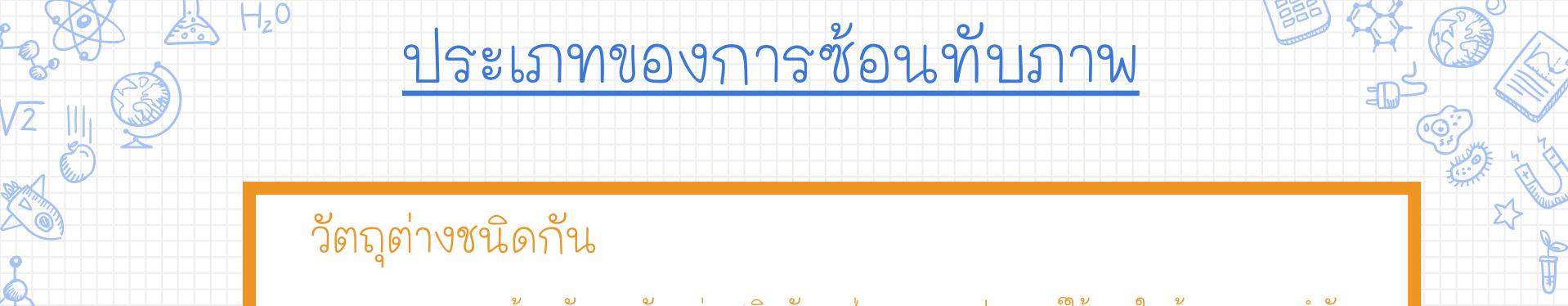
วัตถุนิดเดียว กันแต่ถ่ายภาพต่างวิธีกัน

การซ่อนทับภาพวัตถุนิดเดียว กันแต่ถ่ายภาพต่างวิธีกัน พบประยุกต์ใช้ในงานนี้ที่เราต้องการดึงเอาข้อมูลของเทคโนโลยีในการถ่ายภาพที่ต่างกันมาไว้ในภาพเดียว กัน พบมากในทางการแพทย์ที่ต้องการนำเออลักษณะเด่นของวิธีการถ่ายภาพมาร่วมกัน เช่น การถ่ายภาพด้วยเครื่องเอกซเรย์คอมพิวเตอร์

ตัวอย่าง



รูป 10.2 (a) ภาพถ่าย SPECT ของครึีษะผู้ป่วย; (b) แสดงภาพถ่าย CT ของครึีษะผู้ป่วย;
(c) ภาพผลลัพธ์ของการซ้อนทับภาพ CT และ SPECT

H_2O 

ประโยชน์ของการซ้อนทับภาพ

3

วัตถุต่างชนิดกัน

การซ้อนทับภาพวัตถุต่างชนิดกัน ล้วนมากพบประยุกต์ใช้งานในด้านการจดจำวัตถุ (Pattern Recognition) และการตัดแยกวัตถุ (Classification) ในกรณีนี้เราจะมีฐานข้อมูลเป็นภาพ 2 มิติเช่นภาพใบหน้าคนเป็นต้น เมื่อเราต้องการตรวจสอบภาพใบหน้าทดสอบว่ามีอยู่ในฐานข้อมูลหรือเปล่า และถ้ามีสามารถระบุได้ว่าเป็นใคร ล้วนมากเราจำเป็นต้องซ้อนทับภาพทดสอบให้อยู่ในการเรียงตัว ที่ตรงกันกับภาพในฐานข้อมูลก่อนที่จะทำการ วิเคราะห์ต่อไป นอกจากนั้นอาจใช้ในการซ้อนทับภาพตัดขาดส่องต่างบุคคลกัน เพื่อใช้ เปรียบเทียบลักษณะทางกายภาพที่ต่างกันหรือเหมือนกันอย่างไร

วิธีซ่อนทับภาพ (Registration Technique)

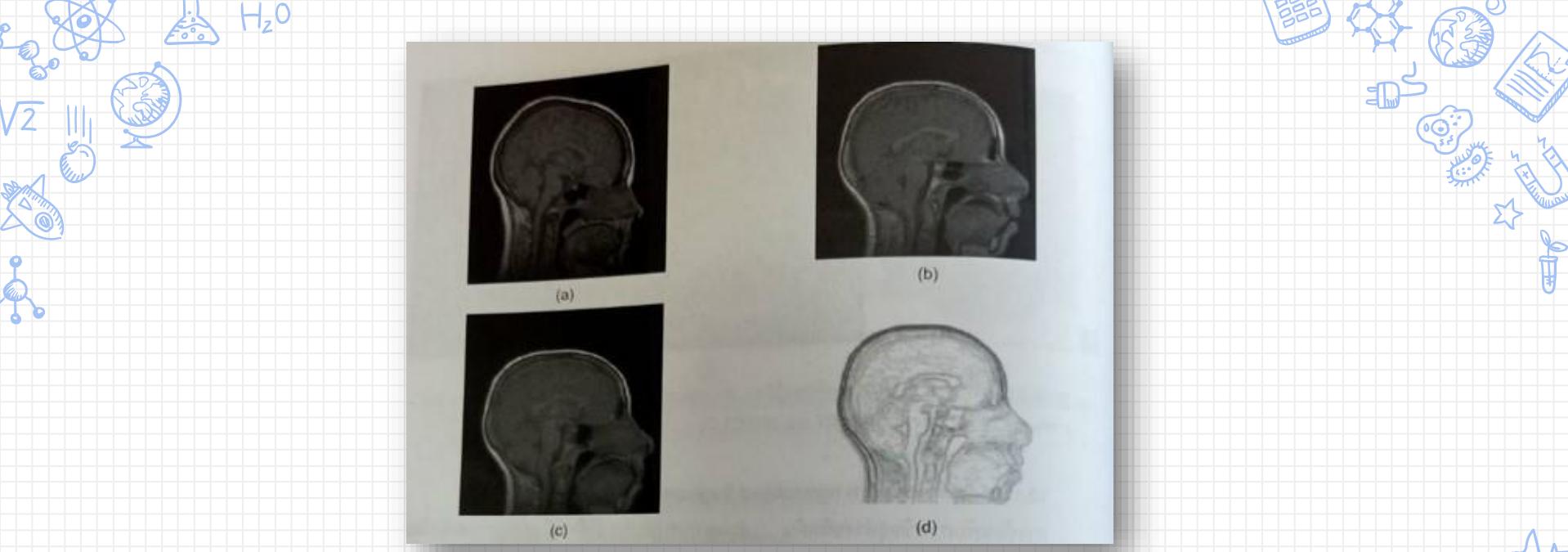
เราสามารถจำแนกวิธีซ่อนทับภาพได้ 4 วิธีตามคุณลักษณะของภาพ

ที่นี่มาใช้ซ่อนทับ ได้แก่



1. วิธีซ่อนทับภาพแบบใช้แลนمارك (Landmark-point based registration)

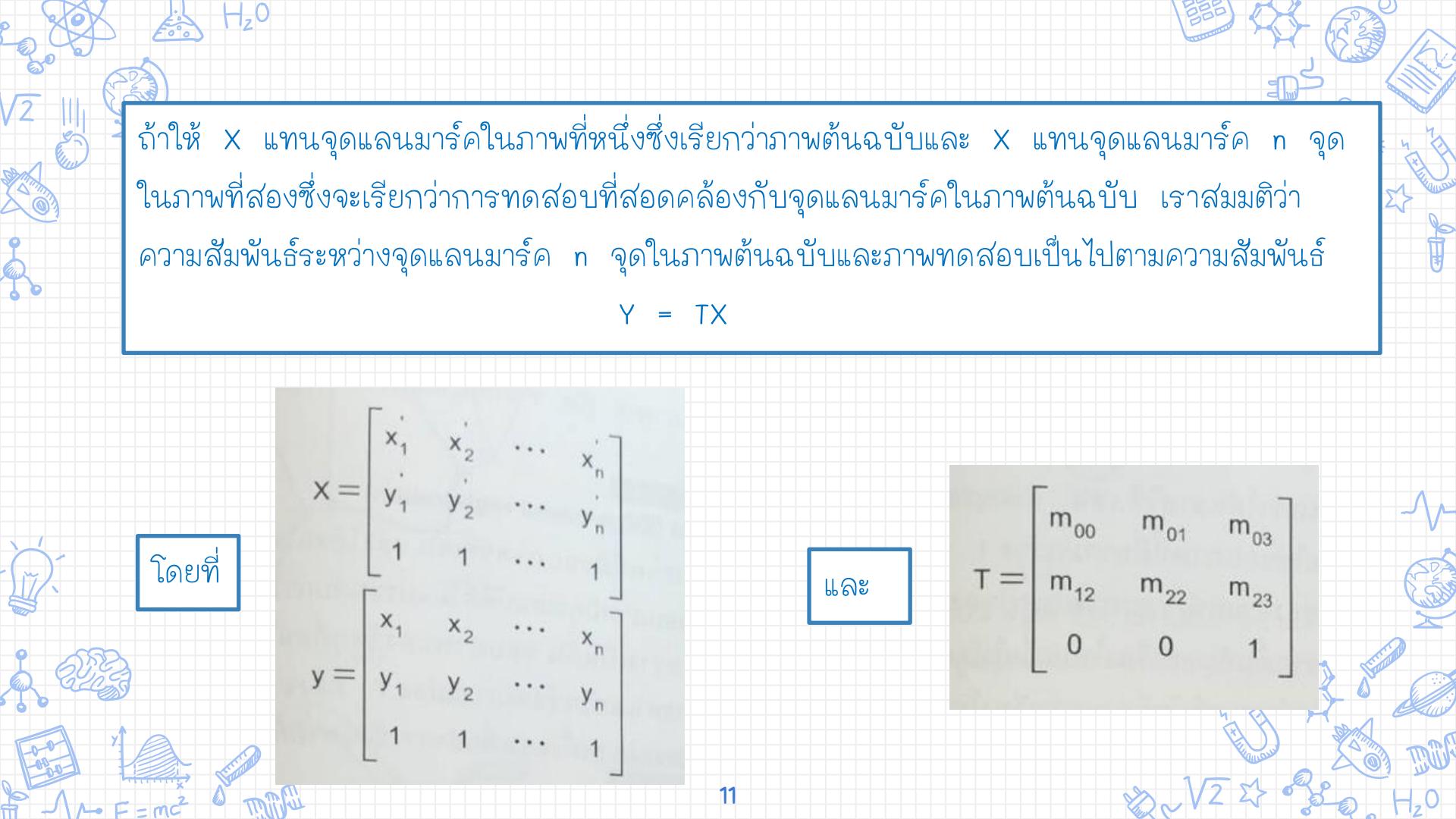
วิธีการซ่อนทับภาพแบบใช้แลนمارك จะใช้จุดที่สอดคล้องกันระหว่างจุดบนภาพทั้งสองภาพในการซ่อนทับภาพ และเรียกจุดเหล่านั้นว่าแลนمارك ดังนั้นความแม่นยำของการซ่อนทับภาพจึงขึ้นอยู่กับความถูกต้องของแลนماركที่หาได้แลนماركอาจจะเป็นจุดซึ่งเป็นคุณสมบัติของวัตถุหรือไม่ใช่ก็ได้ เช่น ในภาพทางการแพทย์จะมีแลนمارك เช่น จุดบนอวัยวะหรือขอบของกระดูกเป็นคุณสมบัติของวัตถุ



รูป 10.3 การซ้อนทับภาพวัตถุต่างชนิดกัน; (a) ภาพถ่าย MRI สมองของบุคคลที่ 1; (b) ภาพถ่าย MRI สมอง ของบุคคลที่ 2; (c) การแปลงภาพ MRI ของสมองบุคคลที่ 1 เพื่อซ้อนทับกับภาพ MRI บุคคลที่ 2; (d) ภาพของขอบเขตซ้อนทับภาพ

แลนมาრ์คที่ได้จากการเอาแอล่งกำเนิดและสังติดไว้ที่ตัวผู้ป่วยตามจุดต่าง ๆ หรือทำเครื่องหมายไว้ที่ตัวผู้ป่วย เป็นแลนมาร์คซึ่งไม่ใช่คุณสมบัติของวัตถุโดยตรง และวิธีที่ได้มาซึ่งแลนมาร์คนั้นมีหลายวิธี เช่น การหาแลนมาร์คด้วยมือหรือการกำหนดจุดซึ่งจำเป็นต้องอาศัยความแม่นยำและความรู้เกี่ยวกับภาพชนิดนั้น ๆ เช่นการหาแลนมาร์คในภาพทางการแพทย์เป็นต้น นอกจากนี้

หลังจากที่เราได้แลนมาร์คแบบอัตโนมัติหรือกึ่งอัตโนมัติ เช่นจุดเปลี่ยนเลี้นโค้งหรือจุดมุขของรูปปานามาก็อนทับ ขั้นตอนต่อไปเราหาเมทริกซ์ของการแปลง (Transformation Matrix) ได้จากการความสัมพันธ์ระหว่างแลนมาร์คบนภาพที่หนึ่งกับภาพที่สอง เช่นในภาพที่มีความล้มพื้นที่กันแบบเชิงเส้นสามารถหาเมทริกซ์ของการแปลงได้จากการประมาณ Minimized Mean Square Error (MMSE)



ถ้าให้ X แทนจุดและมาร์คในภาพที่หนึ่งซึ่งเรียกว่าภาพต้นฉบับ และ X แทนจุดและมาร์ค n จุดในภาพที่สองซึ่งจะเรียกว่าการทดสอบที่สอดคล้องกับจุดและมาร์คในภาพต้นฉบับ เราสมมติว่า ความล้มเหลวของจุดและมาร์ค n จุดในภาพต้นฉบับและภาพทดสอบเป็นไปตามความล้มเหลว

$$Y = TX$$

โดยที่

$$X = \begin{bmatrix} \cdot & \cdot & \cdots & \cdot \\ x_1 & x_2 & \cdots & x_n \\ \cdot & \cdot & \cdots & \cdot \\ y_1 & y_2 & \cdots & y_n \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

$$Y = \begin{bmatrix} \cdot & \cdot & \cdots & \cdot \\ x_1 & x_2 & \cdots & x_n \\ \cdot & \cdot & \cdots & \cdot \\ y_1 & y_2 & \cdots & y_n \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

และ

$$T = \begin{bmatrix} m_{00} & m_{01} & m_{03} \\ m_{12} & m_{22} & m_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

H_2O

ดังนั้นถ้าเราทราบ T เราสามารถทำการแปลงเรขาคณิตของภาพให้ภาพทั้งสองข้อนี้หันกันได้ ใน การหา T เรา尼ิยามค่าผิดพลาดกำลังสองดังนี้

$$\varepsilon^2 = (v_1 - TX)^T(Y - TX) \quad (10.1)$$

ทำการหาอนุพันธ์สมการ (10.1) เทียบกับ T และเทียบผลลัพธ์กับศูนย์ เราได้

$$\varepsilon^2 = -2xT(Y - TX) = 0$$

$$X^T Y = X^T T X$$

$$Y^T X = T^T X^T X$$

$$T^T = (Y^T X)(X^T X)^{-1}$$

$$T = (X^T X)^{-1}(X^T Y)$$

 (10.2)

$$\varepsilon^2 = -2xT(Y - TX) = 0$$

$$X^T Y = X^T T X$$

$$Y^T X = T^T X^T X$$

(10.2)

$$T^T = (Y^T X)(X^T X)^{-1}$$

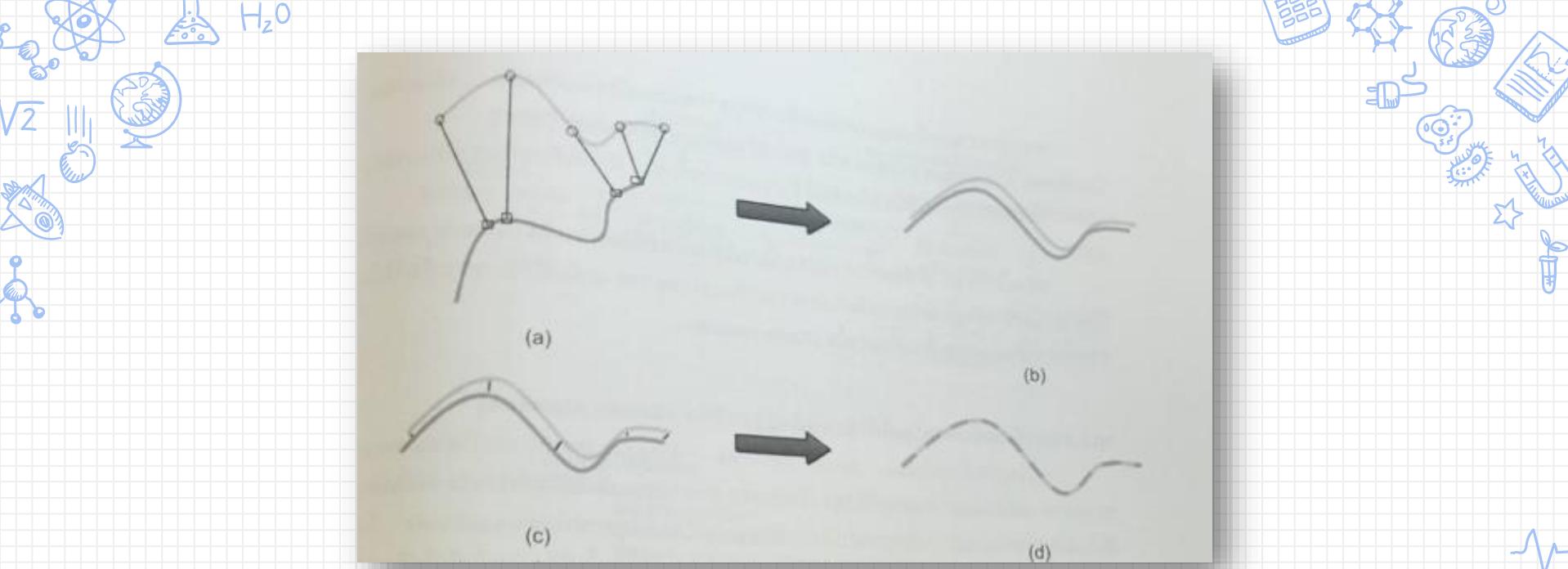
$$T = (X^T X)^{-1}(X^T Y)$$

สมการ (10.2) เรียกว่า สมการนอร์มอล (Normal Equation) หลังจากได้ T แล้วทำการแปลงเรขาคณิตของภาพเพื่อซ้อนทับภาพ X กับ Y รายละเอียดของการแปลงเรขาคณิตของภาพอยู่ในบทที่ 5

วิธีซ่อนทั่บภาพ (Registration Technique)

2. วิธีซ่อนทั่บภาพแบบใช้ขอบภาพ (Edge-based registration)

วิธีซ่อนทั่บภาพแบบนี้นิยมใช้กับภาพที่มีขอบภาพชัดเจน และไม่สนใจข้อมูลอื่นบนภาพยกเว้นขอบภาพเท่านั้น เนื่องจากขอบภาพมีคุณสมบัติที่ไม่แปรผันกับการเปลี่ยนแปลงของลักษณะเดลล์อม เช่นความมืด ความสว่างเป็นต้น ขอบภาพของวัตถุที่สูญเสียหายในภาพ ต้นฉบับและภาพหลอกลวงนำมาใช้ในการหาเมทริกซ์ของการแปลง T หลังจากได้เมทริกซ์ การแปลง T แล้ว จึงทำการแปลงเรขาคณิตของภาพทั้งสองเพื่อทำการซ่อนภาพกัน โดยขั้นตอนของการหาขอบภาพสามารถทำได้หลายวิธี เช่น Template Matching หรือ Gradient เป็นต้น



รูป 10.4 ลักษณะ ICP; (a) แlenmar์คบันเลี้นโครงสร้างทั้งสอง (วงกลมและสี่เหลี่ยม);
 (b) การซ่อนทับภาพครั้งที่ 1 ; (c) แlenmar์คใหม่บันเลี้นขอบ; (d) การซ่อนทับภาพครั้งที่ 1

วิธีซ่อนทับภาพ (Registration Technique)

3. วิธีซ่อนทับภาพแบบพิจารณาความเหมือนกันของภาพ

หลักการของวิธีซ่อนทับภาพแบบพิจารณาความเหมือนกันของภาพคือจะไม่ดึงคุณสมบัติใดๆ ของภาพ แต่จะทำการแปลงภาพ ให้มีนิ่งและทำการวัดความเหมือนของภาพทั้งสอง และจะทำการแปลงภาพนั้นอีกจนกว่าภาพทั้งสองจะเหมือนกันในที่สุด การวัดความเหมือนของภาพจะพิจารณาจากคุณสมบัติของภาพ เช่น ความล้มเหลว, ค่าล้มเหลวสิทธิ์, ฟังก์ชันความล้มเหลว หรือผลรวมของความแตกต่าง วิธีซ่อนทับภาพวิธีนี้จำเป็นต้องใช้กระบวนการทำซ้ำ ทำให้ใช้เวลาในการประมวลผลมาอย่างไรก็ตาม วิธีซ่อนทับภาพแบบนี้อาจให้ผลลัพธ์ที่ผิดพลาดได้ เนื่องจากคุณสมบัติที่ได้กล่าวไปข้างต้นนั้นมีค่าแป้งกับค่าความเข้มแสงของภาพ จะนั่นวิธีนี้จึงเหมาะสมที่จะใช้กับ สภาพที่ถ่ายด้วยเทคนิคเดียวกันเท่านั้น

วิธีซ่อนทับภาพ (Registration Technique)

4. การซ่อนทับภาพโดยใช้เจเนติกอัลกอริทึม (Genetic Algorithm)

ขบวนการพื้นฐานของ เจเนติกอัลกอริทึม จำลองแนวคิดเกี่ยวกับวิวัฒนาการทางชีวภาพ ของสายพันธุ์ต่างๆ ที่มีความสามารถในการปรับตัวและแข่งขันเพื่อยังคงเผื่องเผื่อง ดังนั้นเจเนติกอัลกอริทึมจึงมุ่งเน้นที่จะหาค่าที่ดีที่สุด ซึ่งมาจากลักษณะที่ดีที่สุดของสายพันธุ์นั้นๆ ที่มีความสามารถในการปรับตัว เช่น ความเร็ว ความแรง ความแม่นยำ เป็นต้น กระบวนการค้นหาค่าที่ดีที่สุดนี้จะดำเนินการโดยการสร้างชุดรหัสพันธุกรรม (chromosome) ที่ประกอบด้วยรหัสประจำตัว (genes) ที่มีค่าตั้งแต่ 0 ถึง 1 หรือ -1 ถึง 1 ตามที่กำหนด แล้วคำนวณค่าฟิตเนส (fitness) ของแต่ละรหัสพันธุกรรม ค่าฟิตเนสจะแสดงถึงความสามารถของสายพันธุ์นั้นในการดำเนินการ ต่อจากนั้นจะนำรหัสพันธุกรรมที่มีค่าฟิตเนสสูงสุดมาใช้ในการสร้างรหัสพันธุกรรมใหม่ ผ่านกระบวนการคัดเลือก (selection) และการขับเคลื่อน (mutation) ที่จะเพิ่มความหลากหลายของรหัสพันธุกรรม ทำให้ระบบสามารถค้นหาค่าที่ดีที่สุดได้ในที่สุด

จากรูป วัฏจักรของเจเนติกอัลกอริทึม สามารถแบ่งออกเป็น 5 ขั้นตอนดังนี้

1

กำหนดค่าเริ่มต้นให้ประชากรในป่าเจเนติก อัลกอริทึม โดยการสุ่ม และทำการหาค่าพิตเนส ของโครโมโซม โดยใช้ พงก์ชั่นเป้าหมาย หลังการหาค่าแล้ว โครโมโซม ทั้งหมดจะถูกจัดลำดับโดยค่าพิตเนส

2

เลือกโครโมโซม จากบรรพบุรุษ โดยที่โครโมโซมที่ มีค่าพิตเนสที่ดี ที่สุดนั้นมีโอกาสที่จะอยู่รอดสูง

3

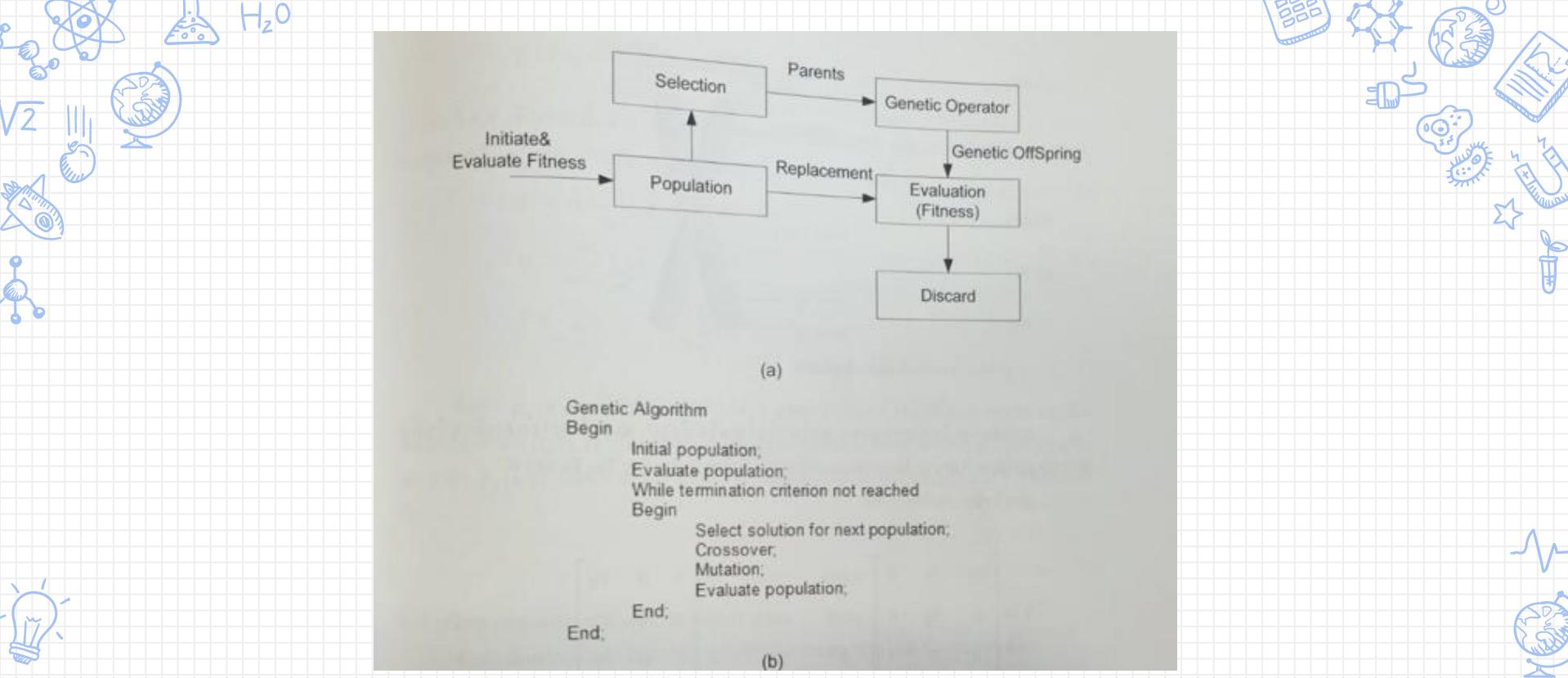
ให้กำเนิดประชากรรุ่นใหม่โดยใช้ ตัวดำเนินการ เจเนติก ซึ่งประกอบด้วย ครอสโอเวอร์ (Crossover) และ มิวเตชัน (mutation)

4

หาค่าพิตเนสของ โครโมโซม โดยใช้ พงก์ชั่นเป้าหมาย หลังจากการหาค่าแล้ว โครโมโซม ทั้งหมดจะถูกจัดลำดับโดยค่าพิตเนส

5

แทนที่จะประชากรุ่นแก่ที่มีค่าพิเศษที่น้อยที่สุดในปัจจุบัน
อีกครึ่งหนึ่ง ด้วยประชากรุ่นใหม่ที่มีค่าพิเศษที่ดีกว่า
ตรวจล้อบคำตอบว่าถูกต้องตามเงื่อนไขที่วางไว้หรือไม่ ถ้าไม่
ย้อนกลับไปทำข้อที่ 3



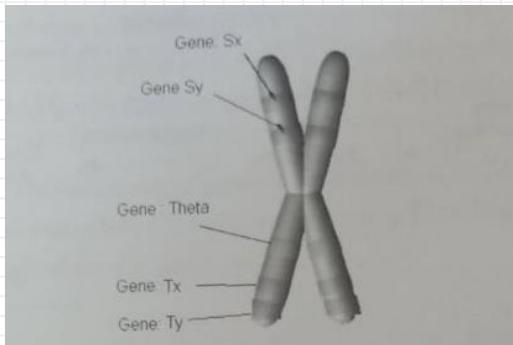
รูป 10.5 วัสดุจัดของ เจเนติก อัลกอริทึม: (a) บล็อกໄດอะแกรม: (b) อัลกอริทึม

การซ่อนทับภาพแบบพิจารณาความเหมือนกันของภาพรวมทั้งการซ่อนทับภาพแบบ ICP เราจำเป็นต้องทำการซ่อนทับภาพเบื้องต้น (Initial Alignment) ในตำแหน่งที่ใกล้เคียงกัน การซ่อนทับภาพที่ถูกต้อง การซ่อนทับภาพที่ไม่ทำการซ่อนทับภาพเบื้องต้น อาจทำให้การซ่อนทับภาพผิดหรือการซ่อนทับภาพนี้จะเข้าลู่จุดโคลออลมินิมัม.(local minimum) ในหัวข้อนี้เราจะเรียนรู้วิธีการซ่อนทับภาพโดยใช้เจเนติกอัลกอริทึมซึ่งไม่จำเป็นต้องทำการซ่อนทับภาพเบื้องต้น

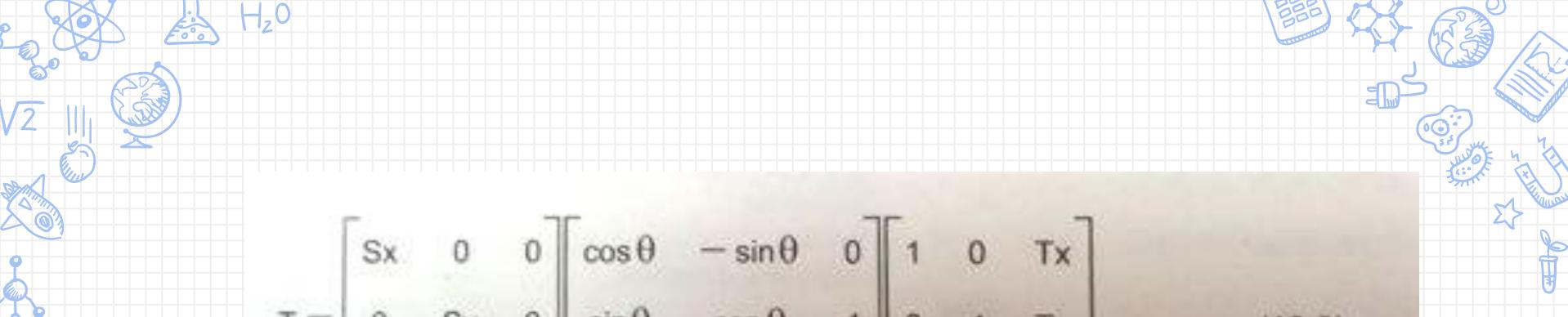
H_2O

รูปแบบการแทนโครง主义

เราสมมติว่าภาพที่ใช้ช้อนทับภาพการแปลงแบบสีมิลาริตี้ (Similarity Transformation) ซึ่งการแปลงแบบสีมิลาริตี้นั้นประกอบไปด้วยการหมุน การเลื่อนพิกัด และ การสเกล โดยสามารถแทนโครง主义ได้ดังรูป ซึ่งประกอบด้วยยืนที่มี พารามิเตอร์สำหรับการแปลงภาพแบบสีมิลาริตี้ ซึ่งได้แก่ S_x , S_y , T_x , T_y และ θ



รูป 10.6 โครง主义ที่
ใช้ในการซ้อนทับภาพ



$$T = \begin{bmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & Tx \\ 0 & 1 & Ty \\ 0 & 0 & 1 \end{bmatrix} \quad (10.3)$$

เราจะใช้เจเนติกอัลกอริทึมในการหาพารามิเตอร์ Sx , Sy , Tx , Ty และ θ ที่ให้คำตอบในการซ่อนทับภาพทดสอบกับภาพต้นฉบับได้ดีที่สุด ค่า Sx , Sy , Tx , Ty และ θ เปรียบเสมือนเป็นค่าเยิน (Gene) ในโครโนโซม

ค่าพิณเนล

การซ่อนทับภาพแบบพิจารณาความเหมือนกันของภาพโดยใช้เจเนติก อัลกอริทึมนนั่น เราแทนค่าพารามิเตอร์ที่อยู่ในยีนของโครโนโซมลงในสมการ จากนั้นทำการแปลง เรขาคณิตของภาพที่ดีสุดของภาพซ่อนทับกับภาพต้นฉบับ เราวัดค่าพิณเนลซึ่งอาจใช้ส่วนกลับของ ค่าเฉลี่ยความผิดพลาดกำลังสอง (Mean-Squared Error) หรืออาจใช้ Correlation Coefficient ที่ใช้วัดความเหมือน ซึ่งแทนด้วยสมการ

H_2O 

ตัวดำเนินการเจนิติก (Genetic Operator)

ตัวดำเนินการเจนิติกใช้สำหรับการให้กำเนิดประชากรรุ่นใหม่ ซึ่งประกอบด้วย

การครอஸโอเวอร์ (Crossover)

การครอஸโอเวอร์ คือ การรวมกันใหม่ของยีนซึ่งก็คือการแลกเปลี่ยน
ยีนกันบางส่วนของ บรรพบุรุษ 2 ตัว เพื่อให้กำเนิดประชากรรุ่น
ใหม่ ซึ่งวิธีการครอஸโอเวอร์สามารถทำได้หลายวิธี สามารถแสดง
ตัวอย่างได้ดังต่อไปนี้

ตัวดำเนินการเจนติก (Genetic Operator)

1

การครอสโซเวอร์แบบ 1 จุด เป็นการนำยีนจากบรรพ บุรุษตัวแรกมาไว้ที่ประชากรรุ่นใหม่ จนกระทั่งถึงจุดครอสโซเวอร์ ก็จะนำยีนจากบรรพบุรุษตัวที่ 2 มาไว้ที่ประชากรรุ่นใหม่ ซึ่ง จุดครอสโซเวอร์ คือจุดที่จะ ทำการแลกเปลี่ยนยีน

2

การครอสโซเวอร์แบบ 2 จุด เป็นการนำยีนจากบรรพบุรุษตัวแรกมาไว้ที่ประชากรรุ่นใหม่ จนกระทั่งถึงจุดครอสโซเวอร์ที่ 1 ก็จะนำยีนจากบรรพบุรุษตัวที่ 2 มาไว้ที่ประชากรรุ่นใหม่ จนถึงจุดครอสโซเวอร์ที่ 2 จะนำยีนจาก บรรพบุรุษตัวที่ 1 มาไว้ที่ประชากรรุ่นใหม่

3

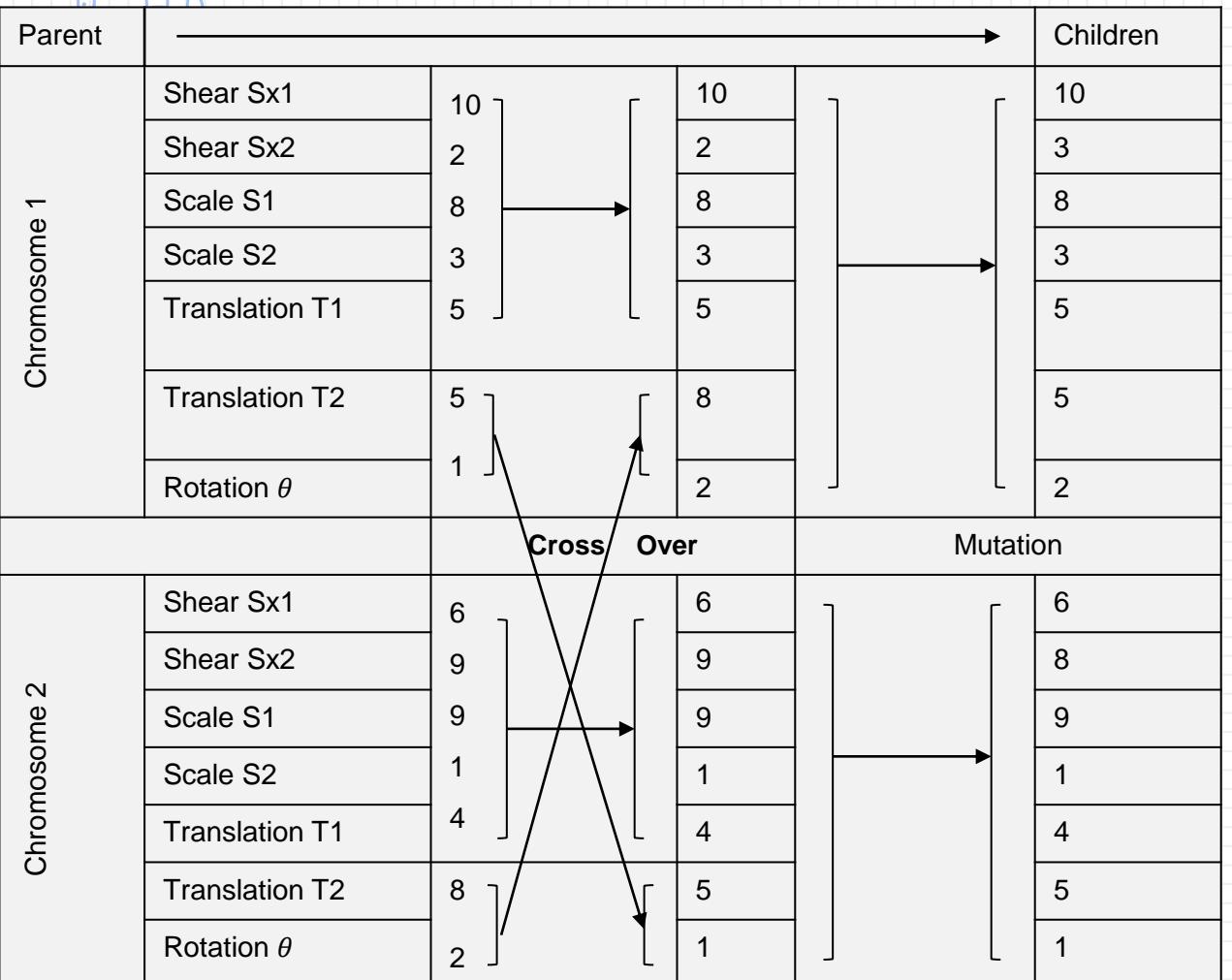
การครอสโซเวอร์ แบบอเริร์เมทธิค (Arithmetic Crossover)
เป็นการนำค่าของ บรรพบุรุษทั้งสองมาคำนวณทางคณิตศาสตร์
แล้วนำผลลัพธ์มาไว้ที่ประชากรรุ่นใหม่

H_2O 

ตัวดำเนินการเจนิติก (Genetic Operator)

การมิวเตชั่น (Mutation)

การมิวเตชั่น ใช้สำหรับหาค่าที่ไม่ครอบคลุม บริเวณที่ค้นหาการมิวเตชั่น (mutate) คือ การลับบิตรางยืนในโครงโโนซิมของทายาทที่เกิดใหม่ซึ่งจะได้ค่าใหม่ของตัวแปรเข้ามาแทนที่



รูป 10.7 แสดงตัวอย่าง
การครอสโอเวอร์ แบบ 1
จุดและการเกิดมิวเตชัน
ยืนที่เกิดมิวเตชันเขียนใน
รูปตัวเอียง

การเลือกโครโมโซม (Chromosome selection)

เป็นขั้นตอนการคัดเลือกโครโมโซม
รุ่นเก่า เพื่อใช้ในการสร้าง
โครโมโซมรุ่นใหม่ แนวคิดการ
วิวัฒนาการทางช่องรวมชาตินั้น
โครโมโซมที่มีค่าพิเศษที่ดีมีโอกาส
สูงกว่าที่จะรอดเพื่อให้กำเนิด
ประชากรรุ่นใหม่ ซึ่งวิธีการเลือก
นั้นมีหลายวิธีดังได้นำมาแสดงเป็น⁹
ตัวอย่าง ส่วนหนึ่งดังต่อไปนี้

วิธีการเลือก
แบบงล้อรุ่น
เลข

วิธีการเลือก
แบบบจัด
ตำแหน่ง

วิธีการเลือก
แบบสถานะ
ไม่
เปลี่ยนแปลง

วิธีการเลือก
แบบอัลกิทิ
ซึม

Matlab กับการซ้อนทับภาพ

ตัวอย่าง การซ้อนทับภาพโดยใช้แลนเดิมาร์ค Matlab มีคำสั่งใน Image Processing Toolbox ที่สามารถนำมาใช้ในการซ้อนทับภาพ แบบใช้แลนเดิมาร์ค ซึ่งแบ่งเป็น 5 ขั้นตอนดังนี้

1. อ่านภาพ 2 ภาพที่ต้องการซ้อนทับภาพหรือซ้อนทับกัน
 - orthophoto = imread('westconcordorthophoto.png');
 - figure(1), imshow(orthophoto)
 - unregistered = imread('westconcordaerial.png');
 - figure(1), imshow(unregistered)

รูป 10.8 แสดงตัวอย่าง 2 ภาพที่ต้องการซ่อนหันภาพ



(a)

orthophoto



(b)

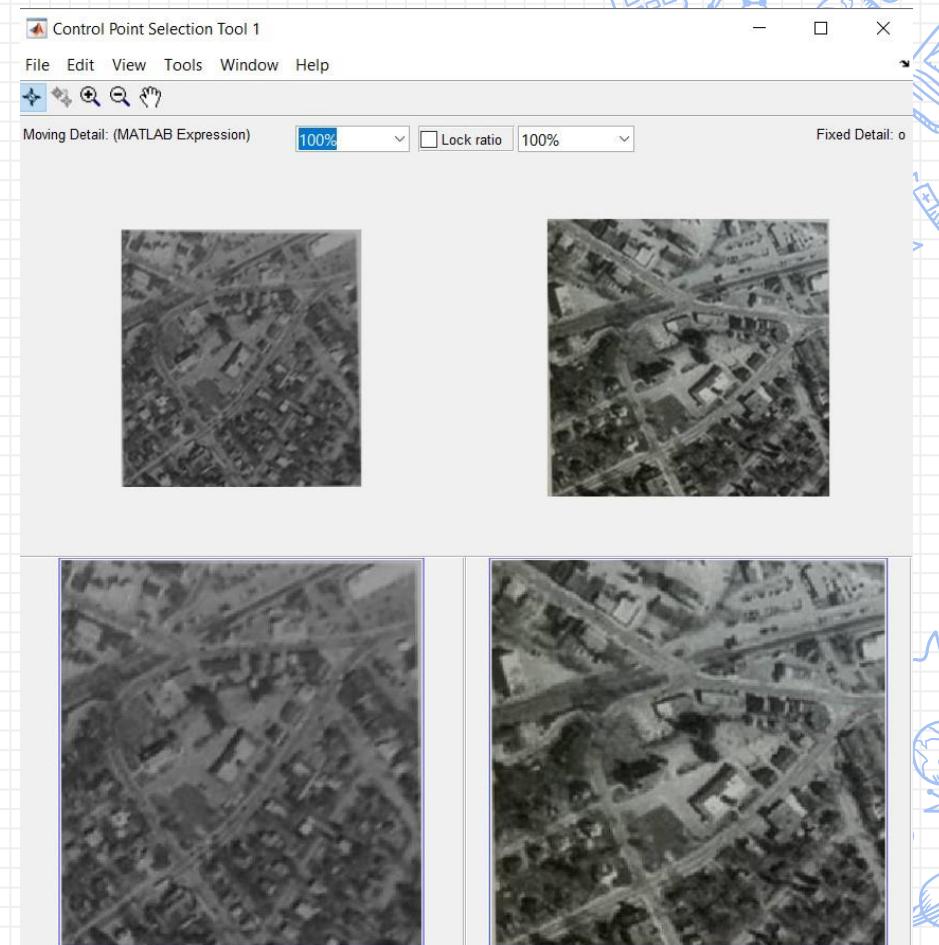
unregistered

2. เลือกจุดแลนด์มาร์คบนภาพทึ้งส่อง พิมพ์คำสั่ง

>> cpselect

(unregistered(:,:,1],orthophoto)

จะปรากฏหน้าต่างให้เลือกจุดแลนด์มาร์คในภาพทึ้งส่อง ดังแสดงในรูป 10.9



3. บັນທຶກຈຸດແລນມາრ্কໂດຍທຸກຄຮ່ງເນື່ອເລືອກ

ຈຸດແລນມາր্ক 1 ຈຸດບັນກາພຕົ້ນນີ້ບັບແລນກາພ
ທີ່ຕ້ອງການຊ້ອນທີ່ບັນກາພໃຫ້ໃຊ້ຄຳລົ່ງ Export

Points to Workspace ລອງພິມພົດຄຳລົ່ງ

ຕ້ອໄປນີ້ເພື່ອແສດງຈຸດແລນມາր්කທີ່ ເລືອກ

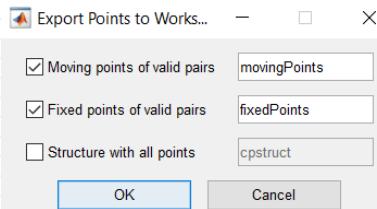
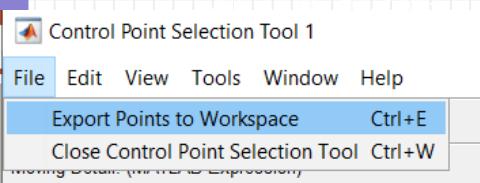
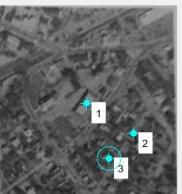
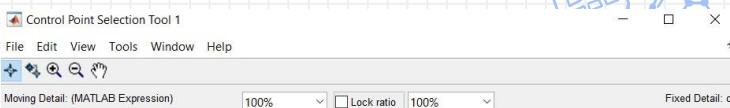
```
movingPoints           >> fixedPoints
```

```
movingPoints =          fixedPoints =
```

```
112.3750   124.6250   155.3750   160.8750
```

```
172.0000   163.0000   204.0000   185.0000
```

```
141.0000   195.0000   185.0000   210.0000
```

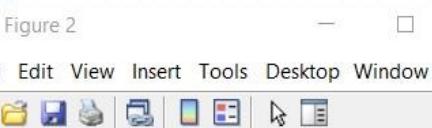
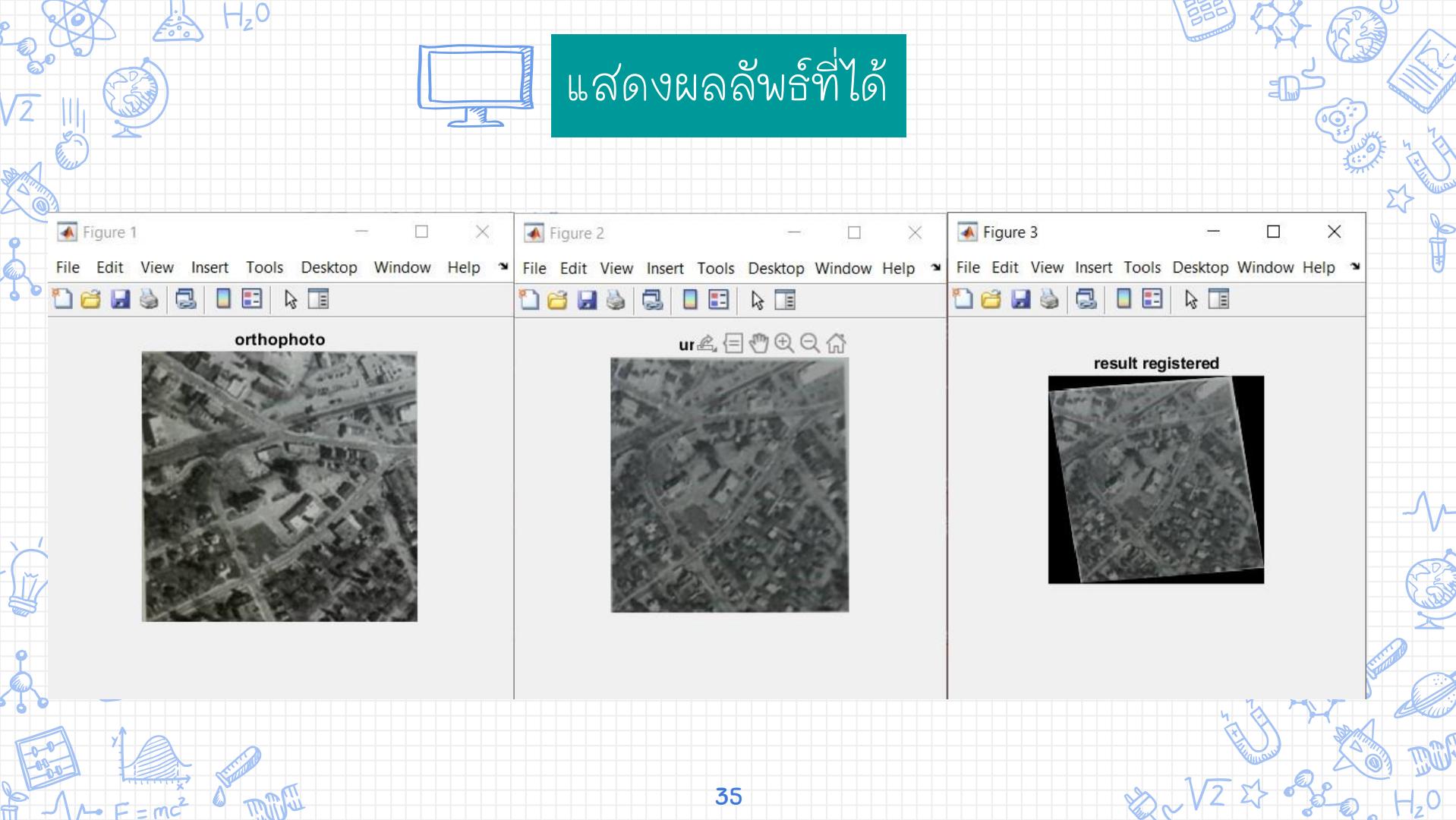


4. เลือกชนิดของการแปลงเรขาคณิตที่ต้องการ ตัวเลือกที่สามารถเลือกได้แสดงในตาราง พร้อมทั้งจำนวนและมาร์คที่ต้องการจากนั้นให้พิมพ์คำสั่ง

```
>> mytform = cp2tform(movingPoints,  
fixedPoints, 'affine');
```

5. ทำการแปลงภาพที่ต้องการซ่อนทับภาพเพื่อให้สามารถซ่อนทับกับภาพต้นฉบับโดยใช้คำสั่ง

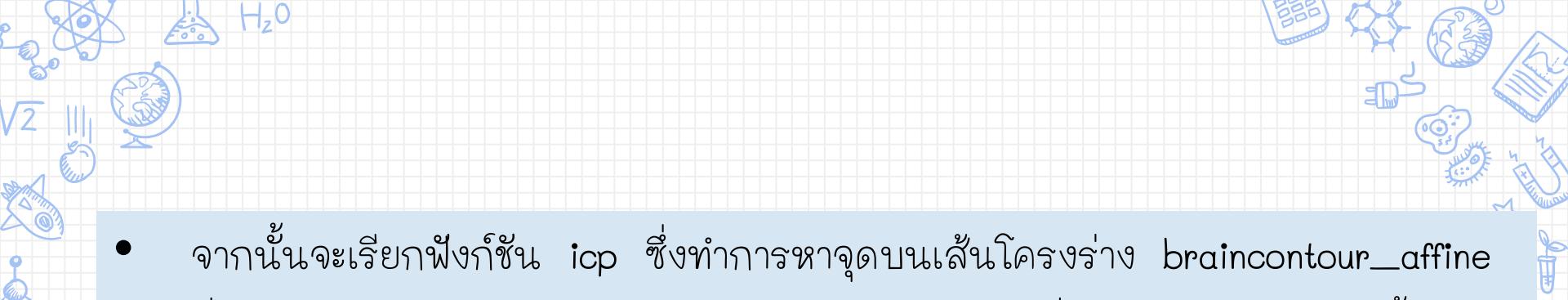
```
>> registered =  
imtransform(unregistered,mytform)
```



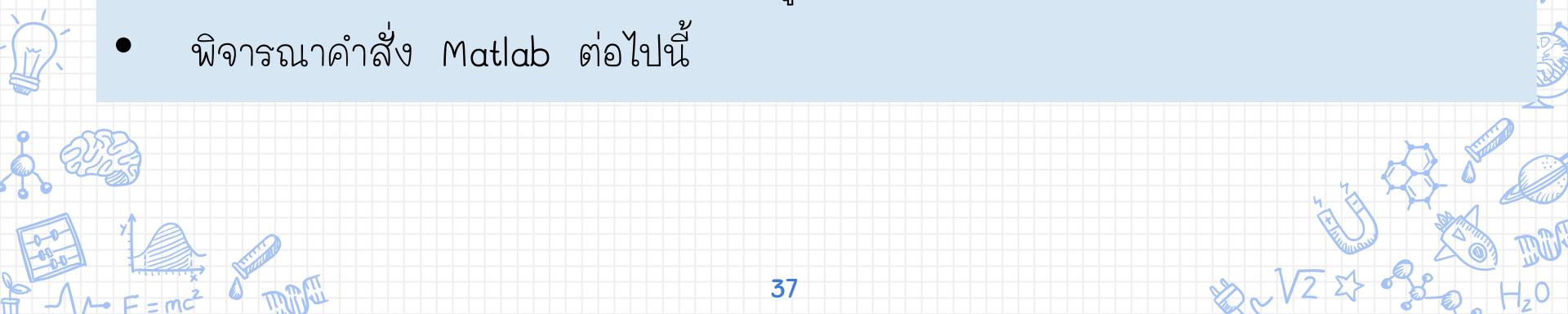
Ex 10.2 ใช้ Matlab เพื่อซ่อนหันเส้นขอบภาพ โดยใช้วิธี ICP (Iterative Closest Point)

ทำการซ่อนหันภาพเลี้นโครงสร้างชื่อ braincontour และ braincontour_affine

- ICP ไม่จำเป็นต้องทราบจุดแลนมาร์คบนเส้นโครงสร้าง
- เราจะทำการสุ่มบันเลี้นโครงสร้างตัวนฉบับ ในที่นี่คือเส้นโครงสร้าง braincontour ด้วยฟังก์ชัน subsample
- โดย subsample มีรูปแบบ subsample(in, n) โดยจะสุ่มเส้นโครงสร้าง in ทุกๆ n จุด เรียกว่าจุดเหล่านี้ว่า ff



- จานนี้จะเรียกฟังก์ชัน icp ซึ่งทำการหาจุดบนเส้นโครงสร้าง braincontour_affine ที่ใกล้กับจุดแลนมาრ์คในเส้นโครงสร้าง braincontour ที่สูมได้ เรียกว่า gg
- จากจุด ff และ gg ทำการคำนวนเมทริกซ์การแปลงที่หาได้ ได้อาทีพุทเป็น out
- แทนเส้นโครงสร้าง braincontour_affine ด้วยเส้นโครงสร้าง out ทำขบวนการที่กล่าวถึงขึ้อกิจ จนกว่าค่าผิดพลาดจะอยู่ในค่าที่ยอมรับได้
- พิจารณาคำลั่ง Matlab ต่อไปนี้



file ในตัวอย่างที่ 10.2 จะมีทั้งหมด 5 ไฟล์ แบ่งเป็น 3 ไฟล์ 1 ไฟล์ คือ EX102Mfile.m และ พงก์ชั่น 4 ไฟล์ ได้แก่ fcoeffa.m icp.m poly2a.m และ subsample.m ดังภาพ

| | | | |
|--|-------------------|----------------------|------|
|  EX102Mfile.m | 9/19/2021 4:13 PM | Objective C Sourc... | 3 KB |
|  fcoeffa.m | 9/19/2021 4:02 PM | Objective C Sourc... | 1 KB |
|  icp.m | 9/19/2021 7:37 PM | Objective C Sourc... | 3 KB |
|  poly2a.m | 9/19/2021 4:04 PM | Objective C Sourc... | 1 KB |
|  subsample.m | 9/19/2021 4:06 PM | Objective C Sourc... | 1 KB |

เตรียมฟังก์ชัน icp.m

```
%ฟังก์ชัน ICP บันทึกเป็น mfile ต่างหาก
function [out,ori,tra]=icp(dd,nnpb,dlimit)
size(dd);m=ans(1);
table=zeros(256,256,1);
for i=1:m
    x=ceil(dd(i,1));
    y=ceil(dd(i,2));
    z=1;
    if x==0
        x=1;
    end
    if y==0
        y=1;
    end
    if z==0
        z=1;
    end
```

```
table(x,y,z)=i;
end
sx=256;
sy=256;
sz=1;
size(nnpb);n=ans(1);
get=zeros(n,1);
for i=1:n
    skip=0;
    nx=ceil(nnpb(i,1));
    ny=ceil(nnpb(i,2));
    nz=ceil(nnpb(i,3));
    if nx<=0 | ny<=0 | nz<=0
        skip=1;
    end
    if nx>sx | ny>sy | nz>sz
        skip=1;
    end
    %[nx ny nz skip]
    if skip==0
        ii=table(nx,ny,nz);
        end
        move=1;
        while ii==0
            if skip==1
```

เตรียมพังก์ชัน icp.m (ต่อ)

```
    break;
end

movexp=nx+move;
if movexp<1
    movexp=1;
end
if movexp>sx
    movexp=sx;
end

ii=table(movexp,ny,nz);
if ii>0
    break;
end
    moveyp=ny+move;
if moveyp<1
    moveyp=1;
end
if moveyp>sy
    moveyp=sy;
end
```

```
ii=table(nx,moveyp,nz);
if ii>0
    break;
end
movezp=nz+move;
if movezp<1
    movezp=1;
end
if movezp>sz
    movezp=sz;
end

ii=table(nx,ny,movezp);
if ii>0
    break;
end
movexp=nx-move;
if movexp<1
    movexp=1;
end
if movexp>sx
    movexp=sx;
end

ii=table(movexp,ny,nz);
if ii>0
```

ຕັດຕິມີ້ນັກໜຳ icp.m (ຕ່ອ)

```
break;
end
moveyp=ny-move;
if moveyp<1
    moveyp=1;
end
if moveyp>sy
    moveyp=sy;
end

ii=table(nx,moveyp,nz);
if ii>0
    break;
end

movezp=nz-move;
if movezp<1
    movezp=1;
end
if movezp>sz
    movezp=sz;
end
```

```
ii=table(nx,ny,movezp);
if ii>0
    break;
end

move=move+1;
if move>dlimit
    break
end

end

get(i,1)=ii;

end
count=0;
for i=1:n
    if get(i)>0
        count=count+1;
        ori(count,:)=1*nnpb(i,:);
        tra(count,:)=1*dd(get(i,1,:));
    end
end
co=fcoeffa(ori,tra);
out=polya(dd,co);
size(out);m=ans(1);
%To homogeneous coordinate
```

เตรียมฟังก์ชัน fcoeffa.m

```
function [co]=fcoeffa(origin,transform);
m=size(origin);
n=size(transform);
if n(1) ~= m(1)
    error('Size unequal');
end
X=zeros(2*m(1),6);
Y=zeros(2*m(1),1);
for j=1:m(1)
    x=transform(j,1);y=transform(j,2);
    i=2*(j-1)+1;
    X(i,1)=x;
    X(i,2)=y;
    X(i,3)=1;
    Y(i,1)=origin(j,1);
    i=2*(j-1)+2;
    X(i,4)=x;
    X(i,5)=y;
    X(i,6)=1;
    Y(i,1)=origin(j,2);
end
coeff=(inv(X'*X))*X'*Y;
co(1:3,1)=coeff(1:3);
co(1:3,2)=coeff(4:6);
```

เตรียมฟังก์ชัน poly

%พิ้งก์ขั้นท้าการแปลงเรขาคณิตบันทึกเป็น mfile ดังหาก

```
function [out]=poly2a(transform,coeff);
m=size(transform);
X=zeros(m(1),3);

for i=1:m(1)
    x=transform(i,1);y=transform(i,2);
    X(i,1)=x;
    X(i,2)=y;
    X(i,3)=1;
end
```

out=X*coeff;

H_2O $\sqrt{2}$ 

เตรียมฟังก์ชัน subsample.m

%ฟังก์ชันสุ่มจุด บันทึกเป็น mfile ด่างหาก

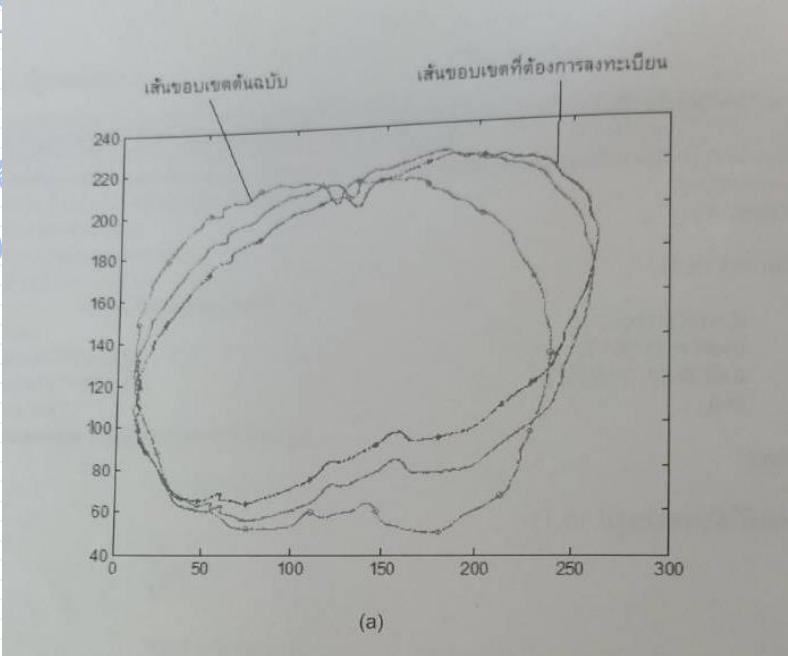
```
function [out]=subsample(in,rate)
m=size(in);
count=0;

for i=1:m(1)

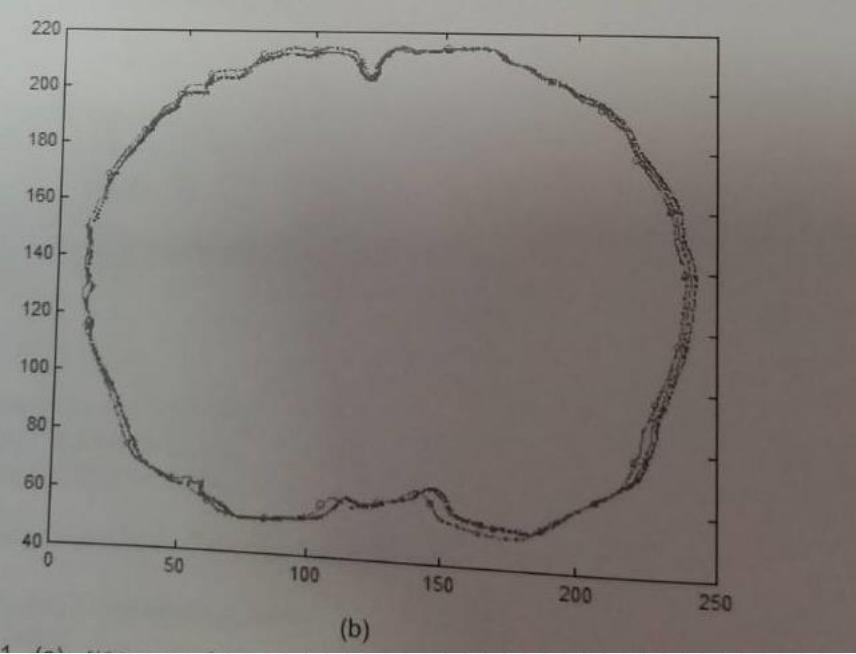
    if rem(i,rate)==0
        count=count+1;
        out(count,:)=in(i,:);
    end

end
```

รูป 10.11



(a)



(b)

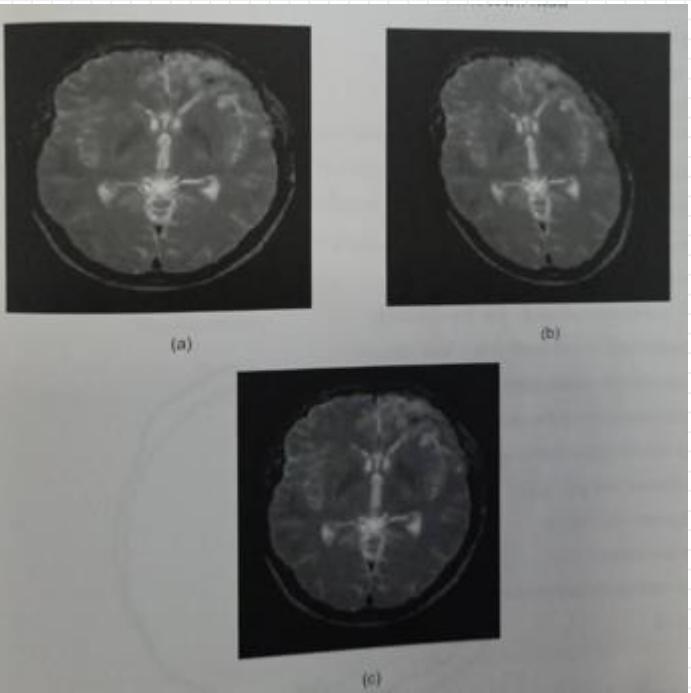
รูป (a) ผลจากการทำ ICP รอบที่ 1 เส้นโครงร่างที่มีแลนมาრ์ค 0 เป็นเส้นโครงร่างตันฉบับ,
เส้นโครงร่างที่มีแลนมาร์ค x เป็นเส้นโครงร่างที่ต้องการซ้อนทับภาพ และเส้นโครงร่างที่ไม่มีแลน
มาร์คเป็นเส้นโครงร่างที่ได้จากการแปลงเส้นโครงร่างที่ต้องการซ้อนทับภาพ;

(b) ผลจากการทำ ICP รอบที่ 5

H_2O

ตัวอย่างที่ 10.3

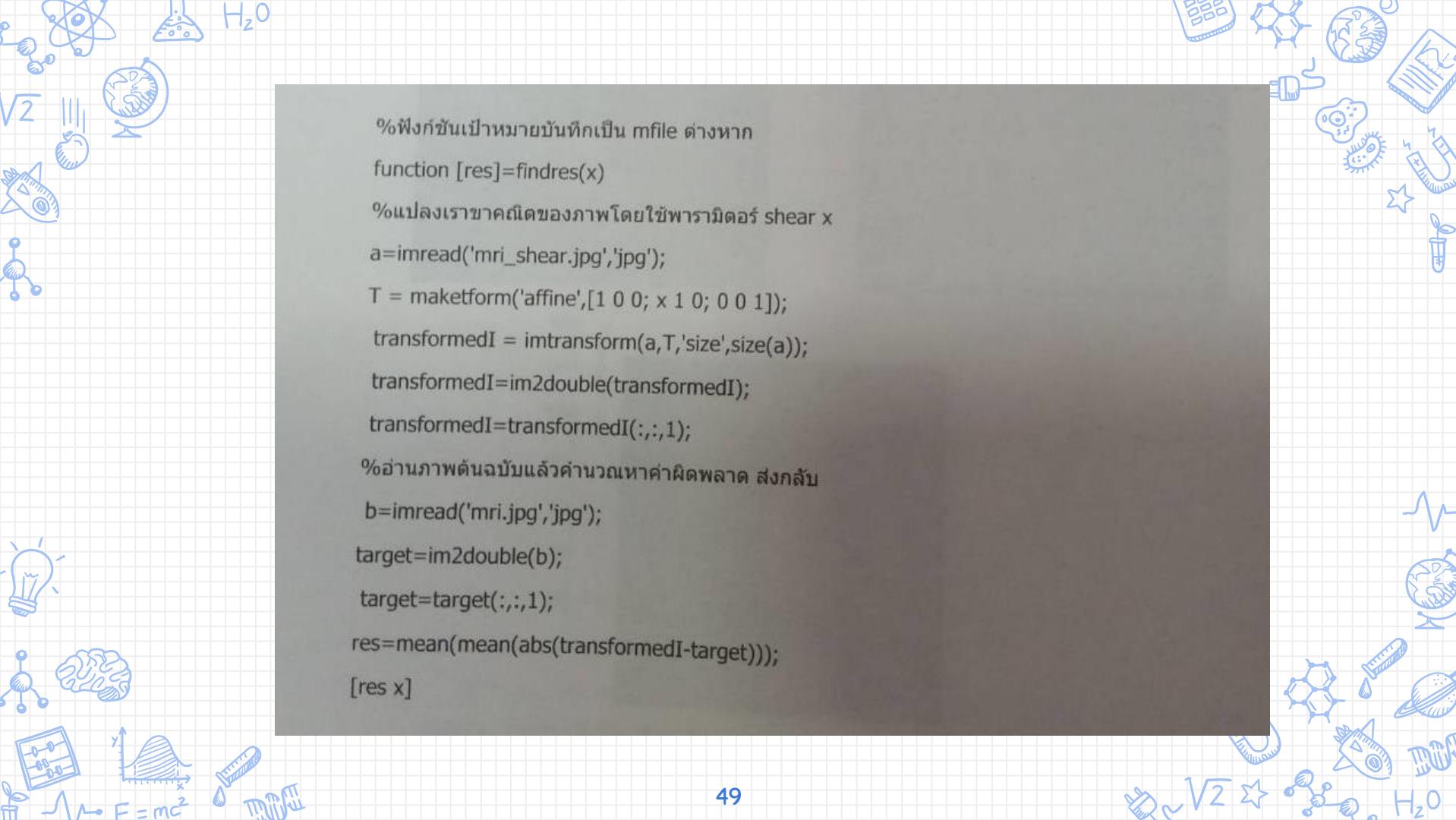
ซ่อนหัวภาพ MRI.jpg เข้ากับภาพ MRI_shear.jpg โดยใช้วิธีพิจารณาความเหมือนกันของภาพ เราทราบว่าภาพ MRI_shear.jpg ถูกแปลงเรขาคณิตแบบเดือน ดังแสดงในรูป 10.12 (b) เราจะใช้คำสั่ง fminsearch Matlab ในการค้นหารากมิเตอร์ ของการแปลงเรขาคณิตแบบเดือน คำสั่ง fminsearch ต้องระบุฟังก์ชันที่ใช้ค้นหาจุดต่ำสุดของฟังก์ชัน ในตัวอย่างนี้คือฟังก์ชัน findres ซึ่งค่าฟังก์ชันคำนวณได้จากการต่างระหว่างภาพเดิมกับการแปลงเรขาคณิตของภาพที่ต้องการซ่อนหัวภาพด้วยเมทริกซ์การแปลงที่มีค่ารามิเตอร์ที่ต้องการหา ซึ่งเราถือว่าถ้าภาพเหมือนกันผลต่างจะมีค่าน้อย



รูป 10.12 (a) ภาพต้นฉบับ; (b) ภาพที่ต้องการซ้อนทับกับภาพต้นฉบับ(ซึ่งถูกแปลงแบบเดือน);
(c) ภาพ (b) หลังจากการซ้อนทับภาพ



```
clear  
[x,r]=fminsearch('findres',0.5);  
%แสดงภาพผลลัพธ์  
I=imread('mri_shear.jpg','jpg');  
T = maketform('affine',[1 0 0; x 1 0; 0 0 1]);  
transformedI = imtransform(I,T,'size',size(I));  
figure(1);image(transformedI)  
a=imread('mri.jpg','jpg');  
figure(2),image(a);
```



```
%ฟังก์ชันเป้าหมายบันทึกเป็น mfile ดังนี้  
function [res]=findres(x)  
%แปลงเรขาคณิตของภาพโดยใช้พารามิเตอร์ shear x  
a=imread('mri_shear.jpg','jpg');  
T = maketform('affine',[1 0 0; x 1 0; 0 0 1]);  
transformedI = imtransform(a,T,'size',size(a));  
transformedI=im2double(transformedI);  
transformedI=transformedI(:,:,1);  
%อ่านภาพต้นฉบับแล้วคำนวณหาค่าผิดพลาด ส่งกลับ  
b=imread('mri.jpg','jpg');  
target=im2double(b);  
target=target(:,:,1);  
res=mean(mean(abs(transformedI-target)));  
[res x]
```

ใน 6 ลูปสุดท้ายเราได้ว่าค่าผิดพลาดที่ต่ำสุดคือ 0.0746 และค่าพารามิเตอร์ shear คือ -0.003

ans =

0.0746 -0.0035

ans =

0.0746 -0.0029

ans =

0.0746 -0.0029

ans =

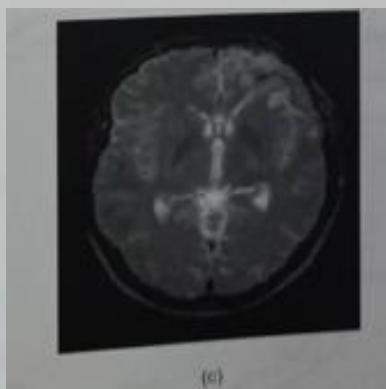
0.0746 -0.0033

ans =

0.0746 -0.0030

ans =

0.0746 -0.0030

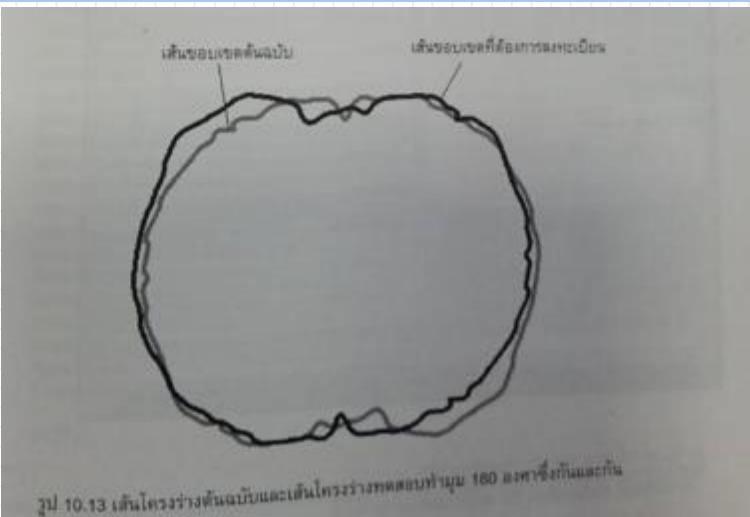


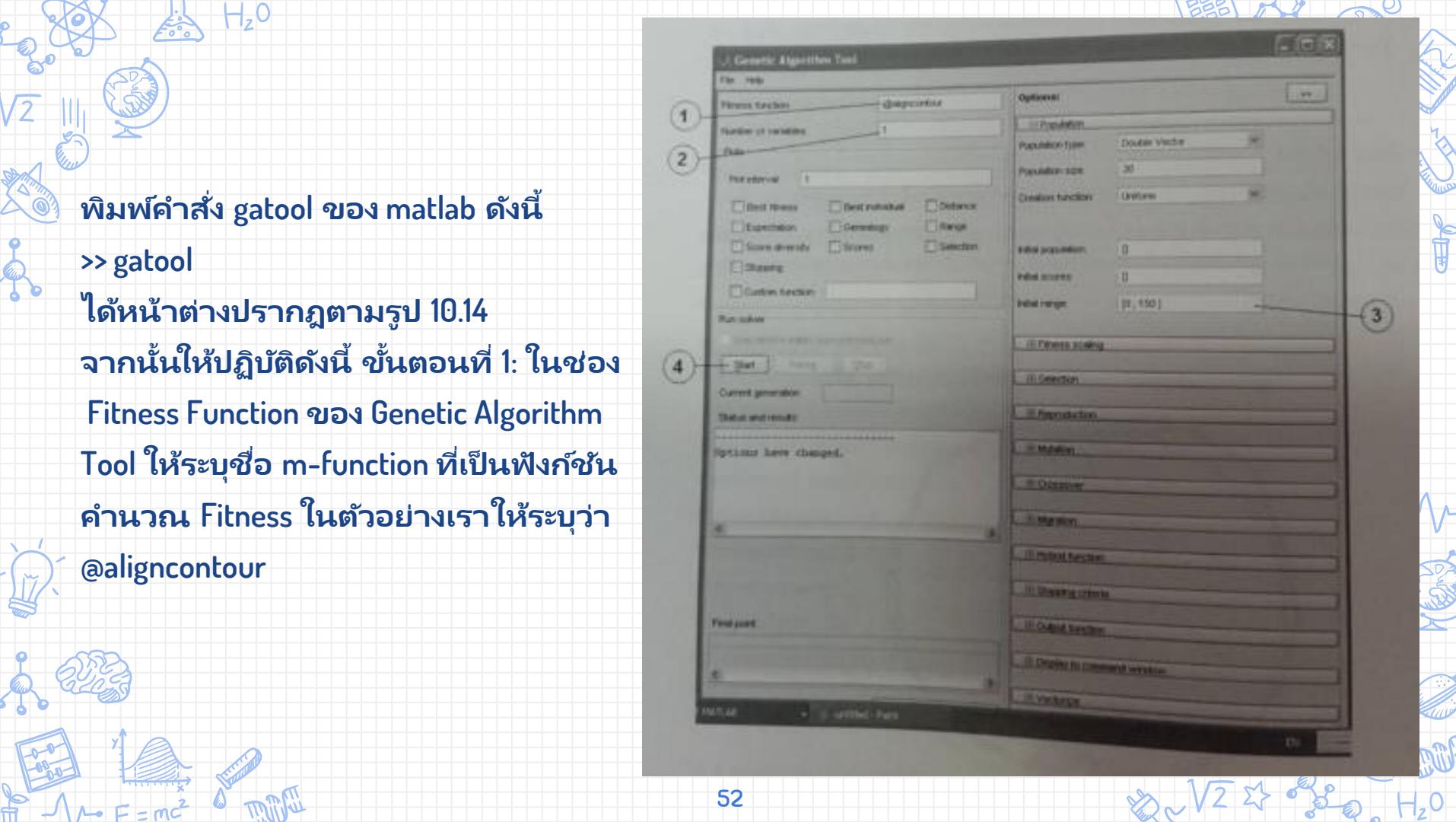
ผลลัพธ์ภาพเอาท์พุทหลังการแปลงด้วยเมทริกซ์ที่มีค่าพารามิเตอร์ที่หาได้แสดงในรูป 10.12
(c) ซึ่งตรงกับภาพด้านบน

H_2O

ตัวอย่างที่ 10.4

ในตัวอย่างนี้เราจะใช้ Gaitool ของ Matlab ในการซ้อนทับภาพสองภาพ สมมติว่า เราต้องการซ้อนทับภาพเล็บโครงร่างของภาพถ่ายลมอง เราทำการหมุนเล็บโครงร่างตันฉบับไป 180 องศาได้เป็นเล็บโครงร่างใหม่ เราเรียกเล็บโครงร่างนี้ว่า เล็บโครงร่างทดสอบ เล็บโครงร่างตันฉบับและเล็บโครงร่างทดสอบแสดงในรูป 10.13



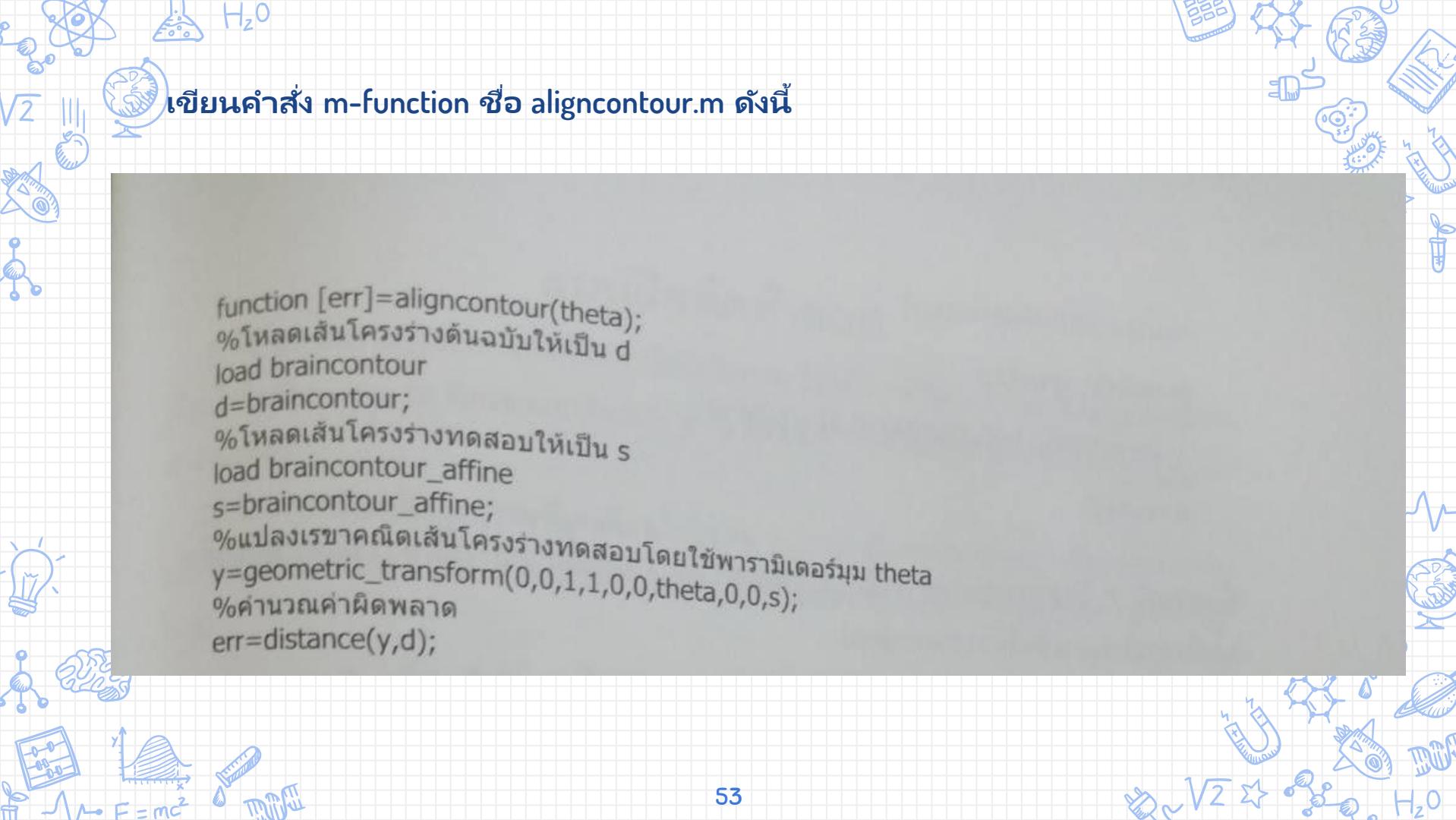


พิมพ์คำสั่ง gatool ของ matlab ดังนี้

>> gatool

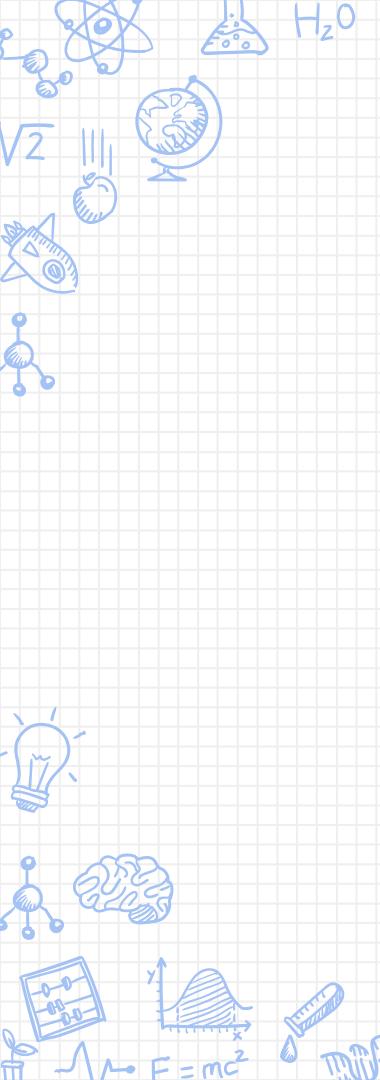
ได้หน้าต่างประมาณรูป 10.14

จากนั้นให้ปฏิบัติตามนี้ ขั้นตอนที่ 1: ในช่อง Fitness Function ของ Genetic Algorithm Tool ให้ระบุชื่อ m-function ที่เป็นฟังก์ชันคำนวณ Fitness ในตัวอย่างเราให้ระบุว่า @aligncontour



เขียนคำสั่ง m-function ชื่อ aligncontour.m ดังนี้

```
function [err]=aligncontour(theta);
%โหลดเส้นโครงร่างตันจะมันให้เป็น d
load braincontour
d=braincontour;
%โหลดเส้นโครงร่างทดสอบให้เป็น s
load braincontour_affine
s=braincontour_affine;
%แปลงเรขาคณิตเส้นโครงร่างทดสอบโดยใช้พารามิเตอร์หมุน theta
y=geometric_transform(0,0,1,1,0,0,theta,0,0,s);
%คำนวณค่าผิดพลาด
err=distance(y,d);
```



%ฟังก์ชัน geometric_transform ทำการแปลงเรขาคณิตแบบ Affine ของเส้นโครงร่าง
%อินพุท x มีขนาด $n \times 3$ เอ้าที่พุท y มีขนาด $n \times 3$ บันทึกเป็น mfile ต่างหาก

```
function [y]=geometric_transform(shx,shy,sx,sy,tx,ty,theta,xc,yc,x)
%shearing
shear=eye(3,3);
shear(1,2)=shx;
shear(2,1)=shy;
%scaling
scale=eye(3,3);
scale(1,1)=sx;
scale(2,2)=sy;
%rotation
tplus=eye(3,3);
tminus=eye(3,3);
rotation=eye(3,3);
tplus(1,3)=xc;
tplus(2,3)=yc;
tminus(1,3)=-1*xc;
tminus(2,3)=-1*yc;
rotation(1,1)=cos(theta*pi/180);
rotation(1,2)=-1*sin(theta*pi/180);
rotation(2,1)=sin(theta*pi/180);
rotation(2,2)=cos(theta*pi/180);
%translation
translate=eye(3,3);
translate(1,3)=tx;
translate(2,3)=ty;
%combined;
T=shear*scale*tplus*rotation*tminus*translate;
```

$y=T*x';$

$y=y';$
%ฟังก์ชัน distance คำนวณระยะห่างเส้นโครงร่าง x และ y ซึ่งถือเป็นค่าfitness บันทึกเป็น mfile

%ต่างหาก

H_2O

```

function [di]=distance(x,y)

m=size(x); n=m(1);
for i=1:m
    d=sqrt((x(i,1)-y(i,1))^2+(x(i,2)-y(i,2))^2);
end
di=sum(d)

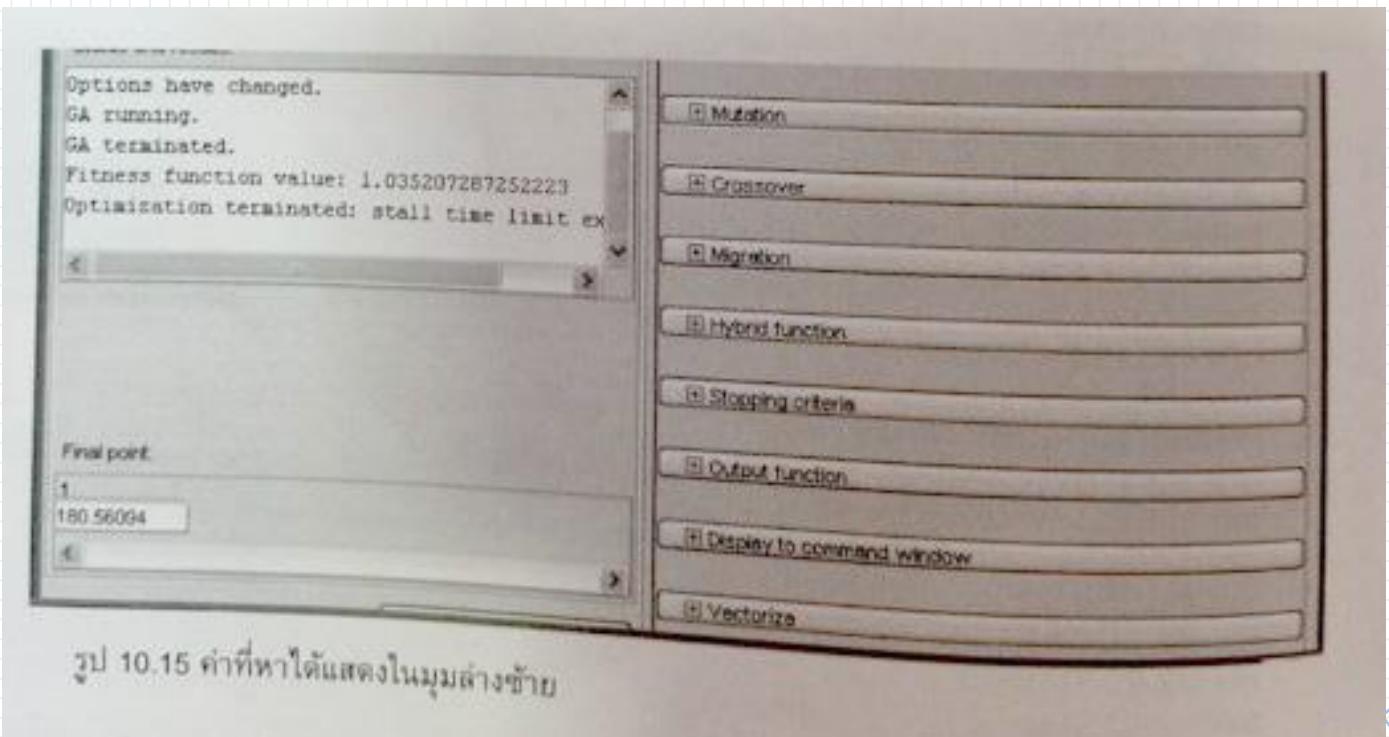
```

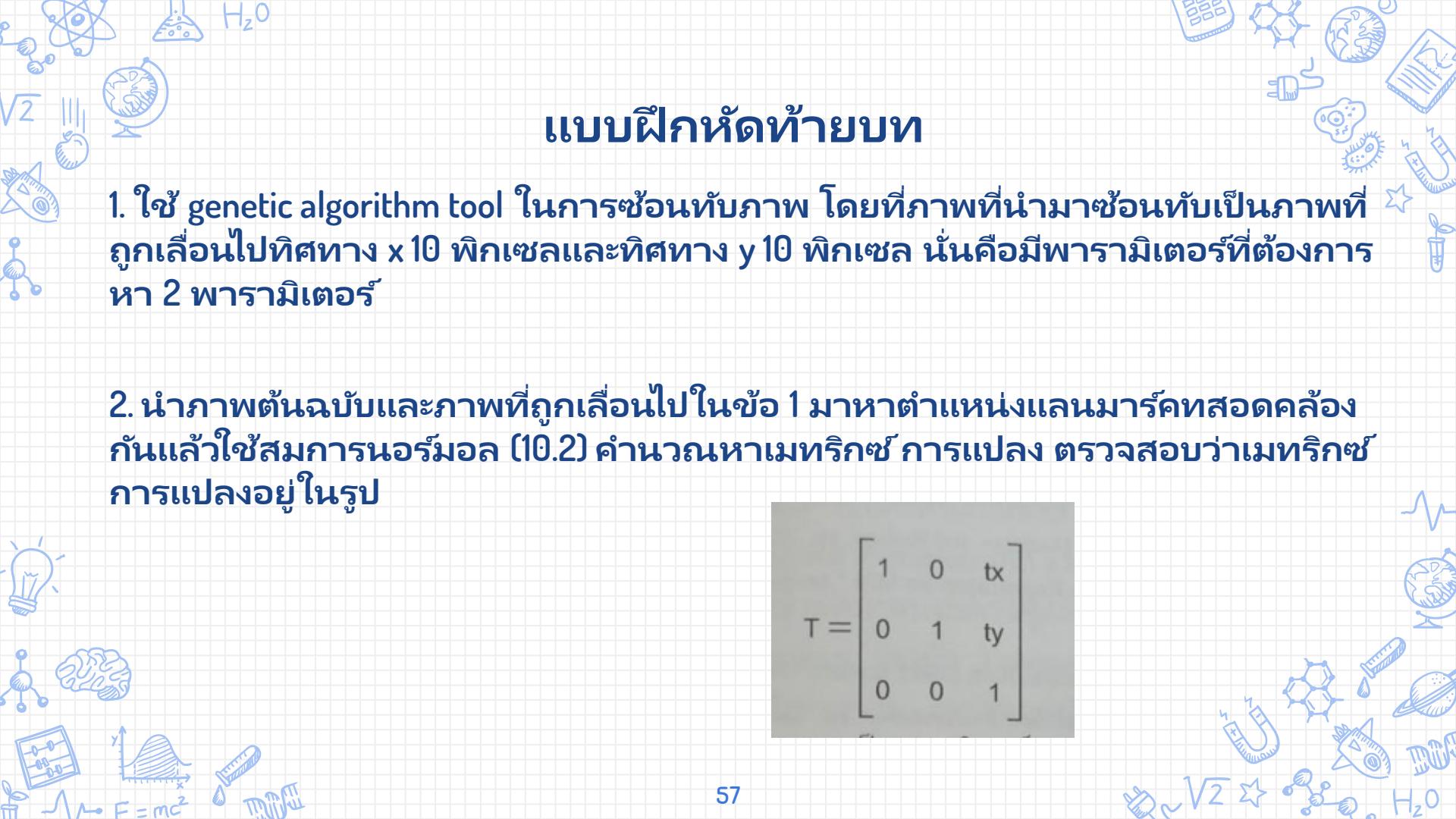
ขั้นตอนที่ 2: ใส่จำนวนของพารามิเตอร์ที่ต้องการหา ในกรณีตัวอย่างนี้มีพารามิเตอร์เดียวคือค่าคุณที่หมุนไปของเส้นโค้งร่างทดสอบ

ขั้นตอนที่ 3: กำหนดช่วงของค่าเริ่มต้นในโค้ดromeเป็น [0:150] ให้ใกล้เคียงกับค่าพารามิเตอร์ที่ต้องหา

ขั้นตอนที่ 4: กดปุ่ม start เพื่อเริ่มทำงาน

เมื่อ Genetic Algorithm ทำงานครบรอบที่ตั้งไว้ ค่าตอบที่หาได้จะแสดงในมุมล่างซ้าย ดังรูป 10.15





แบบฝึกหัดท้ายบท

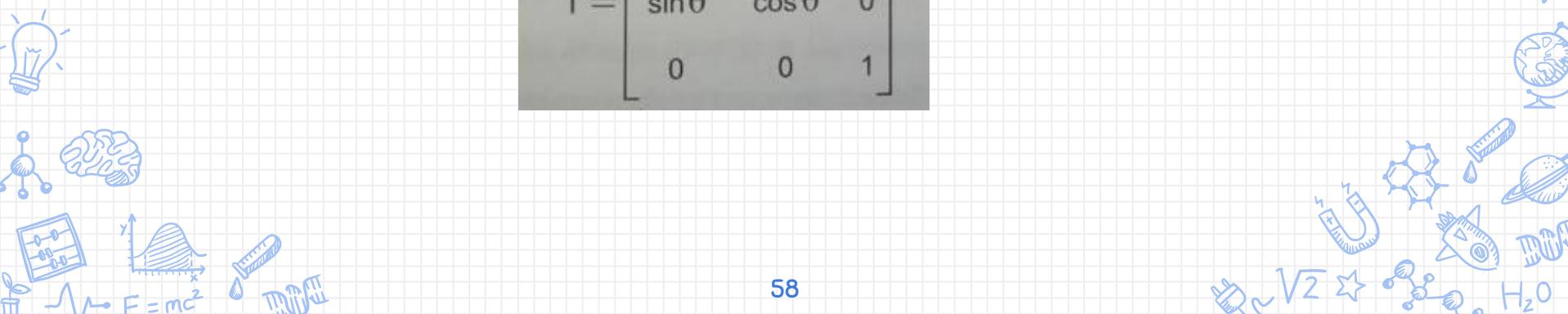
1. ใช้ genetic algorithm tool ในการซ่อนหันภาพ โดยที่ภาพที่นำมาซ่อนหันเป็นภาพที่ถูกเลื่อนไปทิศทาง $x \times 10$ พิกเซลและทิศทาง $y \times 10$ พิกเซล นั่นคือมีพารามิเตอร์ที่ต้องการหา 2 พารามิเตอร์
2. นำภาพต้นฉบับและภาพที่ถูกเลื่อนไปในข้อ 1 มาหาตำแหน่งแลนมาร์คที่สอดคล้องกันแล้วใช้สมการนอร์มอล (10.2) คำนวณหาเมทริกซ์ การแปลง ตรวจสอบว่า เมทริกซ์ การแปลงอยู่ในรูป

$$T = \begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix}$$



3. นำภาพต้นฉบับ และ ภาพต้นฉบับที่ถูกหมุนไป มาหาตำแหน่งแลนมาร์คที่ สอดคล้องกันแล้วใช้สมการนอร์มอล (10.2) คำนวนหาเมทริกซ์การแปลง ตรวจสอบว่าเมทริกซ์การแปลงอยู่ในรูป

$$T = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



4. นำภาพต้นฉบับและภาพต้นฉบับที่ถูกขยายในทิศทาง x 2 เท่าและหดในทิศทาง y 0.8 เท่า มาหาตำแหน่งแลนมาร์คที่สอดคล้องกันแล้วใช้สมการนอร์มอล (10.2) คำนวณหาเมทริกซ์ การแปลง ตรวจสอบว่าเมทริกซ์การแปลงอยู่ในรูป

$$T = \begin{bmatrix} sx & 0 & 0 \\ 0 & sy & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

โดยที่ sx และ sy เป็นพารามิเตอร์ของการสเกล

5. ทำการซ่อนหันภาพในข้อ 2, 3 และ 4 โดยใช้เมทริกซ์การแปลงที่หาได้กับการแปลง เเรขาคณิตของภาพในบทที่ 5