



บทที่ 2

ทฤษฎีการรับรู้ลายมือเขียน

ทฤษฎีเบื้องต้น

การวิจัยทางด้านการรับรู้ลายมือเขียนมีมานานกว่าสองทศวรรษ ตามที่ได้มีการ
เล็งเห็นถึงประโยชน์ในการนำมาประยุกต์ใช้กับงานทางธุรกิจเช่น ระบบสำนักงานอัตโนมัติ
(Office Automation) (HIRANVANICHKORN et al. 1984) โดยอาศัยหลักการ
รับรู้รูปแบบ

จากการรวบรวมงานวิจัยทางด้านการรับรู้รูปแบบตัวอักษร (Character-
Pattern Recognition) ซึ่งมีการวิจัยในภาษาต่างๆ สามารถแบ่งงานวิจัยโดยอาศัย
แหล่งที่มาของตัวอักษรออกเป็น

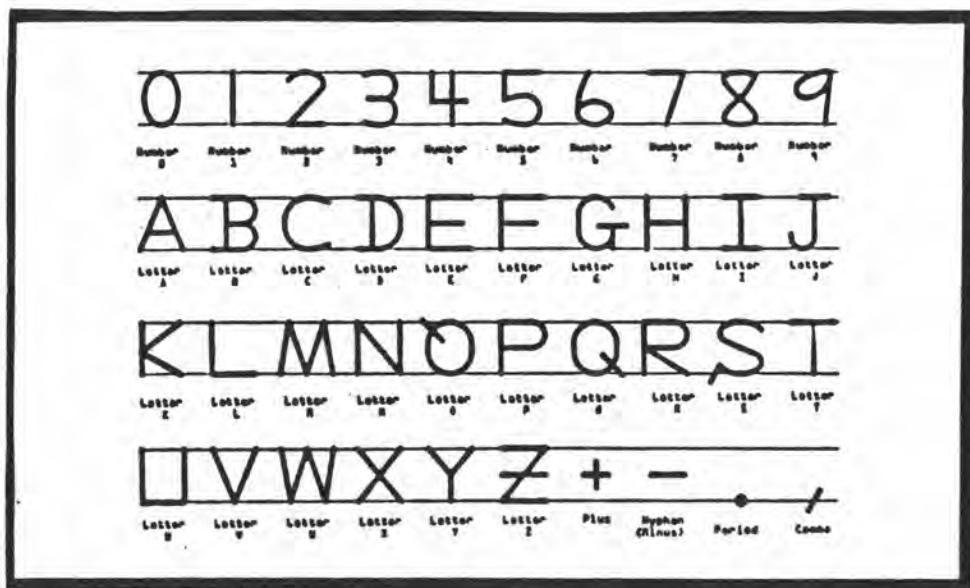
1. การรับรู้รูปแบบอักษรพิมพ์ (Printed Characters Recognition)

รูปแบบของอักษรชนิดนี้ได้มาจากเครื่องพิมพ์ ซึ่งรูปแบบตัวอักษรที่ได้มาจาก
เครื่องพิมพ์มีความแตกต่างเพียงเล็กน้อย ทำให้การรับรู้มีประสิทธิภาพสูง และถ้ามีการเก็บ
ข้อมูลรูปแบบอักษรอ้างอิงเป็นจำนวนเพียงพอ สามารถทำให้ประสิทธิภาพการรับรู้ถึงจุดที่
อาจไม่มีความผิดพลาดในการรับรู้เลย

2. การรับรู้รูปแบบอักษรลายมือเขียน (Handwritten Characters Recognition)

รูปแบบอักษรจะได้รับการเขียนทั่วไป การเขียนไม่สามารถกำหนดรูปแบบ
อักษรที่แน่นอนเหมือนตัวอักษรที่ได้จากการพิมพ์ แม้ว่าการเขียนตัวอักษรจากบุคคลคนเดียว
กันก็จะได้อักษรที่มีลักษณะแตกต่างกันออกไปได้ ด้วยเหตุนี้ขั้นตอนในการรับรู้รูปแบบอักษร
ลายมือเขียนจึงมีความซับซ้อนมาก และมีผลกระทบโดยตรงต่อความแม่นยำในการรับรู้รูป

แบบอักษร การวิจัยในระยะแรกๆ ได้มีการกำหนดมาตรฐานของตัวอักษรให้ผู้เขียนใช้เป็นต้นแบบในการเขียน ซึ่งสามารถจำกัดความแตกต่างของรูปแบบตัวอักษรที่เขียนได้ เป็นผลให้การรับรู้รูปแบบดีขึ้นจากรูปที่ 2.1 แสดงถึงรูปแบบอักษรมาตรฐานของ ANSI (American National Standards Institute) สำหรับเครื่องอ่านลายมือเพื่อเพิ่มความสามารถในการรับรู้ลายมือเขียน (WARD AND BLESSER, 1985)



รูปที่ 2.1 รูปแบบอักษรมาตรฐาน ANSI

ผลกระทบจากการใช้รูปแบบอักษรมาตรฐาน ทำให้ความเร็วในการเขียนลดลง เนื่องจากความไม่เคยชิน และเมื่อมีการเขียนเป็นเวลานานพบว่ารูปแบบอักษรที่เขียนเปลี่ยนไปจากรูปแบบอักษรมาตรฐาน เป็นรูปแบบอักษรทั่วไปที่เคยชินในการเขียน ด้วยเหตุนี้ จึงมีการนำวิธีการของรูปแบบอักษรมาตรฐานมาประยุกต์เป็นระบบการรับรู้รูปแบบลายมือเขียนของคนๆ เดียว กล่าวคือมีรูปแบบอักษรมาตรฐานตามลายมือของคนเขียน เพื่อให้มีความแม่นยำในการรับรู้ที่คล้ายกับวิธีการของการรับรู้รูปแบบอักษรมาตรฐาน ควบคู่กับความเคยชินในการเขียนของผู้ใช้

อัลกอริทึมที่นำมาประยุกต์ในการรับรู้รูปแบบแบ่งออกได้เป็น (HOR, 1985; YAMAMOTO AND MORI, 1988)

1. การวิเคราะห์แบบโกลบอล (Global Analysis)

วิธีการนี้จะพิจารณารูปแบบพร้อมกันทั้งรูปแบบโดย ไม่มีการแยกส่วนใด เพื่อช่วยในการตัดสินใจ การตัดสินใจว่ารูปแบบทดสอบ กับรูปแบบอ้างอิงมีลักษณะคล้ายหรือแตกต่างกัน ทำได้โดยการประยุกต์ใช้ทฤษฎีทางสถิติ อัลกอริทึมสามารถแยกออกได้เป็น

1.1 การกระจายของจุด อัลกอริทึมนี้เป็นการตรวจสอบการกระจายของจุดที่ประกอบเป็นรูปแบบตัวอักษร โดยพิจารณาถึงตำแหน่งของจุด ความหนาแน่นของจุดตามพื้นที่ส่วนต่างๆ ของรูปแบบ ระยะห่างระหว่างจุด และจุดตัด

1.2 การแปลง (Transformation) เป็นการใช้อัลกอริทึมทางคณิตศาสตร์ เช่น Fourier Transformation

1.3 การตรวจวัดรูปแบบ อัลกอริทึมนี้เป็นการตรวจวัดตามลักษณะรูปแบบทั้งรูปแบบ

2. การวิเคราะห์แบบโครงสร้าง (Structural or Syntactic or Local Feature Analysis)

วิธีการนี้ได้รับความสนใจเป็นอย่างมากในปัจจุบัน เพราะมีขีดความสามารถในการตรวจรู้คุณลักษณะตามโครงสร้าง ซึ่งวิธีการทางสถิติยังขาดอยู่ อัลกอริทึมสามารถแยกออกตามลักษณะที่สามารถตรวจรู้

2.1 ขอบ และส่วนของเส้น

2.2 เส้นรอบรูป

2.3 เส้นกึ่งกลางที่ได้จากการลดขนาดเส้น

ในวิทยาการนี้ฉบับนี้เลือกวิธีการไดนามิกโปรแกรมมิ่ง (Dynamic Programming) ในการวิเคราะห์รูปแบบทั้งรูปแบบตามการวิเคราะห์แบบโกลบอล เพื่อ

ตรวจสอบความแตกต่างของรูปแบบ และเลือกอัลกอริทึมรหัสทิศทางแบบลูกโซ่ของฟรีแมน (Freeman's Directional Chain Code) ช่วยในกระบวนการวิเคราะห์ลักษณะเฉพาะส่วนตามการวิเคราะห์แบบโครงสร้าง โดยมีการเก็บรวบรวมรูปแบบอ้างอิงเฉพาะบุคคล หรือต่างบุคคลไว้ในพจนานุกรมรูปแบบ

โดยพื้นฐาน ไดนามิค โปรแกรมมิ่งออกแบบมาสำหรับการแก้ไขปัญหาซับซ้อนที่มีกฎเกณฑ์ดังต่อไปนี้ (TANA, 1976)

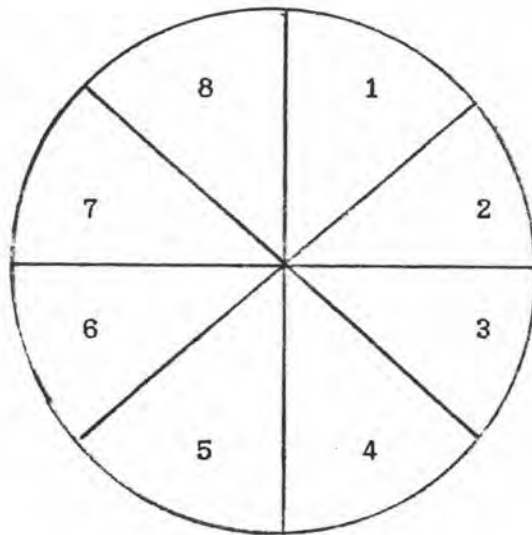
1 ปัญหาสามารถแยกออกเป็นปัญหาย่อย แต่ละปัญหาย่อยสามารถเลือกวิธีการที่เหมาะสมวิธีเดียวจากทางเลือกหลายๆ วิธี โดยไม่จำเป็นต้องทราบถึงเงื่อนไขหรือปัญหาย่อยอื่นๆ ล่วงหน้า

2 ใช้เพียงแต่วิธีการที่ดีที่สุดสำหรับปัญหาย่อยเท่านั้น จะไม่นำวิธีการที่เป็นทางเลือกอื่นๆ มาใช้

3 เชื่อมต่อปัญหาย่อยเข้าด้วยกันด้วยวิธีการที่เหมาะสมเพื่อให้การแก้ไขปัญหาเป็นไปอย่างถูกต้อง

ทฤษฎีไดนามิค โปรแกรมมิ่ง เริ่มนำมาใช้โดย Bellman (Bellman, 1975) เพื่อใช้แก้ปัญหาทางด้านคณิตศาสตร์ ซึ่งมีการประมวลการตัดสินใจหลายขั้นตอน ต่อมาได้นำมาประยุกต์ใช้ทางด้านวิทยาศาสตร์คอมพิวเตอร์อย่างแพร่หลาย ในปี 1967 ได้มีการพิมพ์อัลกอริทึม VITERBI และทฤษฎีการประมวลผลของ MARKOV ซึ่งได้รวมทฤษฎีและอัลกอริทึมทั้งสองไว้ได้มีศึกษามานานกว่า 50 ปี นักวิจัยทางด้านเสียงพูดทั่วไปก็มีการอ้างอิงถึงทฤษฎีไดนามิค โปรแกรมมิ่งนี้มาเป็นเวลานานจะเห็นได้จาก VELICHKO และ ZAGORYKO ชาวรัสเซีย ตีพิมพ์การวิจัยในปี 1970 SAKOE และ CHIBA ชาวญี่ปุ่น ตีพิมพ์การวิจัยในปี 1974 QUERRE, GRESSER และ BARS ชาวฝรั่งเศส ตีพิมพ์การวิจัยในปี 1973 BRIDLE ชาวอังกฤษ ทำการวิจัยปี 1974 แต่ไม่มีการตีพิมพ์งานวิจัย BAKER, ITAKURA และ JELINEK แห่งสหรัฐอเมริกา ต่างได้มีการวิจัยในปี 1974

อัลกอริทึมรหัสทิศทางแบบลูกโซ่ของฟรีแมนเป็นเครื่องมือช่วยในการแยกลักษณะเด่นของรูปแบบออกเป็นพีเซอร์ พีเซอร์ที่ได้นั้นเป็นค่ารหัสทิศทางการลากเส้นต่อเวลา โดยค่ารหัสทิศทางสามารถกำหนดได้จากการหักเหของทิศทางตามเส้นที่ลากตามรูปที่ 2.2 การเปลี่ยนแปลงรหัสทิศทางในช่วงเวลาหนึ่งนั้นสามารถใช้แทนเป็นรูปแบบที่บอกถึงลักษณะของเส้นที่ลาก และความแตกต่างจากเส้นอื่นๆ (BEUS AND TIU, 1987; HUANG AND CHUANG, 1986; KERRICK AND BOVIK, 1988; YAMAMOTO, 1988) ประโยชน์ที่ได้จากรูปแบบรหัสทิศทางนี้สามารถช่วยในการจำแนกกลุ่มรูปแบบ ทำให้ขอบเขตในการรับรู้รูปแบบอยู่ภายในวงจำกัดลง ทั้งยังเป็นรูปแบบในการคำนวณหาความแตกต่างตามทฤษฎีไดนามิก โปรแกรมมิ่งอีกด้วย



รูปที่ 2.2 แสดงค่าความหักเหทิศทาง

ฟีเชอร์

วิธีการรับรูปแบบมีหลายวิธี การเทียบแม่แบบเป็นวิธีการที่ง่ายที่สุดในการรับรูปร่างแบบ วิธีการนี้จะทำการเก็บข้อมูลเป็นเซตของแม่แบบไว้ นำรูปแบบทดสอบมาเทียบกับแม่แบบแต่ละแม่แบบ การตัดสินใจว่ารูปแบบทดสอบคล้ายกับแม่แบบใดมากที่สุดนั้นขึ้นกับกฎเกณฑ์ความเหมือน (Matching Criterion หรือ Similarity Criterion) โดยปรกติถ้าสร้างเป็นระบบแบบง่าย แม่แบบจะเก็บในลักษณะเป็นข้อมูลดิบ วิธีการนี้ใช้กันมากในการรับรูปร่างแบบตัวอักษรพิมพ์ และการรับรู้ตัวเลขบนเค็ดของเครื่องอ่านเค็ดขนาดคาร์ ข้อเสียของวิธีการนี้ถ้ารูปแบบมีการแปรปรวนมากการเลือกข้อมูลต้นแบบที่ดีจะยากมาก (พีเชษฐอุดมประเสริฐ, 2529; HOR, 1985)

วิธีการที่มีความซับซ้อนมากขึ้น คือการจำแนกกลุ่มหรือการรับรูปร่างแบบขึ้นอยู่กับเซตของการเลือกคุณลักษณะเด่นบางส่วนจากรูปแบบ เรียกว่าฟีเชอร์ ฟีเชอร์ที่ตัดทอนออกมาไม่ควรจะมีการแปรปรวนมาก หรือมีผลกระทบน้อยต่อการแปรปรวน หรือการขาดหายไปของคุณลักษณะนั้น และต้องให้มีส่วนที่ซ้ำซ้อนน้อยที่สุด ตามที่ได้กล่าวมาแล้วนั้นสามารถแยกระบบการรับรูปร่างแบบออกได้เป็น 2 ส่วน

1. การเลือกคุณลักษณะเด่นของรูปแบบ

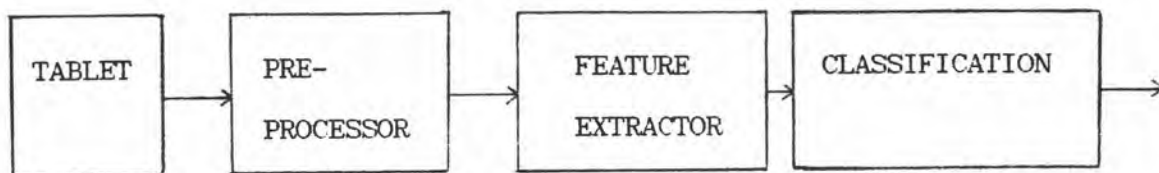
การตัดสินใจเลือกคุณลักษณะสำคัญของรูปแบบนั้นไม่สามารถกำหนดได้แน่นอน จะขึ้นอยู่กับสถานะการณีนั้น ประสบการณ์ และการทดลอง ทั้งนี้เนื่องมาจากมีทฤษฎีสันนิษฐานน้อยเกี่ยวกับการเลือกฟีเชอร์ แต่อย่างไรก็ตามมีการทดสอบเกี่ยวกับฟีเชอร์และลำดับของฟีเชอร์ที่เลือก สถานะของฟีเชอร์หรือลำดับของฟีเชอร์ที่เลือกโดยมากจะขึ้นอยู่กับความสำคัญของฟีเชอร์ที่แสดงถึงลักษณะของรูปแบบ หรือผลที่ได้ของฟีเชอร์ต่อประสิทธิผลของการรับรูปร่างแบบ

2. วิธีการรับรูปร่างแบบ

การเลือกอัลกอริทึมในการรับรูปร่างแบบต้องให้เหมาะสม เพื่อให้สามารถประยุกต์

พีเชอร์ที่ได้ผ่านการตรวจสอบให้เกิดประโยชน์สูงสุด การรับรู้รูปแบบจะคำนวณหาความแตกต่างระหว่างรูปแบบทดสอบ และรูปแบบอ้างอิง

ผังของระบบรับรู้รูปแบบตามรูปที่ 2.3 แสดงถึงส่วนที่ทำการนำพีเชอร์ออกจากรูปแบบเรียกว่า Feature Extractor และส่วนที่ทำหน้าที่ของการรับรู้รูปแบบ หรือการจำแนกกลุ่มเรียกว่า Classifier



รูปที่ 2.3 ผังงานระบบรับรู้รูปแบบ

การตรวจหาความหักเหของการลากเส้น

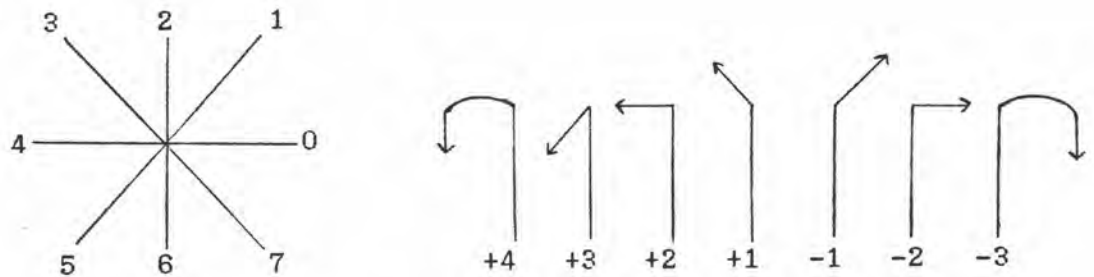
รหัสทิศทางลูกโซ่ฟรีแมนกำหนดโดยลูกโซ่ $\{ a(i)_1 \}$ เมื่อ $a(i) \in \{0, \dots, 7\}$ ตามรูปที่ 2.4 แสดงถึงรหัสทิศทางแทนการหักเหของเส้นที่ลาก วิธีการของฟรีแมนนี้เป็นการใช้เวกเตอร์แต่ละส่วน L_j เชื่อมต่อกันจาก a_{j-s+1} ถึง a_j ตามรหัสทิศทางลูกโซ่ที่ต่อเนื่อง ความยาว D_j ของ L_j จะอยู่ในช่วงของ $S \sqrt{2}$ ถึง 1 เสมอ ความยาวสามารถคำนวณได้จากสมการที่ 2.1 (BEUS, 1987)

$$D_j = [(X_j)^2 + (Y_j)^2]^{1/2} \tag{2.1}$$

โดยที่

$$X_j = \sum_{i=j-s+1}^j a_{ix}$$

$$Y_j = \sum_{i=j-s+1}^j a_{iy}$$



รูปที่ 2.4 รหัสทิศทางแทนการหักเหของการลากเส้น

เมื่อ a_{1x} และ a_{1y} มี X และ Y เป็นองค์ประกอบของการเชื่อมรหัสลูกโซ่ $a_{1x}, a_{1y} \in \{-1, 0, 1\}$ ทิศทางระหว่างแกน X กับการเคลื่อนของการลากเส้น กำหนดให้เป็นส่วนของเส้น

$$\theta_j = \tan^{-1} (Y_j / X_j) \quad \text{ถ้า } |X_j| \geq |Y_j| \quad (2.2ก)$$

$$\theta_j = \cot^{-1} (X_j / Y_j) \quad \text{ถ้า } |X_j| < |Y_j| \quad (2.2ข)$$

ซึ่งมีทิศทางเคลื่อนอย่างอิสระ 0 ถึง 360 เส้น ณ จุดนั้นจะผ่านกระบวนการปรับเรียบได้ตามสมการที่ 2.3

$$d_j = \theta_{j+1} - \theta_{j-1} \quad (2.3)$$

การตรวจวัดมุม

การตรวจวัดมุมของฟรีแมน และเดวิส เป็นการคำนวณความยาวของส่วนรหัสทิศทางลูกโซ่ ส่วนของการเคลื่อนไปด้านหน้า หรือ ด้านหลัง ณ จุดนั้น ต่อมุมที่มีได้เชื่อมต่อของจุดนั้น ซึ่งการตรวจวัดมุม K_j ณ จุด j ได้จากสมการที่ 2.4

$$K_j = \ln(t_1) * \sum_{i=j}^{j+s} d_i * \ln(t_2) \quad (2.4)$$

โดย $t_1 = \max \{t: d_{j-v} (-\Delta, \Delta) \text{ สำหรับทุกค่า } v \text{ โดย } 1 \leq v \leq t\}$

$t_2 = \max \{t: d_{j+s-v} (-\Delta, \Delta) \text{ สำหรับทุกค่า } v \text{ โดย } 1 \leq v \leq t\}$

$$\Delta = \tan^{-1} (1 / (s - 1))$$

การประยุกต์ใช้ค่าเฉลี่ย K ซึ่งได้จากการคำนวณของค่าลำดับ s ณ จุดหนึ่งๆ และการกำหนดขอบเขตความยาวสูงสุดของการเคลื่อนไปทั้งด้านหน้า และด้านหลัง ซึ่งจะเป็นการจำกัดค่า t เพื่อให้ผลลัพธ์ที่ดีขึ้น ตามสมการที่ 2.5 N เป็นขอบเขตสูงสุดของ t_1 และ t_2 คำนวณค่า K_j ณ จุด j จะคำนวณ $s_2 - s_1$ ครั้งในแต่ละจุด และหาค่าเฉลี่ยของ K_j

$$N = C * F \quad C \text{ เป็นค่ายาว}$$

$$F \text{ เป็นส่วน } (0 < F \leq 1)$$

$$K_j = \left(\sum_{s=s_1}^{s_2} [\ln(t_1) * \sum_{i=j}^{j+s} d_i * \ln(t_2)] \right) / (s_2 - s_1) \quad (2.5)$$

โดย $t_1 = \max \{t: d_{j-v} (-\Delta, \Delta) \text{ สำหรับทุกค่า } v \text{ โดย } 1 \leq v \leq t$
และ $t < N\}$

$t_2 = \max \{t: d_{j+s+v} (-\Delta, \Delta) \text{ สำหรับทุกค่า } v \text{ โดย } 1 \leq v \leq t$
และ $t < N\}$

$$\text{และ } \Delta = \tan^{-1} (1 / (s - 1))$$

C เป็นจุดที่นับได้

F เป็นส่วนที่จำกัดขนาดความยาวของเส้น

M เป็นจำนวนจุดมากที่สุด

s1 s2 เป็นค่าต่ำสุด และสูงสุดของ s

การตรวจหาลักษณะเส้นส่วนโค้งเว้า

การตรวจหาลักษณะส่วนโค้งเว้า โดยการกวาดไปตามเส้นขอบเขตเพื่อหาการหักเหไปทางซ้าย หรือขวา ซึ่งสามารถแทนได้จากการลากเส้นตามเข็มนาฬิกาด้วยเครื่องหมาย - หรือการลากเส้นทวนเข็มนาฬิกาแทนด้วยเครื่องหมาย + เส้นขอบเขตแสดงให้เห็นถึงการหักเหตามรหัสทิศทางแบบลูกโซ่ของฟรีแมนที่ตรวจได้ เครื่องหมายจะใช้แทนในแต่ละคู่ลำดับที่ประกอบขึ้นแทนตัวอักษรตามทิศทางที่เปลี่ยนไปตามรูปที่ 2.5

-6	+	+2
-2	*	+2
-1	0	+1

+ ตำแหน่งก่อน

* ตำแหน่งปัจจุบัน

+	+2	+2
-2	*	+1
-2	-1	0

รูปที่ 2.5 ตารางแสดงเครื่องหมายแทนการหักเหของทิศทาง

$$P_n = (P_{x_n}, P_{y_n}) \quad : \quad n = 1, N$$

$$A_n \in (0, 1, 2)$$

P_n เป็นอิลิเมนต์ของตัวอักษรที่กวาดไปบนระนาบ X-Y N เป็นจำนวนอิลิเมนต์

A เป็นอิลิเมนต์ของเซตของเครื่องหมายสำหรับ P

0 เป็นอิลิเมนต์ที่ไม่ได้ใช้ในการหาคุณลักษณะโค้ง การหาคุณลักษณะโค้งจะทำได้โดย

$$C_k = (C_{x_k}, C_{y_k}) \quad : k = 1 - K$$

$$V_k \in \{ 1, 2 \}$$

C_k เป็นอิลิเมนต์ที่กวาดไปที่ไม่รวม $0 - K$ เป็นจำนวนอิลิเมนต์ที่ไม่รวมอิลิเมนต์ $0 - V$ เป็นอิลิเมนต์ของเซตเครื่องหมายสำหรับ C_k

ลักษณะเด่นของเส้นโค้งเว้า หรือหนูนั้น เป็นเส้นที่ลากของอักษรแต่ละเซกเมนต์ โดยปกติแล้วจะได้มากจากการกวาดไปตามเส้นขอบเขตในทิศทางตาม หรือทวนเข็มนาฬิกา การใช้เครื่องหมาย - หรือ + แทนคู่ลำดับที่เปลี่ยนไปของการลากเส้นสามารถทำได้ตามเงื่อนไข

1. กำหนดให้เครื่องหมาย + ที่จุด C_1 เป็นไปตามเงื่อนไขข้อหนึ่งข้อใด

$$1.1 \quad (V_1 > 0) \text{ และ } (V_{1-1} + V_1 + V_{1+1} > 0)$$

$$1.2 \quad (V_1 > 0) \text{ และ } (V_{1-1} + V_1 + V_{1+1} < 0)$$

$$\text{และ } (D_{1-1,1} \text{ และ } D_{1,1+1} > K_1)$$

เมื่อ $D_{i,j} = (C_{x_i} - C_{x_j})^2 + (C_{y_i} - C_{y_j})^2$ และ K_1 เป็นค่ากำหนดให้

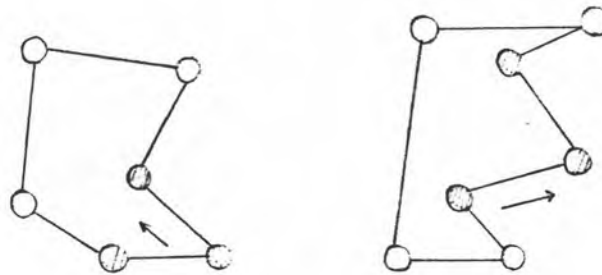
2. กำหนดให้เครื่องหมาย - ที่จุด C_1 เมื่อจุด C_1 เป็นไปตามเงื่อนไขข้อหนึ่งข้อใด

$$2.1 \quad (V_1 < 0) \text{ และ } (V_{1-1} + V_1 + V_{1+1} < 0)$$

$$2.2 \quad (V_1 < 0) \text{ และ } (V_{1-1} + V_1 + V_{1+1} > 0)$$

และ $(D_{1-1,1}$ และ $D_{1,1+1} > K_1)$

ตามรูปที่ 2.6 แสดงถึงตัวอย่างการใช้เครื่องหมาย + แทนคู่ลำดับที่เกิดการหักเหของเส้นที่ลากไปในทิศทางทวนเข็มนาฬิกา ตามเงื่อนไขที่ได้กำหนดไว้ข้างต้น



รูปที่ 2.6 การใช้เครื่องหมายแทนคู่ลำดับที่เกิดการหักเหของเส้น

เส้นโค้งนูน คือเส้นที่มีเครื่องหมาย + อยู่ระหว่าง 2 จุด - เส้นโค้งเว้า คือเส้นที่มีเครื่องหมาย - อยู่ระหว่าง 2 จุด +

ขนาดความยาวของเส้นโค้งตรวจวัดได้จากสมการที่ 2.6 ก และ ข

$$C_j = (C_{xj}, C_{yj}) \quad : \quad j = 1 - J$$

$$L = \sum_{j=1}^{J-1} D_{j,j+1}$$

$$L_v = D_{1,J}$$

$$C_o = L/L_v$$

$$G_x = \left(\sum_{j=1}^{J-1} (C_{x,j} + C_{x,j+1}) * D_{j,j+1} \right) / 2L \quad (2.6ก)$$

$$G_y = \left(\sum_{j=1}^{J-1} (C_{y,j} + C_{y,j+1}) * D_{j,j+1} \right) / 2L \quad (2.6ข)$$

รหัสทิศทางแบบลูกโซ่ของฟรีแมนกับวิธีการวิเคราะห์โครงสร้างแบบทรี

การประยุกต์อัลกอริทึมรหัสทิศทางแบบลูกโซ่ของฟรีแมนในการจำแนกรูปแบบอักษรลายมือเขียน (HOR, 1985)

R : รากของโครงสร้างแบบทรีที่เก็บรหัสทิศทางของรูปแบบ

NP : ตำแหน่งจุดกลางของการหาทิศทางบนตารางข้อมูล

Q : จำนวนจุดที่เชื่อมต่อกัน

LL(200,11) : ขนาดของตารางข้อมูล

LL(Q,0) : ตำแหน่งจุดที่ต่อกัน

LL(Q,1-8) : รหัสทิศทางทั้งแปด

LL(Q,9), LL(Q,11) : กำหนดให้เป็น 0

LL(Q,10) : จำนวนทิศทางใน Q

1. เริ่มการตรวจหาจุดเริ่มต้น โดยเริ่มจากจุดบนซ้ายสุดของตารางข้อมูล จุดแรกที่พบว่ามีค่าไม่เท่ากับศูนย์จะเป็นจุดรากของการหาทิศทางรูปแบบ R

ให้ $P = 0$

$Q = 1$

2. ให้ $P = P + 1$

$NP = LL(P,0)$

ถ้า $NP = 0$ แล้ว ไปประมวลผลต่อข้อ 5

3. เริ่มประมวลผลจากค่า $I = 1$ จนถึง 8 สำหรับการตรวจหารหัสทิศทาง
ทั้งแปด

ถ้า ทิศทาง I มีค่าเป็นศูนย์ หรือตำแหน่งได้บันทึกไว้ใน LL แล้ว ให้
ตรวจหาจุดต่อไป

ไม่เช่นนั้น ให้ $Q = Q + 1$ บันทึกตำแหน่งลงใน $LL(Q, 0)$
บันทึกค่า Q ลงใน $LL(P, I)$
เพิ่มค่าหนึ่งใน $LL(P, 10)$

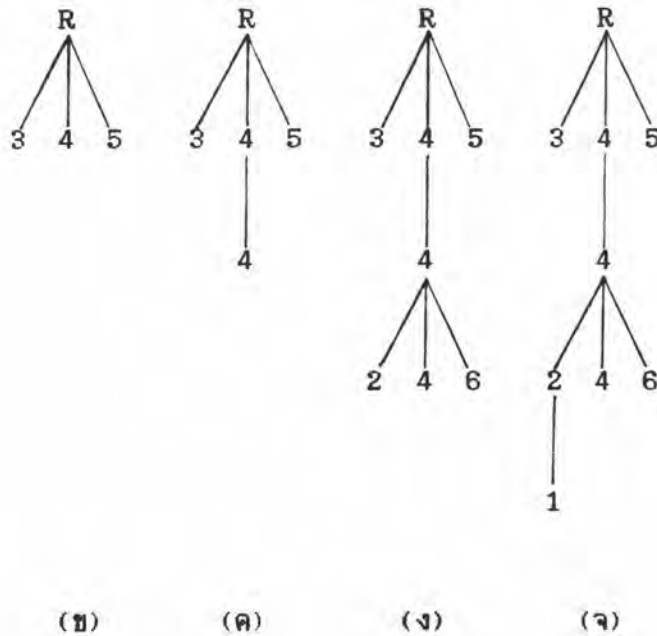
4. ไปประมวลผลที่ข้อ 2

5. หยุด

รูปที่ 2.7 แสดงถึงขั้นตอนการทำงานตามอัลกอริทึม โดยตำแหน่งราก R อยู่ที่
(2,2) ค่า $NP = 8$ จุดที่อยู่ต่อจากราก R ไปในทิศทางที่ 3 4 และ 5

0	0	0	0	0	0
0	1	1	0	1	0
0	1	1	0	0	0
0	0	0	1	0	0
0	0	1	0	1	0
0	0	0	0	0	0

รูปที่ 2.7 การทำงานของอัลกอริทึมรหัสทิศทางแบบลูกโซ่



รูปที่ 2.7 ต่อ

หลังจากที่หารหัสทิศทางแบบลูกโซ่ของฟรีแมนของแต่ละจุดบนตารางข้อมูลแล้ว
ขั้นตอนต่อไปเป็นการดึงลักษณะเด่นจากรหัสทิศทางโดยมีอัลกอริทึมดังนี้

1. เริ่มประมวลผลทุกรหัสทิศทางแบบลูกโซ่ เริ่มจาก $I = 1$ ถึง Q

ถ้า $LL(I, 10) = 0$ แล้ว

 ไปประมวลผลยังรหัสทิศทางลูกโซ่ต่อไป

 รวมรหัสทิศทางเดียวกันเข้าด้วยกันเป็นเซกเมนต์ทิศทางจนกว่าจะได้

 เงื่อนโซ่ตามนี้

- 1.1 ไม่มีรหัสทิศทางแบบลูกโซ่ต่อไป
- 1.2 ความยาวของเซกเมนต์ทิศทางมากกว่า 5
- 1.3 มีการเลื่อนไปยังรหัสทิศทางแบบลูกโซ่ต่อไป

สำหรับทุกเซกเมนต์ทิศทาง J ในรหัสทิศทางแบบลูกโซ่ I

 ถ้า ความยาวน้อยกว่า 4 ไม่มีการเลื่อนรหัสลูกโซ่และ รหัสลูกโซ่ I

มีเชกเมนต์ทิศทางมากกว่าหนึ่งเชกเมนต์ แล้ว

ลดเชกเมนต์ J ลงหนึ่ง

บันทึกตัวชี้ย้อนหลัง I และ J ใน $LL(N, 11)$

$$LL(N, 11) = I * 10 + J$$

โดย N เป็นรหัสทิศทางแบบลูกโซ่ต่อจาก I

2. ประมวลผลทุกรหัสทิศทางแบบลูกโซ่ เริ่มจาก $I = 2$ จนถึง Q
รวมทิศทางเดียวกันที่เชกเมนต์ยาวน้อยกว่า 4
รวมเชกเมนต์ยาวหนึ่งเชกเมนต์ เข้ากับเชกเมนต์อื่น
รวมเชกเมนต์ขนาดเล็กของทิศทางที่ต่างกัน
3. ประมวลผลรหัสทิศทางแบบลูกโซ่ที่มีอย่างน้อยหนึ่งเชกเตอร์
ถ้า ขนาดเชกเมนต์ $LL(I, J)$ ยาวมากกว่า 4 แล้ว
บันทึกเชกเมนต์ลงในแอสต์ด
ไม่เช่นนั้น รับค่ารหัสทิศทางแบบลูกโซ่ M ที่มีอย่างน้อยหนึ่งเชกเมนต์
ถ้า ขนาดเชกเมนต์ $LL(M, N)$ ยาวมากกว่า 4 แล้ว
ลบ $LL(I, J)$ และ
บันทึกเชกเมนต์ลงในแอสต์ด
ไม่เช่นนั้น ไปประมวลผลต่อที่ข้อ 2
4. ประมวลผลเชกเมนต์ขนาดเล็ก
ถ้า รหัสลูกโซ่ทั้ง I และ M มีมากกว่าหนึ่งเชกเมนต์ แล้ว
เปลี่ยน M ให้มีเพียงหนึ่งเชกเมนต์
ถ้า รหัสลูกโซ่ K ถัดจาก M มีเพียงหนึ่งเชกเมนต์ แล้ว
ให้ตรวจสอบจนกระทั่ง
 - 4.1 ไม่มีรหัสทิศทางแบบลูกโซ่ถัดไป
 - 4.2 เชกเมนต์ของรหัสทิศทางแบบลูกโซ่ถัดไปยาวมากกว่า 3
 - 4.3 มีเชกเมนต์มากกว่าหนึ่งในรหัสทิศทางแบบลูกโซ่ถัดไป
 ถ้า เชกเมนต์ I M และ K มีความยาวรวมน้อยกว่า 3 แล้ว
กำหนดรหัสทิศทางใหม่

บันทึกเชกเมนต์ลงไนเสต์ค

ไม่เช่นนั้น

ถ้า เป็นเงื่อนไขที่ ข หรือ ค แล้ว

ลบรรหัสทิศทางแบบลูกโซ่ I และ M ที่มีเชกเมนต์เดียว

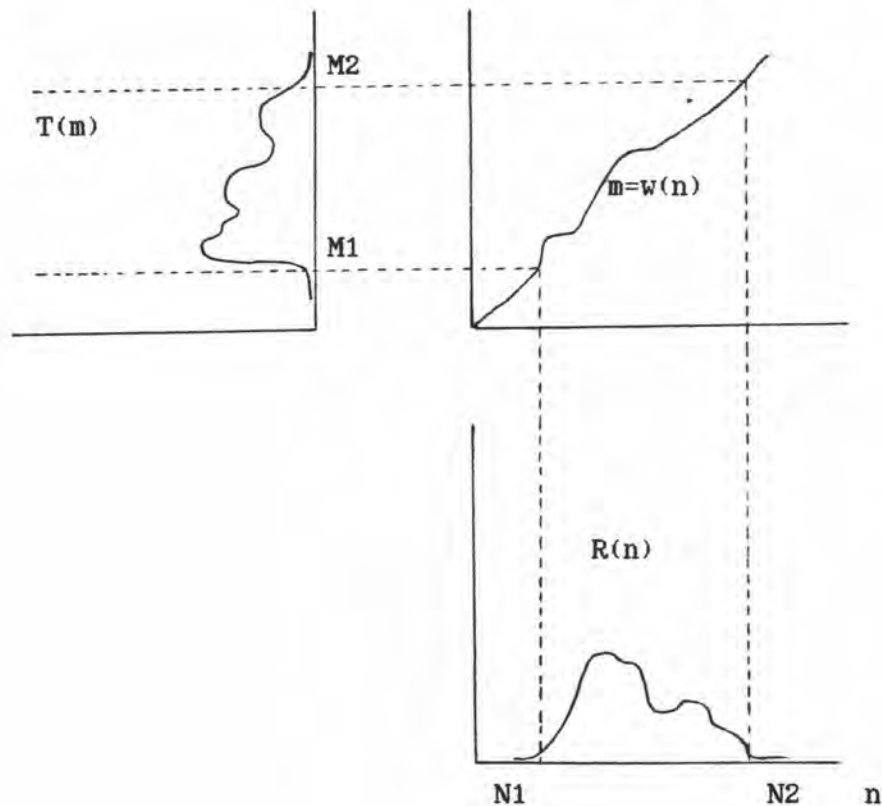
และ บันทึกเชกเมนต์ลงไนเสต์ค

5. ดึงเชกเมนต์จากเสต์ค และ
ดึงค่ารหัสลูกโซ่ I ถัดไป
ถ้า มีค่าไนเสต์คค้างอยู่ แล้ว
ไปประมวลผลข้อ 3

6. หยุด

ไดนามิค โปรแกรมมิ่ง

หลักทฤษฎีพื้นฐานหนึ่งในกระบวนการรับรู้รูปแบบรวมถึงรูปแบบลายมือเขียน และรูปแบบของเสียงพูด คือ ฟังก์ชันแวร์ปีง (Warping Function) หรือการปรับตัวตามแกนเวลาเพื่อลดความแตกต่างของแกนเวลาระหว่างสองรูปแบบจากเวลาที่บันทึกไว้ของรูปแบบอ้างอิงสู่รูปแบบทดสอบบนแกนเวลา อัลกอริทึมซึ่งได้รับความนิยมเพื่อใช้แก้ปัญหานี้คือ ฟังก์ชันแวร์ปีงแบบยืดหยุ่น เป็นการปรับรูปแบบบนแกนเวลาแบบยืดหยุ่น (SAKOE AND CHIBA, 1971; ITAKURA, 1975; RABINER, ROSENBERG AND LEVINSON, 1978) รูปที่ 2.8 แสดงถึงปัญหาทั่วไปของการปรับทางแกนเวลา ให้ $R(n)$, $0 < n < N$ แสดงถึงเส้นรอบรูปของรูปแบบอ้างอิง $T(m)$, $0 < m < M$ เป็นเส้นรอบรูปของรูปแบบทดสอบ โดยมีจุดปลายทั้งสองเป็น N_1, N_2 สำหรับ $R(n)$ และ M_1, M_2 สำหรับ $T(m)$



รูปที่ 2.8 แสดงถึงการปรับตัวตามแกนเวลา

ฟังก์ชันแวก์บิง

จุดประสงค์ของการปรับทางแกนเวลา คือการกำหนดความสัมพันธ์ระหว่างเวลาของทั้งสองรูปแบบที่ชี้ โดย n, m ความสัมพันธ์สามารถแสดงได้โดย W ระหว่าง n, m ด้วย

$$m = W(n) \quad (2.7)$$

ฟังก์ชัน W จะต้องเป็นไปตามเงื่อนไขขอบเขตของจุดปลายของสองรูปแบบ ตัวอย่างเช่น ทั้งจุดเริ่มต้น และจุดสิ้นสุดของรูปแบบที่ได้มีการปรับทางแกนเวลา

$$M_1 = W(N_1) \quad (2.8ก)$$

$$M_2 = W(N_2) \quad (2.8ข)$$

เพื่อให้การกำหนดการทำงานของฟังก์ชัน W สมบูรณ์ จึงสมควรกำหนดสมมุติฐานของรูปลักษณะของ $W(n)$ โดยฟังก์ชัน $W(n)$ จะมีการปรับทางแกนเวลา โดยการหดหรือ ยืด บนแกนเวลาหนึ่ง ซึ่งสัมพันธ์กับอีกแกนเวลาหนึ่งระหว่างจุดปลาย (RABINER, 1978) ในขั้นขั้นซ้อนขึ้น และให้มีประสิทธิภาพเพื่อนำไปสู่ทฤษฎีฟังก์ชันแวก์บิง จึงมีการตั้งข้อจำกัดของ ฟังก์ชันแวก์บิงให้เป็นไปตามเซทของเงื่อนไขต่อไปนี้

$$W(n+1) - W(n) = 0, 1, 2 \quad (W(n) \neq W(n-1)) \quad (2.9ก)$$

$$= 1, 2 \quad (W(n) = W(n-1)) \quad (2.9ข)$$

ทั้งสองสมการจะต้องได้ $W(n)$ ที่เพิ่มขึ้นอย่างต่อเนื่องไปในทิศทางเดียวกัน โดยมีค่าความชันสูงสุด 2 และต่ำสุด 0 เว้นแต่ว่าค่าความชันก่อนหน้าเป็น 0 ค่าต่ำสุดจะเท่ากับ 1

จากเงื่อนไขของขอบเขต และเงื่อนไขความต่อเนื่อง ทำให้ฟังก์ชันแวก์บิง W อยู่ภายในแนวราบรูปสี่เหลี่ยมด้านขนาน (m, n) ตามรูปที่ 2.9 โดยกำหนดค่าให้ $N_1 = M_1 = 1$, และ $N_2 = N$, $M_2 = M$ จุดหักเห A และ B ได้จากการตัดกันของเส้นทั้งสอง

$$m - 1 = 2(n-1) \quad \text{จุด A} \quad (2.10ก)$$

$$m - M = (n-N)/2 \quad (2.10ข)$$

และ

$$m - 1 = (n-1)/2 \quad \text{จุด B} \quad (2.11ก)$$

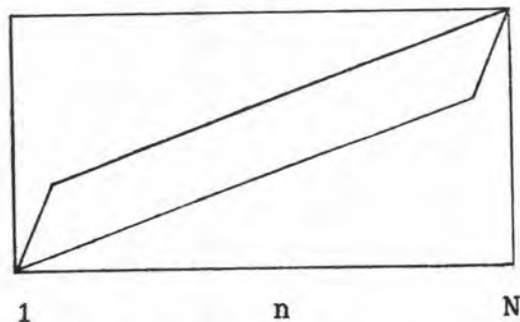
$$m - M = 2(n-N) \quad (2.11ข)$$

ไดนามิคฟังก์ชันแวกซ์บิง w ถูกกำหนดให้เลือกไปในทิศทางภายใต้ระนาบรูปสี่เหลี่ยมด้านขนานตามรูปที่ 2.9

ฟังก์ชันความแตกต่างของรูปแบบ D จะต้องคำนวณหาทุกๆ จุดของ (n, m) ภายใต้ระนาบรูปสี่เหลี่ยมด้านขนานตามรูปที่ 2.9 ค่าของ D น้อยลงมากเท่าใดก็เท่ากับรูปแบบอ้างอิง $R(n)$ กับรูปแบบทดสอบ $T(m)$ มีความลักษณะใกล้เคียงมากเท่านั้น

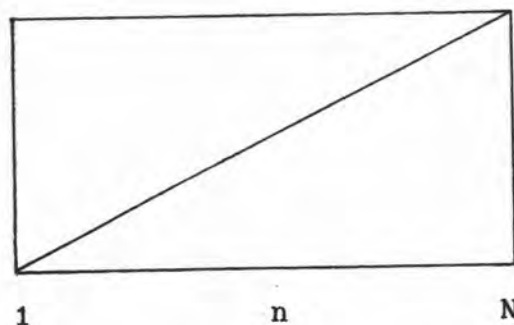
อัตราส่วน 3 ต่อ 2 ของ n และ m

$$m = 2n/3$$



อัตราส่วน 2 ต่อ 1 ของ n และ m

$$m = n/2$$



รูปที่ 2.9 แสดงถึงขอบเขตที่ฟังก์ชันแวกซ์บิงสามารถเคลื่อนไปได้

วิธีการหนึ่งที่มีประสิทธิภาพสูงในการคำนวณหาทิศทางในการเลือกการปรับ เวลาแต่ละช่วงเวลาของไดนามิคฟังก์ชันแเวอร์บิง P ที่ดีที่สุด คือ ไดนามิคโปรแกรมมิ่ง

เวกเตอร์ของรูปแบบ

การเปรียบเทียบแบบ ไดนามิค โปรแกรมมิ่งที่จะกล่าวถึงนี้เป็นอัลกอริทึมการเปรียบเทียบรูปแบบโดยสามารถปรับตัวชี้คหยุดทางแกนเวลาได้ในตัว ความคิดพื้นฐานนี้เป็นที่ยอมรับเมื่อการทดลองเริ่มแรกกับการรับรู้คำพูดตัวเลขภาษาญี่ปุ่น โดยสามารถรับรู้ถูกต้องถึงร้อยละ 99.8 แสดงให้เห็นถึงประสิทธิภาพของไดนามิคโปรแกรมมิ่ง (SAKOE et al., 1978) วิธีการเปรียบเทียบแบบไดนามิค โปรแกรมมิ่งเริ่มด้วยแยกลักษณะรูปแบบ สามารถแทนได้ด้วยลักษณะสำคัญของรูปแบบที่แยกออกเป็นเวกเตอร์ (Vector)

$$A = a_1, a_2, \dots, a_i, \dots, a_l \quad (2.12ก)$$

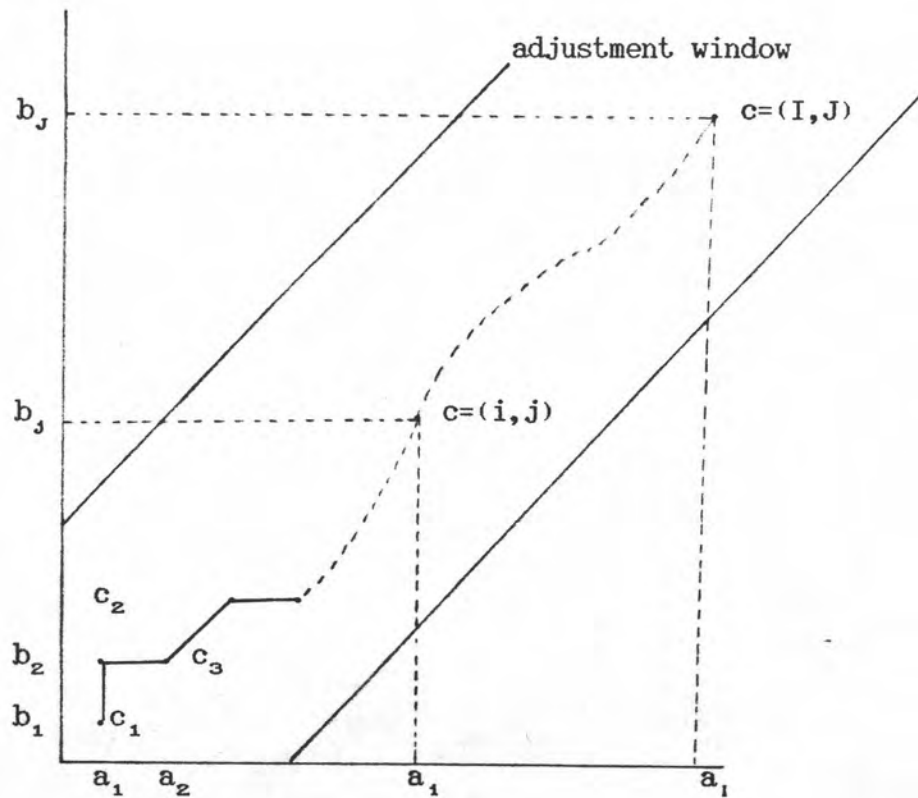
$$B = b_1, b_2, \dots, b_j, \dots, b_r \quad (2.12ข)$$

การแก้ปัญหาความแตกต่างของแกนเวลาระหว่างสองรูปแบบ โดยที่รูปแบบ A อยู่บนแกน i และรูปแบบ B อยู่บนแกน j A และ B นั้นเป็นรูปแบบที่อยู่ในกลุ่มเดียวกัน ความแตกต่างของเวลาระหว่างสองรูปแบบสามารถแสดงได้โดยลำดับจุด $C = (i, j)$ ตามรูปที่ 2.10 แสดงถึงการเคลื่อนตัวของฟังก์ชันแเวอร์บิงภายใต้การปรับขนาดของขอบเขต

$$F = C(1), C(2), \dots, C(k), \dots, C(K) \quad (2.13)$$

โดย $C(k) = (i(k), j(k))$

ลำดับจุดนั้นสามารถนำไปแทนฟังก์ชันได้เพื่อประมาณสัดส่วนการปรับแกนเวลาของรูปแบบ A กับรูปแบบ B เรียกว่าฟังก์ชันแเวอร์บิง ฟังก์ชันแเวอร์บิงจะตรงกับเส้นทะแยงมุม $j = i$ เมื่อไม่มีความแตกต่างของเวลาระหว่างรูปแบบทั้งสอง และจะผันแปรห่างจากเส้นทะแยงมุมมากขึ้นตามความแตกต่างของเวลาที่มากขึ้น



รูปที่ 2.10 แสดงถึงฟังก์ชันแวร์บิง และการปรับขนาดขอบเขต

ความแตกต่างของเวกเตอร์

ความแตกต่าง (Distance) ระหว่างสองเวกเตอร์ a_1 และ b_j สามารถคำนวณได้ตามสมการ 2.14

$$d(c) = d(i,j) = || a_1 - b_j || \quad (2.14)$$

เมื่อมีการให้น้ำหนักแก่เส้นทางย่อยให้ได้ความแตกต่างสะสมของรูปแบบตามฟังก์ชันการปรับตัวตามแกนเวลา

$$E(F) = \sum_{k=1}^K d(c(k)) \cdot w(k) \quad (2.15)$$

โดย $w(k)$ เป็นค่าสัมประสิทธิ์ที่มีค่าเป็นบวกเสมอ เพื่อให้การคำนวณของ $E(F)$ มีลักษณะยืดหยุ่น และเป็นการคำนวณที่เหมาะสมกับฟังก์ชันแวร์บ์บั้ง F โดยจะมีค่าน้อยที่สุดเมื่อกำหนดฟังก์ชันแวร์บ์บั้งให้มีการปรับแกนเวลาที่ตีที่สุด ค่าความแตกต่างที่แท้จริงระหว่างรูปแบบ A และ B จะได้หลังจากที่ตัดทอนความแตกต่างของเวลาระหว่างสองรูปแบบ และจะคงที่ต่อความยืดหยุ่นของแกนเวลา ดังนั้นสมการความแตกต่างระหว่างสองรูปแบบสามารถกำหนดได้โดย

$$D(A, B) = \min_F \left[\frac{\sum_{k=1}^K d(c(k)) \cdot w(k)}{\sum_{k=1}^K w(k)} \right] \quad (2.16)$$

โดยตัวหาร $\sum w(k)$ นั้นเป็นค่าที่ใช้ลดผลกระทบของ K จำนวนจุดบนฟังก์ชันแวร์บ์บั้ง สมการนี้เป็นเพียงพื้นฐานของความแตกต่างของรูปแบบ (SAKOE et al., 1978) ผลกระทบต่อคุณลักษณะที่สำคัญในการตรวจวัดรูปแบบโดยอาศัยทฤษฎีทางไดนามิคโปรแกรมมิ่ง ขึ้นอยู่กับการกำหนดฟังก์ชันแวร์บ์บั้ง และกำหนดสัมประสิทธิ์น้ำหนัก การกำหนดคุณลักษณะของการวัดการปรับความแตกต่างทางแกนเวลานั้นเปลี่ยนแปลงไปตามคุณสมบัติของรูปแบบ จึงกำหนดเงื่อนไข 2 ข้อสำหรับใช้กับรูปแบบโดยทั่วไป

1. การสุ่มตัวอย่างของรูปแบบทำโดยอาศัยพื้นฐานเดียวกัน และช่วยเวลาการสุ่มเดียวกัน

2. ไม่ทราบว่าส่วนใดของรูปแบบมีส่วนสำคัญในการบอกลักษณะข้อกำหนดของ ฟังก์ชันแวน์ปริง

ฟังก์ชันแวน์ปริง F ที่กำหนดในสมการที่ 2.13 เป็นแบบของแกนเวลาที่เปลี่ยนแปลงของรูปแบบ ตามการประมาณคุณสมบัติของการเปลี่ยนแปลงบนแกนเวลาจริง ซึ่งสามารถมองได้ว่าฟังก์ชัน F เป็นการเชื่อมโยงแกนเวลาของรูปแบบหนึ่ง ไปสู่อีกรูปแบบหนึ่ง โดยมีให้เสียลักษณะและโครงร่างของรูปแบบสิ่งสำคัญ คือ โครงสร้างแกนเวลาของรูปแบบ เป็นไปในลักษณะต่อเนื่อง (Continuity) และเป็นไปในลักษณะเดียวกัน (Monotonicity) จากเงื่อนไขสามารถนำมาประยุกต์กับฟังก์ชันแวน์ปริง F หรือจุด $c(k) = (i(k), j(k))$

1) เป็นลักษณะเดียวกัน

$$i(k-1) \leq i(k) \quad \text{และ} \quad j(k-1) \leq j(k)$$

2) การต่อเนื่อง

$$i(k) - i(k-1) \leq 1 \quad \text{และ} \quad j(k) - j(k-1) \leq 1$$

ผลที่ได้จากเงื่อนไขทั้งสอง แสดงให้เห็นถึงความสัมพันธ์ระหว่างจุดต่อเนื่องสองจุด

$$c(k-1) = \begin{cases} i(k), j(k-1), \\ (i(k)-1, j(k)-1), \\ \text{หรือ} (i(k)-1, j(k)) \end{cases} \quad (2.17)$$

3. เงื่อนไขขอบเขต

$$i(1)=1, \quad j(1)=1 \quad \text{และ}$$

$$i(k)=I, \quad j(k)=J \quad (2.18)$$

4. การปรับขนาดขอบเขต

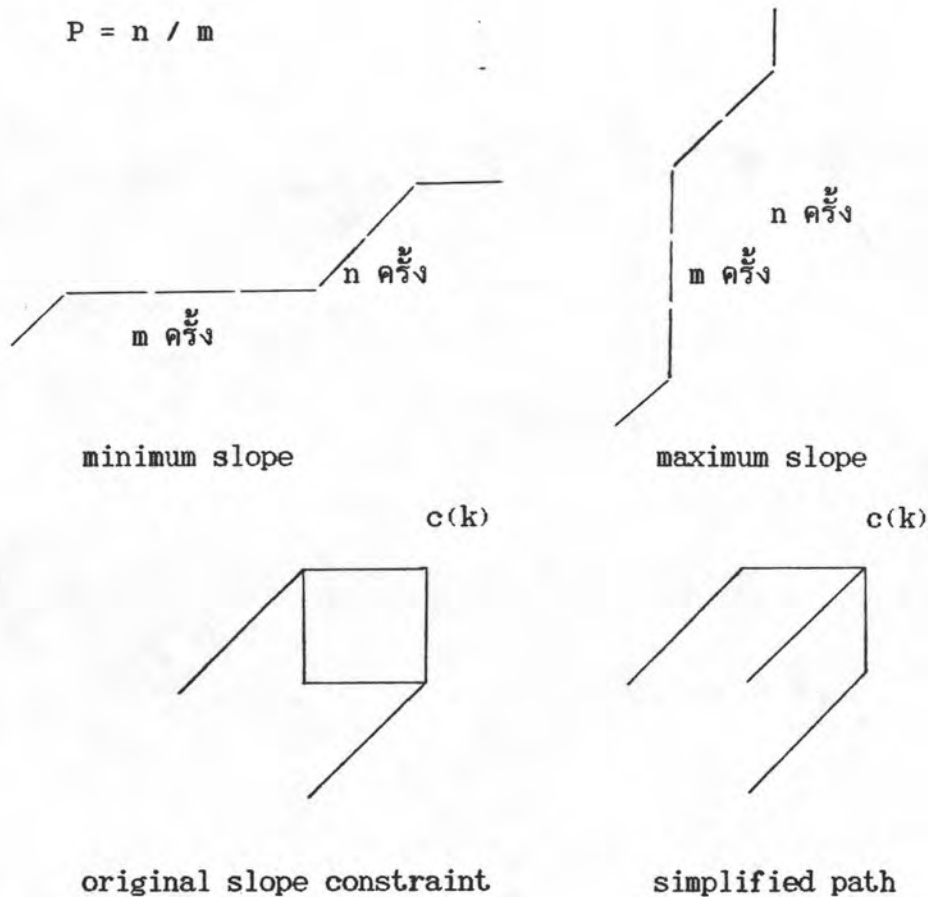
$$| i(k)-j(k) | \leq r \quad (2.19)$$

r จะเป็นค่าเลขจำนวนเต็มที่เป็นขนาดความยาวของขอบเขตในการคำนวณ

5. ขนาดความชัน

เพื่อไม่ให้เป็นการกระทบต่อแกนเวลาที่ต้องการให้มีการปรับ จึงจำเป็นต้องสามารถรับสถานการณ์ของความชัน ไม่ว่าจะเป็นความชันมากหรือน้อย ความสามารถนี้คือการติดตามจุดถึงจุดของฟังก์ชันเวร์บั้งอย่างต่อเนื่อง เพื่อให้ดูชัดเจนขึ้น จากรูปที่ 2.11 (ก) และ (ข) จุด $c(k)$ เคลื่อนที่ไปในทิศทางของแกน i หรือ j ติดต่อกัน m ครั้ง จุด $c(k)$ ก็จะไม่สามารถเคลื่อนต่อไปในทิศทางเดิมอีกจนกว่าจะได้เคลื่อนไปในทิศทางทะแยง n ครั้ง ผลของความชันสามารถประเมินได้โดย

$$P = n / m$$



รูปที่ 2.11 ขนาดความชันของฟังก์ชันเวร์บั้ง

เมื่อ $P = 0$ นั้นก็คือ ไม่มีข้อกำหนดต่อฟังก์ชันแวร์บิง

เมื่อ $P = \infty$ นั้นก็คือ $m = 0$ แสดงว่าฟังก์ชันแวร์บิงนั้นถูกกำหนดให้เป็นเส้น
 ทะแยงมุม $j = i$ ข้อกำหนดของความชันจะมีการปรับให้อยู่ในระดับที่เหมาะสม ถ้ามีค่า
 มากเกินไปอาจทำให้การปรับตัวทางแกนเวลาของรูปแบบไม่มีประสิทธิภาพดีพอ แต่ถ้าน้อย
 เกินไปอาจทำให้การแยกความแตกต่างระหว่างรูปแบบโดยเฉพาะรูปแบบต่างกลุ่มเป็นไปได้
 ยาก

จากรูปที่ 2.11 (ค) และ (ง) การเคลื่อนไปที่จุด $c(k)$ ภายใต้เงื่อนไข
 ขนาดความชัน $p=1$ นั้น รูปที่ (ค) เป็นการเคลื่อนไปตามทิศทางโดยตรงตามเงื่อนไขที่ได้
 กล่าวมาแล้วในขั้นต้น ส่วนในรูปที่ (ง) เป็นการประมาณการโดยเพิ่มข้อกำหนดให้การ
 เคลื่อนไปของแต่ละจุดต่อมาจะ ไม่มีการเปลี่ยนทิศทางวันแต่จะเปลี่ยนทิศไปในทิศทางทะแยง
 ซึ่งจะเป็นการลดการคำนวณตามจำนวนทิศทางเคลื่อนที่ๆ น้อยลง

สัมประสิทธิ์น้ำหนักจากสมการที่ 2.16 ซึ่งใช้ค่าความแตกต่างรวมของรูปแบบ
 ส่วนของสมการเขียนได้เป็น

$$N = \sum_{k=1}^K w(k) \quad (2.20)$$

ตัวหารของสมการที่ 2.16 ซึ่งเป็นอิสระจากฟังก์ชันแวร์บิงสามารถแยกออก
 จากวงเล็บและเขียนสมการใหม่ได้เป็น

$$D(A,B) = \frac{1}{N} \min_F \left[\sum_{k=1}^K d(c(k)) \cdot w(k) \right] \quad (2.21)$$

ผลกระทบของการให้สัมประสิทธิ์น้ำหนักต่อวิธีการแก้ปัญหาของไดนามิคโปรแกรม
 มิ่งสามารถแบ่งออกเป็น 2 ประเภท

1. แบบแกนเวลาสมมาตร (Symmetric) เป็นการแปลงแกนเวลาของทั้งสองรูปแบบไปสู่แกนเวลาร่วมชั่วคราว

$$w(k) = (i(k) - i(k-1)) + (j(k) - j(k-1)) \quad (2.22)$$

ดังนั้น

$$N = I + J \quad (2.23)$$

ซึ่ง I และ J เป็นความยาวของรูปแบบตามลำดับ

2. แบบแกนเวลาไม่สมมาตร (Asymmetric) เป็นการแปลงแกนเวลาของรูปแบบหนึ่งไปยังแกนเวลาของอีกรูปแบบหนึ่ง

$$w(k) = (i(k) - i(k-1)) \quad (2.24)$$

ดังนั้น

$$N = I \quad (2.25)$$

หรือ

$$w(k) = (j(k) - j(k-1))$$

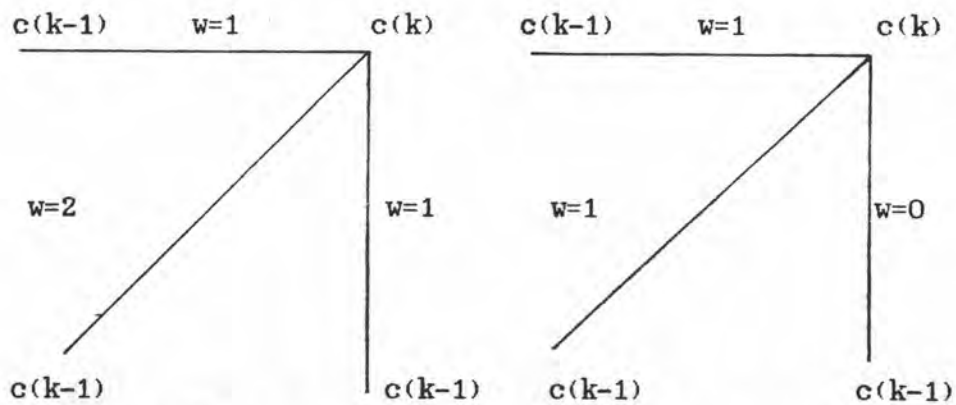
และ

$$N = J$$

แนวทางเบื้องต้นของแบบสมมาตร และแบบไม่สมมาตรนั้น เป็นแนวทางที่เริ่มต้นจาก Sakoe และ Chiba (SAKOE et al., 1978)

แบบแกนเวลาสมมาตรนั้น สรุปได้ว่าสมการที่ 2.16 มีแกนเวลาชั่วคราว $l = i+j$ ส่วนแบบแกนเวลาไม่สมมาตรนั้นแกนเวลาจะขึ้นอยู่กับแกนเวลา i หรือ j แกนหนึ่งแกนใดตามที่กำหนดจากผลของความแตกต่าง ทำให้ค่าความแตกต่างของรูปแบบที่คำนวณได้จากแกนเวลาสมมาตร $D(A,B) = D(B,A)$ ส่วนการคำนวณหาความแตกต่างของรูปแบบจากแกนเวลาไม่สมมาตร $D(A,B) \neq D(B,A)$ เนื่องจากไม่มีแกนเวลาชั่วคราว l ผลลัพธ์ที่สำคัญอันเกิดจากความแตกต่างนี้คือ ตามรูปที่ 2.12 สัมประสิทธิ์น้ำหนัก $w(k)$ ลดลงเหลือศูนย์ในแบบแกนเวลาไม่สมมาตร เมื่อจุดของฟังก์ชันแควร์บิงเคลื่อนไปในทิศทางของแกนเวลา j หรือ $c(k) = c(k-1) + (0,1)$ นี้หมายถึงว่าลักษณะเวกเตอร์

b_j บางเวกเตอร์จะถูกกันออกไป ในทางตรงข้ามการคำนวณแบบแกนเวลาสมมาตร ค่า น้อยที่สุดของ $w(k)$ จะได้เท่ากับ 1 ซึ่งหมายถึงจะไม่เกิดการกันเวกเตอร์ได้ออกจากการ คำนวณ ตามที่ได้กล่าวมาแล้วนั้นจะคล้ายกับว่าการคำนวณด้วย วิธีแกนเวลาแบบสมมาตร จะได้ผลลัพธ์ที่ดีกว่า แต่อย่างไรก็ตามถ้ามีการผสมผสานที่ดีกับข้อจำกัดของค่าความชันแล้ว จุดบนฟังก์ชัน แวร์บ์ปึงจะเคลื่อนไปในทิศทางของแกนเวลา j หรือ i ได้จำกัด ทำให้ ความแตกต่างในคำนวณแบบแกนเวลาสมมาตรและแบบแกนเวลาไม่สมมาตรลดน้อยลง จนอาจไม่มีความแตกต่างเลยก็ได้



รูปที่ 2.12 สัมประสิทธิ์น้ำหนัก $w(k)$ สำหรับอัลกอริทึมทั้งแบบแกนเวลาสมมาตร และแบบแกนเวลาไม่สมมาตร

สมการไดนามิกโปรแกรมมิ่ง

การหาความแตกต่างระหว่างเวกเตอร์ตามสมการที่ 2.21 สามารถเขียนเป็น อัลกอริทึมพื้นฐานเพื่อใช้ในการคำนวณได้ดังนี้

เงื่อนไขเริ่มต้น

$$g_1(c(1)) = d(c(1)) \cdot w(1) \tag{2.26}$$

สมการไดนามิกโปรแกรมมิ่ง

$$g_k(c(k)) = \min_{c(k-1)} [g_{k-1}(c(k-1)) + d(c(k)) \cdot w(k)] \tag{2.27}$$

ระยะทางการปรับตัวตามแกนเวลา

$$D(A,B) = (1/N) g_k(c(K)) \quad (2.28)$$

จากสมการทั้งหมดนี้ สมมติให้ $c(0) = (0,0)$ การคิดสัมประสิทธิ์น้ำหนักแบบ
 แขนเวลาสมมาตร $w(1)=2$ และแบบแขนเวลาไม่สมมาตร $w(1)=1$ เพื่อให้เป็นไปตามข้อ
 จำกัดของฟังก์ชันแวกซ์บิงที่ได้กล่าวมาแล้วสามารถแทนค่าสมการ 2.22 และ 2.23 ลงใน
 สมการ 2.27

ตัวอย่างการคำนวณตามอัลกอริทึมสำหรับแบบแขนเวลาสมมาตร โดยกำหนดไม่
 ให้เงื่อนไขของการเปลี่ยนความชันคือ $P = 0$ จะได้ว่า

เงื่อนไขเริ่มต้น

$$g(1,1) = 2d(1,1) \quad (2.29)$$

สมการไดนามิคโปรแกรมมิ่ง

$$g(i,j) = \min \begin{cases} g(i,j-1)+d(i,j) \\ g(i-1,j-1)+2d(i,j) \\ g(i-1,j)+d(i,j) \end{cases} \quad (2.30)$$

ระยะทางการปรับตัวตามแขนเวลา

$$D(A,B) = (1/N) g(I,J) \quad (2.31)$$

โดย $N = I+J$

การเลือกใช้ไดนามิคโปรแกรมมิ่งแบบแขนเวลาไม่สมมาตร หรือการใช้ข้อจำกัด
 ของค่าความชันจำเป็นที่จะต้องแก้ไขอัลกอริทึม โดยเฉพาะอย่างยิ่งสมการของไดนามิค
 โปรแกรมมิ่งให้มีความเหมาะสม ตามตารางที่ 2.1 สรุปลงอัลกอริทึมสำหรับทั้งแบบแขน
 เวลาสมมาตร และแบบแขนเวลาไม่สมมาตร พร้อมข้อจำกัดค่าความชันแบบต่างๆ สมการ
 ไดนามิคโปรแกรมมิ่งในตารางแสดงการปรับปรุงของแบบแขนเวลาไม่สมมาตร สมการ
 ไดนามิคโปรแกรมมิ่งแรกในแบบแขนเวลาไม่สมมาตรสำหรับค่า $P = 1$ นั่นก็คือ
 $g(i-1,j-2) + (d(i,j-1) + d(i,j)) / 2$ สอดคล้องกับ $C(k-1) = (i(k),$
 $j(k) - 1)$ และ $C(k-2) = (i(k-1) - 1, j(k-1) - 1)$ ตามที่ได้กำหนดไว้ข้างใน

สมการที่ 2.24 ไม่ได้นำมาใช้ $w(k)$ มีค่าเป็นศูนย์ ในขณะที่เดียวกัน $w(k-1)$ มีค่าเป็นหนึ่ง ซึ่งเป็นการลดค่ารวม $d(c(k))$ โดยสิ้นเชิง เพื่อเป็นการหลีกเลี่ยงสถานการณ์ที่กล่าวมานี้ สัมประสิทธิ์น้ำหนัก $w(k-1) = 1$ ถูกแบ่งให้อยู่ระหว่างสัมประสิทธิ์น้ำหนัก $w(k-1)$ และ $w(k)$ เป็นผลทำให้สามารถแทนค่า $(d(i, j-1) + d(i, j)) / 2$ สำหรับ $d(i, j-1) + 0 * d(i, j)$ ในสมการ การแก้ไขในทำนองเดียวกันก็นำมาใช้กับสมการไดนามิกโปรแกรมมิ่งแบบแกนเวลาไม่สมมาตรอื่นๆ ด้วย จากการปรับปรุงนี้ทำให้ผลที่ได้จากการคำนวณแบบแกนเวลาไม่สมมาตรดีขึ้น

ได้มีการใช้อัลกอริทึมทางไดนามิกโปรแกรมมิ่งต่างๆ กันตามกลุ่มนักวิจัยการรับรู้แบบไม่ว่าจะเป็นของเสียงหรืออักษร ซึ่งสามารถเปรียบเทียบอัลกอริทึมได้จากตารางที่ 2.2 ซึ่งได้มีการแก้ไขต่างๆ กันออกไป เพื่อให้ได้ผลเป็นไปตามเงื่อนไขโดยที่ไม่กระทบต่อการปรับตัวตามแกนเวลาของอัลกอริทึม

การรับรู้รูปแบบลายมือเขียนด้วย ไดนามิก โปรแกรมมิ่ง

รูปแบบของตัวอักษรสามารถแทนได้ด้วย $s_1(i)$, $1 < i < I$ และ $s_2(j)$, $1 < j < J$ โดยดัชนีมีความสัมพันธ์ต่อเส้นแต่ละส่วน ซึ่งประกอบขึ้นเป็นอนุกรมการเชื่อมต่อของพิกัดที่สุ่มได้จากดิจิทัลเซอร์ ตามรูปที่ 2.13 ซึ่งแต่ละส่วนเป็นเวกเตอร์สองมิติ แสดงถึงการหักเหทิศทางตามแกนเวลา สามารถคำนวณหาค่าความแตกต่างได้ตามสมการที่ 2.32 (BURR, 1983; CHENG AND FU, 1986)

$$S(i, j) = D(i, j) + \text{Min} \begin{cases} S(i, j-1), \\ S(i-1, j), \\ S(i-1, j-1) \end{cases} \quad (2.32)$$

$D(i, j)$ เป็นผลต่างที่คำนวณได้ ระหว่างหน่วยที่ i ของ s_1 และ หน่วยที่ j ของ s_2 $S(i, j)$ เป็นค่าความแตกต่างสะสม ค่าต่ำสุดของข้อกำหนดการปรับตัวสามารถหาได้ด้วยวิธีที่ได้กล่าวมาแล้ว ส่วนค่า เริ่มต้นจะได้จาก สมการที่ 2.33 ก ข และ ค

ตารางที่ 2.1 อัลกอริทึมของไดนามิกโปรแกรมมิ่งแบบแทนเวลาสมมาตร และแบบแทนเวลาไม่สมมาตร

P	Schematic explanation	Symmetric Asymmetric	DP-equation $g(i,j) =$
0		Symmetric	$\min \begin{cases} g(i,j-1)+d(i,j) \\ g(i-1,j-1)+2d(i,j) \\ g(i-1,j)+d(i,j) \end{cases}$
		Asymmetric	$\min \begin{cases} g(i,j-1) \\ g(i-1,j-1)+d(i,j) \\ g(i-1,j)+d(i,j) \end{cases}$
1/2		Symmetric	$\min \begin{cases} g(i-1,j-3)+2d(i,j-2)+d(i,j-1)+d(i,j) \\ g(i-1,j-2)+2d(i,j-1)+d(i,j) \\ g(i-1,j-1)+2d(i,j) \\ g(i-2,j-1)+2d(i-1,j)+d(i,j) \\ g(i-3,j-1)+2d(i-2,j)+d(i-1,j)+d(i,j) \end{cases}$
		Asymmetric	$\min \begin{cases} g(i-1,j-3)+(d(i,j-2)+d(i,j-1)+d(i,j))/3 \\ g(i-1,j-2)+(d(i,j-1)+d(i,j))/2 \\ g(i-1,j-1)+d(i,j) \\ g(i-2,j-1)+d(i-1,j)+d(i,j) \\ g(i-3,j-1)+d(i-2,j)+d(i-1,j)+d(i,j) \end{cases}$
1		Symmetric	$\min \begin{cases} g(i-1,j-2)+2d(i,j-1)+d(i,j) \\ g(i-1,j-1)+2d(i,j) \\ g(i-2,j-1)+2d(i-1,j)+d(i,j) \end{cases}$
		Asymmetric	$\min \begin{cases} g(i-1,j-2)+(d(i,j-1)+d(i,j))/2 \\ g(i-1,j-1)+d(i,j) \\ g(i-2,j-1)+d(i-1,j)+d(i,j) \end{cases}$
2		Symmetric	$\min \begin{cases} g(i-2,j-3)+2d(i-1,j-2)+2d(i,j-1)+d(i,j) \\ g(i-1,j-1)+2d(i,j) \\ g(i-3,j-2)+2d(i-2,j-1)+2d(i-1,j)+d(i,j) \end{cases}$
		Asymmetric	$\min \begin{cases} g(i-2,j-3)+2(d(i-1,j-2)+d(i,j-1)+d(i,j))/3 \\ g(i-1,j-1)+d(i,j) \\ g(i-3,j-2)+d(i-2,j-1)+d(i-1,j)+d(i,j) \end{cases}$

ตารางที่ 2.2 การเปรียบเทียบอัลกอริทึมของไดนามิกโปรแกรมมิ่งแบบต่างๆ

Algorithm	Initial Condition $g(1,1) =$	Normalization Coefficient N	DP-equation $g(i,j) =$
Sakoe and Chiba	$d(1,1)$	I	$\min \begin{cases} g(i-1, j) + d(i, j) \\ g(i-1, j-1) + d(i, j) \\ g(i-1, j-2) + d(i, j) \end{cases}$
Velichko and Zagoruyko	$a(1,1)$	$\max(I, J)$	$\max \begin{cases} g(i, j-1) \\ g(i-1, j-1) + a(i, j) \\ g(i-1, j) \end{cases}$ where $a(i, j) = 1 - d(i, j)$
White and Neely	$d(1,1)$	$(I + J)$	$\min \begin{cases} g(i-1, j) + d(i, j) \\ g(i-1, j-1) + d(i, j) \\ g(i, j-1) + d(i, j) \end{cases}$
Itakura	$d(1,1)$	I	$\min \begin{cases} g(i-1, j) + a \cdot d(i, j) \\ g(i-1, j-1) + d(i, j) \\ g(i-1, j-2) + d(i, j-1) \end{cases}$ where $a = \infty \text{ (} i(k-1) = i(k-2) \text{)}$ $a = 1 \text{ (} i(k-1) \neq i(k-2) \text{)}$

$$S(1,1) = D(1,1) \quad (2.33ก)$$

$$S(1,j) = D(1,j) + S(1,j-1) \quad (2.33ข)$$

$$S(i,1) = D(i,1) + S(i-1,1) \quad (2.33ค)$$

วิธีการเปรียบเทียบสามารถทำได้โดยการคำนวณหาค่าความแตกต่างสะสมของรูปแบบ หาบรรทัดฐานของค่าที่คำนวณได้เป็นอัตราส่วนของความยาวหน่วย เพื่อลดผลกระทบเนื่องมาจากความแตกต่างของขนาดรูปแบบอ้างอิง การคำนวณหาค่าความแตกต่างของแต่ละหน่วยของรูปแบบนั้นขึ้นอยู่กับตำแหน่งสัมพัทธ์ และการหักเห การให้น้ำหนักตำแหน่งก่อนหน้า และขนาดบรรทัดฐาน จะช่วยเพิ่มความแม่นยำในการหาตำแหน่งพิกัด ส่วนการหาบรรทัดฐานของความเอียงจะช่วยในการหาความหักเห สำหรับเวกเตอร์ทั้งสองที่กำหนดให้ โดยเวกเตอร์จะได้มาจากแต่ละส่วนโค้ง ให้เวกเตอร์ที่ยาวกว่าเป็นเวกเตอร์อ้างอิง (A) การคำนวณได้จากสมการที่ 2.34

$$D(A,B) = (|B \sin \theta| + |C \sin \phi| + y + E)^2 \quad (2.34)$$

เมื่อ C เป็นเวกเตอร์ที่ต่อจากส่วนหางของ A ไปที่จุดกึ่งกลางของ B และสองตัวแปรแรกเป็นการหาความแตกต่างของทิศทาง และตำแหน่ง θ และ ϕ เป็นมุมระหว่าง B และ A และระหว่าง C และ A y เป็นค่าน้ำหนักเพื่อความแม่นยำในการหาความหักเห E เป็นค่าชดเชยการปรับตำแหน่ง การยกกำลัง 2 ในสมการก็เพื่อที่จะให้ได้ค่าความแตกต่างเด่นชัดขึ้นสำหรับค่าไม่เข้าใกล้ค่า 0 และจะช่วยให้ค่าที่เข้าใกล้ค่า 0 เด่นชัดขึ้นด้วย สมการที่ใช้ในการคำนวณจะเป็นสมการสมมาตรโดยที่ $D(A,B) = D(B,A)$



รูปที่ 2.13 เวกเตอร์ของรูปแบบอักษร

ภาคจัดการล่วงหน้าสำหรับข้อมูลรูปแบบออนไลน์

การรับรู้รูปแบบลายมือเขียนออนไลน์สามารถแยกหาลักษณะเด่นเฉพาะของข้อมูลได้จาก จำนวนเส้นที่ลาก หรือลำดับของพิกัดตามแนวเส้นที่ลาก ซึ่งลักษณะเด่นนี้สามารถช่วยในการแยกย่อยกลุ่มอักษรได้ เนื่องจากการป้อนข้อมูลลายมือเขียนนี้สามารถที่จะลากเส้นได้มากกว่าหนึ่งเส้น ในการเขียนอักษรหนึ่งตัว จึงได้กำหนดช่วงเวลาในการรอรับข้อมูลประมาณ 300 ในนั้นของวินาทีหลังจากที่ได้มีการยกปากกาออกจากหน้าสัมผัสของดิจिटัลเซอร์ก่อนที่จะรวมเส้นที่ลากประกอบเป็นหนึ่งอักษร (LOY AND LANDAV, 1982) ซึ่งการรับข้อมูลแบบออนไลน์นั้นสามารถปรับแต่งข้อมูลที่ได้รับให้มีคุณภาพ และประสิทธิภาพในการประมวลผลหาความแตกต่างของรูปแบบ

การปรับเรียบ

ส่วนแรกของภาคจัดการล่วงหน้าคือ การปรับเรียบการหักเหของพิกัดที่สุ่มได้จากดิจिटัลเซอร์ เพื่อกรองการรบกวนของสัญญาณที่รับได้ สามารถคำนวณหาค่าของพิกัดที่ผ่านการกรองตามสมการที่ 2.35 (ARAKAWA, 1983)

$$X_{st} = \Delta S_1 X_{st-1} + \Delta S_2 X_t \quad (2.35ก)$$

$$Y_{st} = \Delta S_1 Y_{st-1} + \Delta S_2 Y_t \quad (2.35ข)$$

โดยที่ X_t , Y_t เป็นตำแหน่งพิกัดที่สุ่มได้จากดิจिटัลเซอร์ ณ ช่วงเวลา t ΔS_1 ΔS_2 เป็นพารามิเตอร์ในการปรับเรียบ โดยที่ $\Delta S_1 + \Delta S_2 = 1$ และ X_{st} , Y_{st} เป็นจุดปรับเรียบที่ผ่านการกรองสัญญาณรบกวน

การสุ่มข้อมูล

การสุ่มตัวอย่างนั้นอาจมีการบิดเบือน และสร้างความซ้ำซ้อนในการประมวลผล ถ้าการสุ่มตัวอย่างไม่ดีพอ โดยเฉพาะการจดปากกาอยู่กับที่ในระหว่างการลากเส้น เพื่อลด

ความซ้ำซ้อนในการประมวลผล จึงจำเป็นต้องลดจำนวนจุดลง การเลือกข้อมูลที่สุ่มได้ สามารถคัดเลือกโดยการสร้างเงื่อนไขตามสมการที่ 2.36

$$(X_{st} - X_{st-1})^2 + (Y_{st} - Y_{st-1})^2 = (\Delta L)^2 \quad (2.36)$$

โดยที่ค่าพารามิเตอร์ $L = 1, 2, \dots$

มีเพียงพิกัดผลลัพธ์ที่มีคุณสมบัติตามที่กำหนด เท่านั้นที่จะผ่านเข้าไปทำการประมวลผลหาความแตกต่างของรูปแบบ

การสร้างรูปแบบบรรทัดฐาน

การสร้างรูปแบบบรรทัดฐานของรูปแบบสามารถแบ่งออกได้เป็น 3 ขั้นตอน (Burr, 1983)

- 1 กำหนดให้จุด $(0,0)$ เป็นจุดศูนย์กลางของรูปแบบ
- 2 กำหนดขนาดบรรทัดฐานโดยที่มีขนาดตามแกนตั้ง และแกนนอนที่แน่นอน
- 3 ลดความเอียงของเส้นให้เป็นการหักเหตามแนวตั้ง

การกำหนดขนาดมาตรฐานโดย

ขนาดตามแกนแนวนอน

$$H = X_{\max} - X_{\min}$$

ขนาดตามแกนแนวตั้ง

$$V = Y_{\max} - Y_{\min}$$

จุดศูนย์กลางของเส้น

$$O = ((X_{\max} + X_{\min}) / 2 , (Y_{\max} + Y_{\min}) / 2)$$

และทำการเคลื่อนจุดศูนย์กลางของเส้นสู่จุด $(0,0)$

การเขียนตัวหนังสือเอียงจะมีผลกระทบในการเปรียบเทียบของสัดส่วน ถึงแม้จะมีการปรับตัวตามแกนเวลาช่วยก็ตาม ดังนั้นจึงจำเป็นต้องทำการปรับความ

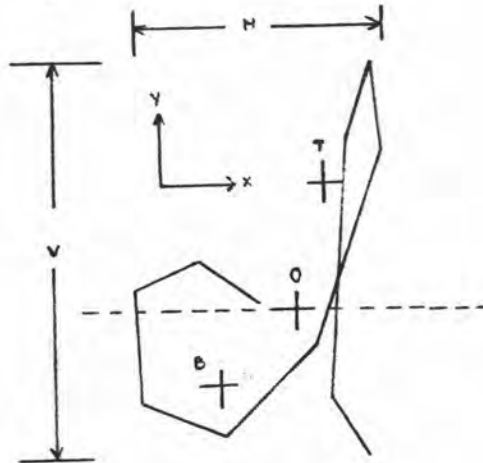
เอียงของอักษรก่อนที่จะสร้างเป็นรูปแบบบรรทัดฐาน วิธีการที่มีประสิทธิภาพวิธีหนึ่งได้ประยุกต์ใช้การคำนวณหาจุดศูนย์กลางของน้ำหนักร่างบนเส้นที่ลากอยู่เหนือ หรือใต้แกน X หรือแกน Y ได้ด้วย T, B ตามลำดับ ตามรูปที่ 2.14 โดยถือว่าทุกสัดส่วนของเส้นจะมีการกระจายน้ำหนักร่างเป็นเอกลักษณ์ มุมของเส้นที่เชื่อมต่อ T, B จะเป็นความเอียงของเส้น จึงจำเป็นที่จะต้องปรับระดับเส้น เพื่อให้ความเอียงนั้นสั่นไหวตั้ง ซึ่งสามารถทำได้โดยแทนค่า X ของแต่ละพิกัดใหม่ตามสมการที่ 2.37

$$X_{\text{new}} = X - Y * \frac{(T_x - B_x)}{(T_y - B_y)} \quad (2.37)$$

เมื่อ $T = (T_x, T_y)$

$B = (B_x, B_y)$

x, y เป็นค่าหลังจากการปรับจุดศูนย์กลาง



รูปที่ 2.14 น้ำหนักร่างของรูปแบบ

การวิเคราะห์ตัวอักษรไทย

อักษรไทยประกอบด้วยพยัญชนะ 44 ตัว สระ 17 ตัว วรรณยุกต์ 4 ตัว และ
อักษรพิเศษ 4 ตัว

ลักษณะของอักษรไทย

พยัญชนะ

ก	ข	ฃ	ค	ฅ	ฉ	ง	จ
ฉ	ช	ฌ	ฎ	ญ	ฎ	ฏ	ฐ
ฑ	ฒ	ณ	ด	ต	ถ	ท	ธ
น	บ	ป	ผ	ฝ	พ	ฟ	ภ
ม	ย	ร	ล	ว	ศ	ษ	ส
ห	ฬ	อ	ฮ				

สระ

ะ	า	ิ	ี	ึ	ุ	ู	ึ	ุ
เ	แ	โ	ใ	ไ	ย	อ	ฤ	ฦ

วรรณยุกต์

.	◌	◌	◌	◌
---	---	---	---	---

อักษรพิเศษ

๐	๑	๒	๓
---	---	---	---

ในระบบรับรู้รูปแบบลายมือเขียนแบบออนไลน์นั้นสามารถตรวจหาลักษณะเด่นของ
รูปแบบได้จาก จำนวนเส้นที่ลากประกอบเป็นตัวอักษร ลำดับของพิกัดที่ได้จากการลากเส้น
หรือตำแหน่งสัมพันธ์ อักษรไทยสามารถเขียนเป็นเส้นเพียงเส้นเดียว หรือบางอักษรประ
กอบด้วยการลากเส้นมากกว่าหนึ่งเส้น ลักษณะเด่นของเส้นโค้งเว้า หรือมุมของส่วนโค้งนั้น
สามารถที่จะนำมาใช้ในการประมวลผลหาความคล้ายคลึงของส่วนโค้งของอักษรไทยได้

(HINANVANICHAKORN et al., 1985 b) วิธีการนี้จะใช้เวลามากในการประมวลผลของอักษรจำนวนมาก ที่มีโครงสร้างซับซ้อน

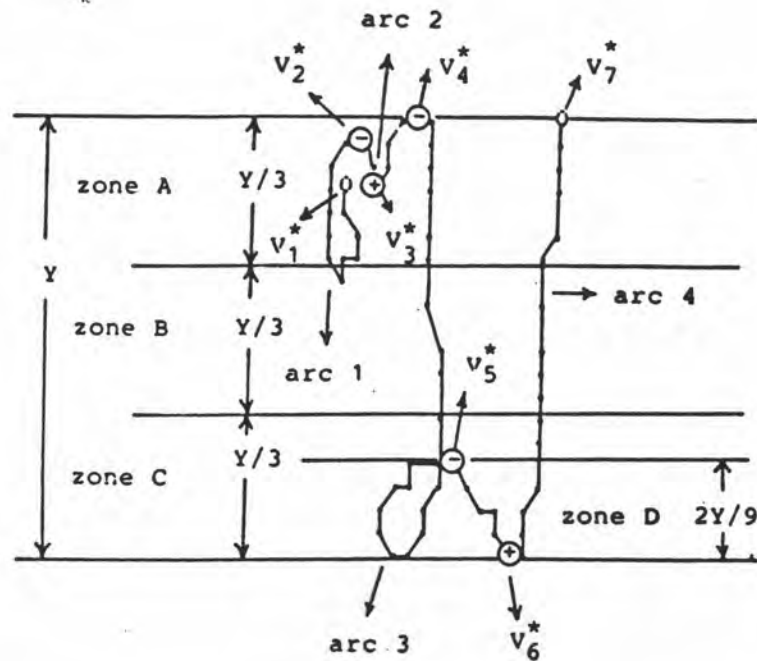
การแยกลักษณะเด่นของอักษรจากตำแหน่งสัมพันธ์ หรือสัดส่วนความโค้ง จะมีประสิทธิภาพมากในการรับรู้รูปแบบ วิธีการสามารถทำได้โดยการแบ่งขอบเขตตามขนาดความสูงของตัวอักษร และขอบเขตดังกล่าวสามารถนำมาใช้ช่วยในการตรวจหาคุณลักษณะของอักษร ส่วนข้อมูลลำดับของเซกเมนต์ของเส้นที่ลาก ตำแหน่งของจุดเริ่มต้น และสิ้นสุดของการลากเส้น สามารถนำมาประยุกต์กับการหาลักษณะเด่นของอักษรไทยได้ โดยที่ข้อมูลเหล่านี้สามารถช่วยในการแยกย่อยกลุ่ม เพื่อลดเวลาการประมวลผลได้ การรับรู้รูปแบบอักษรไทยนั้นสามารถตรวจได้จาก ลำดับของเส้นโค้งที่ลาก ตำแหน่งสัมพันธ์ของเส้นโค้ง และการมีวนของเส้น การตรวจหาความหักเหของทิศทางสามารถคำนวณได้จากสมการที่ 2.37

$$V_j = (V_{xj}, V_{yj}) \quad : \quad j = 1 - I$$

$$dy_{j,j+1} = V_{y,j+1} - V_{yj} \quad (2.37)$$

V_j เป็นจุดสูงสุดของเส้นโค้งในระนาบ X-Y $dy_{j,j+1}$ ได้จากการคำนวณต่างๆ ค่าของจุดยอด ($j = 1 - J-1$) จุดยอดที่มีเครื่องหมายของ $dy_{j,j+1}$ เปลี่ยนจากค่าลบไปเป็นค่าบวกเป็นจุดบน (TP) และการเปลี่ยนค่าจากบวกเป็นค่าลบเป็นจุดล่าง (BP) ทั้งจุดบนและจุดล่างจะเป็นจุดนับของส่วนโค้ง

แสดงถึงระยะห่างระหว่างจุดปลายบนสุด และล่างสุดของเส้นขอบเขตอักษรแบ่งตามความสูงออกเป็นขอบเขต ตามที่อักษรไทยโดยปกติจะ เริ่มเขียนจากขอบเขตส่วนปลายบนสุด หรือส่วนปลายล่างสุด มีอักษรบางส่วนที่เริ่มการเขียนที่ประมาณขอบเขตส่วนกลางในขอบเขตส่วนปลายล่างสุดสามารถแบ่งย่อยออกเป็น 2 ส่วนใน 3 เป็นขอบเขตย่อยอีกส่วนหนึ่งตามรูปที่ 2.15



รูปที่ 2.15 แสดงถึงการแบ่งขอบเขตตามความสูงของอักษรไทย

จากการรวบรวมลักษณะเด่นที่สามารถใช้แทนตัวอักษรตามตารางที่ 2.3 โดยที่

- 1 NS เป็นจำนวนเส้น
- 2 $q_{ns,1}$ เป็นจำนวนเส้นโค้ง และ ค่า ns เริ่มจาก 1 ถึง NS
- 3 $q_{ns,2}$ เป็นค่าเครื่องหมายบวก เมื่อเส้นโค้งแรกเป็นเส้นโค้งไปในทิศทางทวนเข็มนาฬิกา และมีค่าเครื่องหมายลบ เมื่อเส้นโค้งแรกเป็นเส้นโค้งไปในทิศทางตามเข็มนาฬิกา
- 4 $q_{ns,3}$ เป็นค่า 1 เมื่อมีการม้วนของการเริ่มลากเส้น 0 เมื่อไม่มีการม้วน โดยกำหนดให้ k เป็นขนาดขอบเขตในการตรวจสอบนับจากจุดเริ่มของการลากเส้น
- 5 $q_{ns,4}$ เป็นตำแหน่งของจุดเริ่มในการลากเส้น เมื่อมีการม้วนของการเริ่มลากเส้น $q_{ns,4}$ จะบอกตำแหน่งของเส้นม้วนตามขอบเขตของการแบ่งตัวอักษร

- $q_{ns,4} = 1$ ถ้าตำแหน่งอยู่ในขอบเขต A
 $q_{ns,4} = 2$ ถ้าตำแหน่งอยู่ในขอบเขต B
 $q_{ns,4} = 3$ ถ้าตำแหน่งอยู่ในขอบเขต C
 6 $q_{ns,5}$ มีค่าเป็น 1 เมื่อมีการม้วนเกิดขึ้นนอกเหนือจากข้อ 4 และมีค่าเป็น 0 เมื่อไม่มีการม้วน
 7 $q_{ns,6}$ เป็นค่าแทนตำแหน่งของการม้วนของเส้นตามข้อ 6
 $q_{ns,6} = 1$ ถ้าตำแหน่งอยู่ในขอบเขต A
 $q_{ns,6} = 2$ ถ้าตำแหน่งอยู่ในขอบเขต B
 $q_{ns,6} = 3$ ถ้าตำแหน่งอยู่ในขอบเขต C
 8 $q_{ns,7}$ เป็นลำดับที่ของส่วนโค้งที่พบการม้วนของเส้นตามข้อ 6
 9 $q_{ns,8}$ มีค่าเป็น 1 เมื่อส่วนโค้งเริ่มต้นอยู่ในขอบเขต D หรือมีค่าเป็น 0 เมื่อส่วนโค้งเริ่มต้นไม่ได้้อยู่ภายในขอบเขต D
 10 $q_{ns,9}$ มีค่าเป็น 1 เมื่อจุดพับสุดท้ายของส่วนโค้งสุดท้ายเป็นจุดบน TP หรือมีค่าเป็น 0 เมื่อจุดพับสุดท้ายของส่วนโค้งสุดท้ายไม่เป็นจุดบน TP
 11 $q_{ns,10}$ มีค่าเป็น 1 เมื่อจุดพับสุดท้ายของส่วนโค้งแรกเป็นจุดบน TP หรือมีค่าเป็น 0 เมื่อจุดพับสุดท้ายของส่วนโค้งแรกไม่เป็นจุดบน TP
 12 $q_{ns,11}$ มีค่าเป็น 1 เมื่อผลต่างของ X ส่วนโค้งแรก และส่วนโค้งสุดท้าย มีค่าเป็นบวก และมีค่าเป็น 0 เมื่อผลต่างนั้นมีค่าเป็นลบ
 13 $q_{ns,12}$ มีค่าเป็น 1 เมื่อ $q_{ns,9}$ มีค่าเป็น 1 และตำแหน่งของปลายเส้นโค้งในส่วนโค้งสุดท้ายอยู่ในขอบเขต C หรือมีค่าเป็น 0 เมื่อไม่เป็นไปตามเงื่อนไขข้างต้น
 14 $q_{ns,13}$ มีค่าเป็น 1 เมื่อ $q_{ns,10}$ มีค่าเป็น 1 และตำแหน่งของปลายเส้นโค้งในส่วนโค้งแรกอยู่ในขอบเขต C หรือมีค่าเป็น 0 เมื่อไม่เป็นไปตามเงื่อนไขข้างต้น
 15 ความยาวของเส้น
 16 $q_{ns,15}$ เป็นอัตราส่วนความยาวของเส้นจากจุดพับสุดท้ายของส่วนโค้งแรกถึงจุดสุดท้ายของเส้น ต่อระยะทางระหว่างจุดทั้งสอง ซึ่งการคำนวณค่า $q_{ns,15}$ จะคำนวณเมื่อ $q_{ns,1}$ มีค่ามากกว่า 1 $q_{ns,9}$ มีค่าเป็น 2 และ $q_{ns,10}$ มีค่าเป็น 1

ตารางที่ 2.3 ตัวอย่างลักษณะเด่นของอักษรไทย

	NS	$q_{1,1}$	$q_{1,2}$	$q_{1,3}$	$q_{1,4}$	$q_{1,5}$	$q_{1,6}$	$q_{1,7}$	$q_{1,8}$	$q_{1,9}$
ส	2	2	-	1	3	0	0	0	1	1
ล	1	2	-	1	3	0	0	0	1	1
ก	1	3	-	0	3	0	0	0	1	1
ถ	1	3	-	1	3	0	0	0	1	1
ฝ	1	3	+	1	2	0	0	0	1	0
ผ	1	3	+	1	1	0	0	0	1	0
ย	1	3	+	1	1	0	0	0	0	0
ม	1	2	-	1	1	1	3	1	1	0
ฆ	1	4	-	1	1	1	3	3	0	0
ฌ	1	4	-	1	2	1	3	3	1	0
ฬ	1	4	-	1	2	0	0	0	1	0
ฬ	1	4	-	1	2	1	1	4	1	0

การแยกตัวอักษรไทยออกเป็นกลุ่ม (KIMPAN, 1983; HIRANVANICHAKORN et al., 1985 b)

ก ก ฎ
 ฅ ฅ ฎ
 ค ศ ค ต
 ม พ
 บ ป ษ ช ษ ษ
 ร ฎ
 ณ ณ ณ ณ
 ฉ น
 พ ฟ ฟิ ผ ผ
 ท ห
 ล ล จ
 อ ฮ ว
 ำ ๗
 ๑ ๒ ๓ ๔
 ๖ ๗ ๘

ความคล้ายคลึงทำให้ยากในการตัดสินใจถึงความแตกต่าง ดังนั้นจำเป็นต้องอาศัยวิธีการสุ่มตัวอย่าง (Sampling) จุดต่างๆ ของอักขระที่ดี เพื่อมิให้สูญเสียคุณสมบัติสำคัญของตัวอักษร

การแยกย่อยกลุ่มอักษรตามลักษณะเด่น

1 การเขียนจะมีสามระดับทำให้สามารถแยกกลุ่มของอักษรออกได้

ระดับบน

๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ ๑๐ ๑๑ ๑๒ ๑๓ ๑๔ ๑๕

ระดับกลาง

ก ข ค ฆ ง จ ฉ ช ฌ ญ
ฎ ฏ ฐ ท ฒ ณ ด ต ถ ท
ธ น บ ป ผ ฝ พ ฟ ภ ม
ย ร ล ว ศ ษ ส ห ฬ อ
ฮ ฤ ะ า แ โ ใ ไ ๗
๑

ระดับล่าง

๑ ๒

2 ลักษณะการเขียนรูปแบบอักษรระดับกลาง ซึ่งสามารถแบ่งกลุ่มออกเป็นกลุ่มได้จากการเริ่มต้นเขียนหัว หรือเขียนแบบไม่มีหัว

แบบไม่มีหัว

ก ฐ ๑

แบบหัวหมุนตามเข็มนาฬิกา

ข ฌ ง จ ฉ ช ฌ ญ ฎ ฏ
ฒ ณ ด ต ถ ท น บ ป พ
ฝ ม ล ษ ส ห ฬ แ โ
ใ ไ

แบบหัวหมุนทวนเข็มนาฬิกา

ค ฎ ฏ ฝ ฝ ภ ย ร ว ศ
อ ฮ

3 ตำแหน่งจุดเริ่มต้นของอักษรระดับกลางสามารถแยกอักษรออกเป็นกลุ่ม

เริ่มจุดบน

ข พ ง ช ช ท ก น บ ป
ผ ผ พ ฟ ม ย ห ห ฬ

เริ่มจุดกลาง

ค จ ฉ ฐ ฒ ด ต ศ อ ฮ

เริ่มจุดล่าง

ก ญ ฎ ฏ ณ ถ ภ ร ล ว
ส

4 ตำแหน่งจุดสุดท้ายของอักษรระดับกลางสามารถแยกอักษรออกเป็นกลุ่มได้

ระดับบน

ข พ ช ช ญ ฒ ณ ธ น บ
ป ผ ผ พ ฟ ม ย ร ห ฬ
ฮ แ ไ ใ

ระดับกลาง

ง จ ฉ ล ว ส อ

ระดับล่าง

ก ค ท ฎ ฏ ฐ ฒ ต ถ ท
ภ ศ ห ฤ

เมื่อรวมคุณสมบัติต่างๆ ที่ได้กล่าวมาแล้วนั้น เข้าด้วยกันใช้เป็นเครื่องมือในการ
จำแนกกลุ่มตัวอักษร