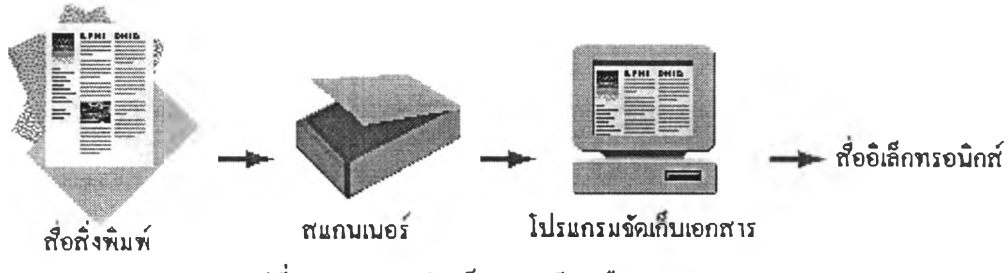
บทที่ 2

ความรู้พื้นฐานที่เกี่ยวข้อง

บทนี้จะกล่าวถึงข้อมูลความรู้พื้นฐานที่เกี่ยวข้องกับการวิจัย โดยจะกล่าวถึง ระบบภาพเอกสาร มาตรฐานการบีบอัดข้อมูลภาพ JBIG เอกสาร HTML การทำเทมเพลตแมตชิง (template matching) และเน็ตสเคปปลักอิน (Netscape plug-in)

2.1 ระบบภาพเอกสาร (Document Imaging System)

ระบบภาพเอกสาร คือ การแปรรูปเอกสารจากสิ่งพิมพ์กระดาษ เช่น หนังสือ บทความ แบบฟอร์ม จดหมาย ฯลฯ ให้อยู่ในรูปสื่ออิเล็กทรอนิกส์ที่คอมพิวเตอร์อ่านได้ เริ่มต้นเอกสารแต่ละหน้าจะถูกสแกนโดยเครื่องสแกน ข้อมูลที่ได้จะอยู่ในรูปบิตแมพ หรือภาพเอกสาร ภาพเอกสารนี้จะใช้เนื้อที่ในการเก็บมาก และต้องการเวลาในการถ่ายโอนข้อมูลมาก จึงจำเป็นต้องทำการบีบอัดข้อมูล (compression) ก่อนเก็บ



รูปที่ 2.1 ระบบจัดเก็บและเรียกคืนภาพเอกสาร

ภาพเอกสารที่ได้มาจะอยู่ในรูปบิตแมพ เราไม่สามารถทราบได้ว่าในภาพเอกสารหนึ่งๆ ประกอบด้วยอะไรบ้าง เนื้อความของเอกสารกล่าวถึงเรื่องใด ดังนั้น ในระบบจัดเก็บเอกสารจึงต้องมีส่วนให้ผู้จัดทำใส่ข้อมูลที่จำเป็น โดยขึ้นกับประเภทของเอกสาร เช่น ชื่อเอกสาร , ชื่อผู้แต่ง , สำนักพิมพ์ มีบทคัดย่อ (abstract) และคำสำคัญ (key word) ข้อมูลเหล่านี้จะถูกจัดเก็บแยกจากข้อมูลภาพเอกสารเพื่อใช้ในการสืบค้นเอกสาร

การเรียกคืนภาพเอกสาร หรือในส่วนของการแสดงภาพเอกสารผ่านทางจอภาพ โดยปรกติจอภาพคอมพิวเตอร์ที่ความละเอียด 1024x768 จุด ไม่สามารถแสดงภาพเอกสารขนาด A4 ที่สแกนด้วยความละเอียด 200 จุดภาพต่อนิ้วได้ทั้งหมด จะแสดงได้เพียงบางส่วนเท่านั้น ดังนั้นใน

ส่วนของการแสดงภาพเอกสารจึงต้องสามารถแสดงภาพเอกสารในความละเอียดต่างๆกันได้ เช่น แสดงทั้งหน้าภายในจอภาพเดียว หรือแสดงหน้าทั้งหมดของเอกสารโดยอยู่ในรูปไอคอน

2.1.1 ข้อดีของการนำภาพเอกสารมาใช้

1. เพิ่มผลผลิต และประสิทธิภาพของพนักงาน โดยสามารถออนไลน์เอกสารให้ผู้ใช้โดยตรง จึงลดเวลาสูญเสียจากการค้นหาเอกสารที่สูญหายหรือวางไว้ผิดที่

2. ทำให้การบริการแก่ลูกค้าดีขึ้น ด้วยการตอบสนองอย่างรวดเร็วในการค้นหา จัดเตรียมแฟกซ์เอกสาร

3. เพิ่มความพึงพอใจให้แก่พนักงาน ด้วยการกำจัดเอกสารที่ไร้ระเบียบ

4. สามารถทำสำเนาเอกสาร เพื่อใช้เป็นข้อมูลสำรอง

5. ลดค่าใช้จ่ายในการใช้เนื้อที่เก็บเอกสาร(storage) และการจัดการเอกสาร

2.1.2 สิ่งที่ต้องคำนึงในการเลือกใช้ระบบภาพเอกสาร

1. ไม่ขึ้นกับแพลทฟอร์ม (ทำงานบน PC, MAC, UNIX, Windows, NT ฯลฯ)

2. ใช้รูปแบบที่เป็นมาตรฐานเพื่อให้สามารถใช้งานได้ระยะยาว

3. ไม่ขึ้นกับซอฟต์แวร์ เพราะจะได้ไม่ต้องยึดติดกับบริษัทผู้ผลิต

4. มีโครงสร้างของดัชนีที่ง่าย เพื่อลดภาระของผู้ใช้ และการยึดติดกับบริษัทผู้ผลิต

5. มีระบบบันทึกข้อมูลอิเล็กทรอนิกส์ที่ใช้งานง่าย และไม่แพง

2.2 เทคนิคการบีบอัดข้อมูลภาพ JBIG

JBIG (Joint Bi-level Image experts Group) เป็นเทคนิคการบีบอัดข้อมูลภาพที่จัดทำโดยสถาบัน ISO ร่วมกับ CCITT และถูกผลักดันให้เป็นมาตรฐานของการเข้ารหัสข้อมูลภาพสองระดับแบบ progressive (ITU-T Recommendation T.82)

JBIG เป็นการบีบอัดข้อมูลภาพ ที่ไม่ยอมให้มีการสูญเสียข้อมูลของภาพ (lossless) โดยมุ่งเน้นไปที่ข้อมูลภาพสองระดับ (bi-level) หรือภาพขาวดำ แต่ก็สามารถใช้บีบอัดข้อมูลภาพหลายระดับได้ (multilevel) โดยการเข้ารหัสระนาบของบิต (bit plane) แต่ละระนาบอิสระกัน นอกจากนั้น JBIG ยังมีคุณสมบัติโปรเกรสซีฟ (progressive) กล่าวคือเมื่อทำการถอดรหัสภาพที่เข้ารหัสแบบ progressive จะสร้างภาพที่มีความละเอียดต่ำ (low-resolution) ก่อน แล้วจึงได้ภาพที่มีความละเอียดเพิ่มมากขึ้นตามลำดับ

2.2.1 ข้อดีของ JBIG

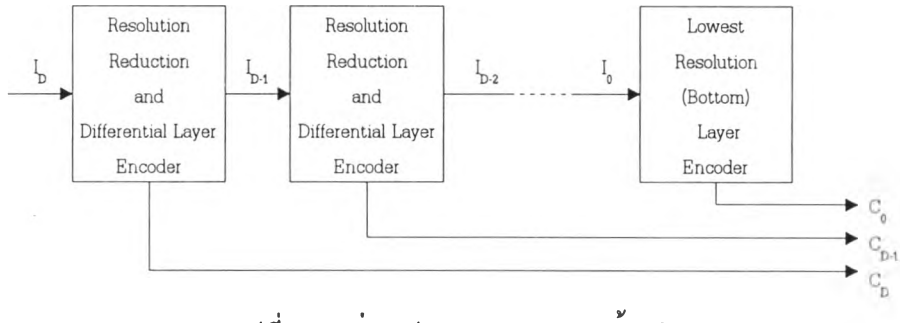
มีอัตราการบีบข้อมูลมากกว่ามาตรฐานอื่นๆ เมื่อใช้กับภาพสองระดับในโหมดที่ไม่มีการสูญเสียข้อมูล[1]

เปรียบเทียบกับอัลกอริธึมในการเข้ารหัสแบบ MMR (ใน Recommendation T.4(G3) และ T.6(G4)) พบว่า เมื่อใช้กับภาพที่ได้จากการสแกนของตัวพิมพ์ (scanned images of printed characters) จะให้อัตราการบีบข้อมูลมากกว่า 1-5 เท่า และเมื่อใช้กับภาพตัวพิมพ์ที่สร้างโดยคอมพิวเตอร์ (computer generated images of printed characters) จะให้อัตราการบีบข้อมูลมากกว่า 5 เท่า และเมื่อใช้กับภาพประเภท halftoning หรือ dithering จะให้อัตราการบีบข้อมูลมากกว่า 2-30 เท่า [2]

เปรียบเทียบกับมาตรฐาน JPEG (ITU-T Recommendation T.81) ในโหมดที่ไม่มีการสูญเสียข้อมูลของภาพ (lossless) เมื่อใช้กับภาพทีมีระดับสีต่ำกว่า 6 บิตต่อจุดภาพ JBIG มีอัตราการบีบข้อมูลมากกว่า และมีอัตราการบีบข้อมูลใกล้เคียงกันในระดับสี 6-8 บิตต่อจุดภาพ [2]

การที่ JBIG มีคุณสมบัติ progressive ทำให้ข้อมูลภาพเดียวกันสามารถนำไปใช้กับจอภาพที่มีความละเอียดต่างๆกันได้ และสำหรับระบบสื่อสารที่มีอัตราการส่งข้อมูลต่ำสามารถใช้เป็น image browsing ได้ กล่าวคือ ภาพที่มีความละเอียดต่ำจะถูกส่งมาก่อนและแสดงทางจอภาพได้อย่างรวดเร็ว และข้อมูลที่ส่งตามมาจะทำให้ภาพชัดขึ้นเรื่อยๆ ผู้ใช้จึงสามารถเห็นรายละเอียดของภาพคร่าวๆ ก่อน และสามารถยกเลิกการส่งข้อมูลภาพได้ตามต้องการ

2.2.2 หลักการเข้ารหัสของ JBIG

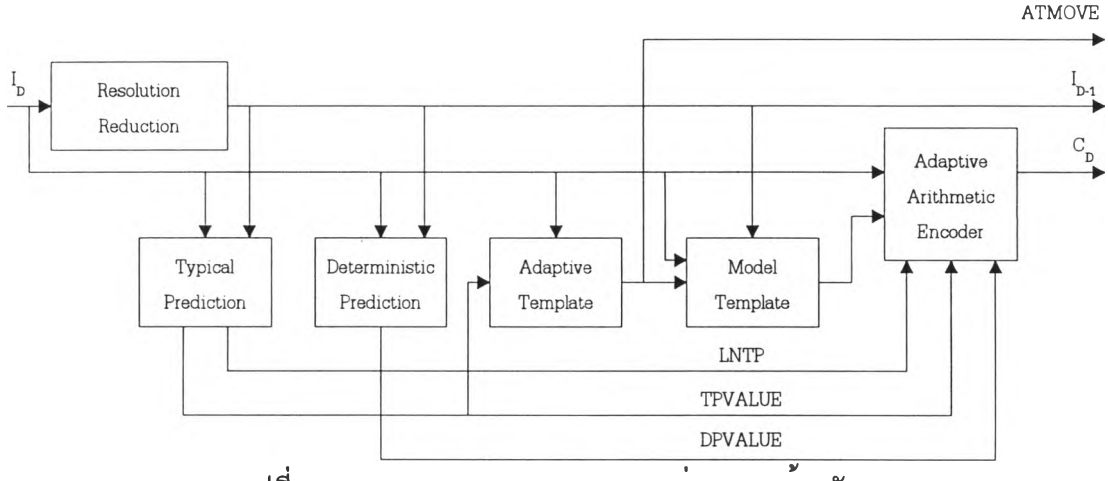


รูปที่ 2.2 ส่วนประกอบของการเข้ารหัส

จากรูปที่ 2 JBIG ประกอบด้วยชั้น (layer) จำนวน D+1 ชั้น แต่ละชั้นจะเก็บข้อมูลภาพที่มีความละเอียดต่างๆ กัน ชั้นใดมีชั้นถัดไปเป็นภาพที่มีความละเอียดต่ำกว่าชั้นนั้นจะถูกเรียกว่าชั้นผลต่าง (differential layer) ส่วนชั้นสุดท้าย(ไม่มีชั้นถัดไป) จะถูกเรียกว่าชั้นล่างสุด (bottom layer)

กรณีที่มีแต่ bottom layer (D = 0) เรียกว่าการเข้ารหัสแบบลำดับ (sequential coding) คือจะไม่มีคุณสมบัติ progressive

เริ่มต้นข้อมูลภาพ I_D ที่รับมาเมื่อผ่านชั้นต่างๆ จะถูกชักตัวอย่างย่อย (subsampling) ให้มีความละเอียดลดลง ภาพที่ชั้นแรกมีความละเอียดสูงสุด ส่วนภาพที่ชั้น bottom layer มีความละเอียดต่ำสุด ต่อจากนั้นจะเข้ารหัสข้อมูลภาพที่ความละเอียดต่างๆ กัน ได้ผลลัพธ์เป็นคำรหัส C_D



รูปที่ 2.3 Resolution Reduction และส่วนการเข้ารหัส

จากรูปที่ 3 แสดงถึงส่วนต่างๆ ของชั้น differential layer ซึ่งประกอบด้วยส่วนลดความละเอียดของภาพ (resolution reduction) , ส่วนการทำนายการเกิดจุดภาพ (pixel prediction) , ส่วนของการคำนวณค่าคอนเท็กซ์ (context) และส่วนการเข้ารหัสแบบเลขคณิตปรับเปลี่ยนได้ (adaptive arithmetic coder)

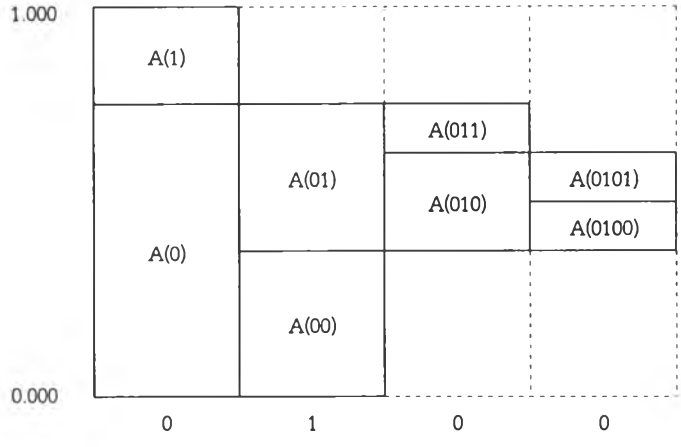
resolution reduction ทำหน้าที่ subsampling โดยจะรับข้อมูลภาพความละเอียดสูงและให้ผลลัพธ์เป็นภาพที่มีความละเอียดต่ำ (เป็นครึ่งหนึ่งของแนวตั้ง และแนวนอน) สำหรับภาพ bi-level ที่มีทั้งตัวหนังสือ(text) และเส้นวาด การทำ subsampling จะทำให้เส้นบางๆหายไป จึงต้องออกแบบวิธีพิเศษเพื่อรักษา edge , line , periodic pattern และ dither pattern จึงได้ผลที่ดีกับภาพแบบต่างๆ (เช่น text, line art, dithered greyscale, halftoned greyscale หรือ error-diffused greyscale)

typical prediction มีไว้เพื่อเพิ่มความเร็วโดยจะข้ามการทำงานของส่วนอื่นสำหรับจุดภาพที่อยู่ในบริเวณแถบสีเดียวกัน (solid color region)

deterministic prediction จะบอกส่วนของการเข้ารหัสไม่ให้เข้ารหัสจุดภาพที่สามารถทำนายได้ว่าจะเกิดค่าใด (จุดภาพที่บอกได้เมื่อรู้จุดภาพที่อยู่ใน low-resolution ที่สอดคล้องกับ high-resolution)

model template จะพิจารณาจุดภาพที่อยู่รอบๆ จุดภาพปัจจุบัน ค่าของจุดภาพเหล่านี้จะแทนด้วยเลขจำนวนเต็มเรียกว่า context เพื่อร่วมในการกำหนดค่าความน่าจะเป็นและใช้ในการเข้ารหัส

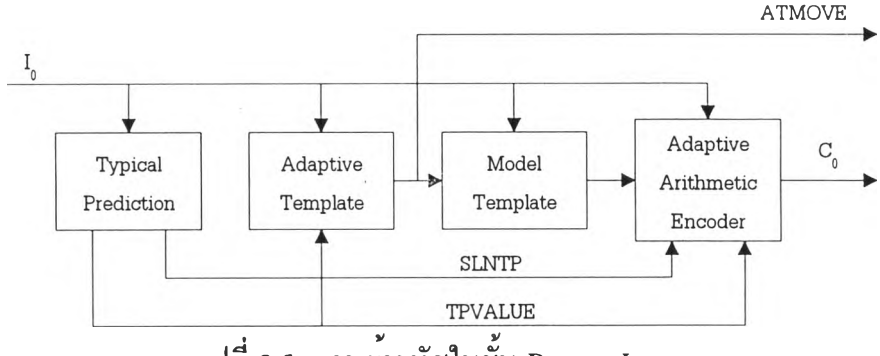
การเข้ารหัสเป็น adaptive arithmetic code (เป็น entropy coder) หลักการพื้นฐานของการเข้ารหัสแบบนี้คือ เริ่มต้นจะมองว่าปัจจุบันมีช่วงอยู่ช่วงหนึ่ง เมื่อต้องการเข้ารหัสเหตุการณ์จะแบ่งช่วงที่มีอยู่ออกเป็นช่วงย่อยๆ ขนาดของแต่ละช่วงย่อยจะเป็นสัดส่วนกับค่าความน่าจะเป็นที่จะเกิดเหตุการณ์นั้น ต่อมาเราจะเลือกช่วงย่อยที่แบ่งไว้ตามที่เหตุการณ์นั้นเกิดขึ้นจริง ผลลัพธ์จากการเข้ารหัสแบบนี้ก็คือตำแหน่งของช่วงย่อยสุดท้ายที่อยู่บนช่วงเริ่มแรก



รูปที่ 2.4 การแบ่งช่วงของการเข้ารหัสแบบ Arithmetic

เนื่องจากผลลัพธ์ของการเข้ารหัสเหตุการณ์แต่ละเหตุการณ์สามารถเป็นเศษของจำนวนบิตได้ จึงมีประสิทธิภาพในการบีบอัดข้อมูลดีกว่า prefix code เช่น Huffman coding ซึ่งเป็น entropy coder เหมือนกัน [4],[6]

ในส่วนของชั้น bottom layer จะต่างจากชั้น differential layer ตรงที่ไม่มีส่วนของ resolution reduction และ deterministic prediction เพราะจะพิจารณาข้อมูลที่มีความละเอียดต่ำสุดเท่านั้น



รูปที่ 2.5 การเข้ารหัสในชั้น Bottom Layer

สำหรับการบีบอัดข้อมูลภาพหลายระดับ (multilevel) ทำได้โดยเข้ารหัสระนาบของบิต (bit plane) แต่ละระนาบอิสระกัน นอกจากนั้น การเปลี่ยนระนาบของบิตจากปรกติ (natural binary) เป็นรหัสเกรย์ (Gray code) สามารถเพิ่มประสิทธิภาพของการบีบอัดข้อมูลภาพได้ประมาณ 15-22%[3] ทั้งนี้เพราะระดับสีที่อยู่ติดกันจะต่างกันเพียงบิตเดียว ทำให้การเปลี่ยนแปลงของบิตเป็นไปอย่างราบเรียบ

2.3 เอกสาร HTML

2.3.1 ความหมายของ HTML

ในการตีพิมพ์เผยแพร่ข้อมูลไปทั่วโลก จำเป็นจะต้องมีภาษาอันหนึ่งซึ่งเป็นที่รู้จักกันทั่วไป SGML (Standard Generalized Markup Language) เป็นมาตรฐาน(ISO 8879) ที่ใช้นิยามภาษาที่เรียกว่า markup language และ HTML เป็นตัวอย่างหนึ่งของ markup language โดย HTML ย่อมาจาก HyperText Markup Language และใช้อย่างแพร่หลายใน World Wide Web

2.3.2 โครงสร้างของเอกสาร HTML

เอกสาร HTML ประกอบด้วย 3 ส่วน คือ

1. บรรทัดที่บอกถึงข้อมูลเกี่ยวกับเวอร์ชันของ HTML
2. ส่วนหัวของเอกสาร (ถูกครอบด้วย HEAD element)
3. ตัวเอกสาร เป็นส่วนเนื้อความของเอกสาร ซึ่งอาจใช้ BODY element หรือ FRAMESET element

ตัวอย่างเอกสาร HTML อย่างง่าย

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
<HTML>
  <HEAD>
    <TITLE> เอกสาร HTML อย่างง่าย</TITLE>
  </HEAD>
  <BODY>
    <P> ตัวเอกสาร
  </BODY>
</HTML>
```

ส่วนหัวและตัวเอกสารจะถูกครอบด้วย HTML element

2.3.3 ข้อมูลเกี่ยวกับเวอร์ชัน

จะบอกให้รู้ว่าใช้ HTML เวอร์ชันใดในเอกสาร และชนิดเอกสารที่นิยามไว้ (document type definition ,DTD)

ตัวอย่างเช่น

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">

เป็น HTML เวอร์ชัน 3.2 และ

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
"http://www.w3.org/TR/REC-html40/strict.dtd">

เป็น HTML เวอร์ชัน 4.0 และ DTD ชนิด strict

2.3.4 ส่วนหัวของเอกสาร

HEAD element ประกอบด้วยข้อมูลเกี่ยวกับเอกสาร เช่น ชื่อเรื่อง คำสำคัญ ซึ่งจำเป็นสำหรับตัวค้นหา (search engine) และข้อมูลอื่นๆ ที่ไม่นับเป็นเนื้อความของเอกสาร

TITLE element เป็นส่วนที่ใช้ระบุชื่อของเอกสาร บราวเซอร์จะใช้ชื่อเอกสารนี้เป็นชื่อหน้าต่าง และเอกสาร HTML ทุกอันจะต้องมี TITLE element ในส่วน HEAD section เสมอ

ข้อมูลเมทา (meta data) HTML อนุญาตให้กำหนดรายละเอียดเกี่ยวกับข้อมูลเมทาได้ดังนี้

<META name="xxx" content="yyy">

xxx ตือคุณสมบัติของ META element และ yyy คือค่าที่กำหนดให้ xxx

เราสามารถใช้แอตทริบิว http-equiv แทน name ได้ และมีความสำคัญเมื่อเอกสารถูกเรียกผ่านโปรโตคอล HTTP (hypertext transfer protocal) โดยเซิฟเวอร์ HTTP อาจใช้ชื่อคุณสมบัติที่ระบุในแอทริบิว http-equiv เพื่อเป็นข้อมูลสำหรับส่วนหัวของ HTTP เช่น

<META http-equiv="Expires"
content="Tue,20 Aug 1996 14:25:27 GMT">

ส่วนหัวของ HTTP จะเป็น

Expires: Tue,20 Aug 1996 14:25:27 GMT

ซึ่งจะมีประโยชน์ในการตรวจไฟล์ในแคชว่า เมื่อใดจึงต้องไปอ่านเอกสารนั้นใหม่

การใช้ META โดยทั่วไปคือ ใช้ระบุชื่อผู้แต่ง และคำสำคัญ เพื่อให้ผลของการค้นหาคำของ search engine มีคุณภาพดีขึ้น ตัวอย่างเช่น

```
<HEAD>
   <TITLE>ระบบจัดเก็บและเรียกคืนภาพเอกสาร</TITLE>
   <META name="author" content="ภาณุมาศ หาดทรายทอง">
   <META name="keywords" content="ภาพเอกสาร, JBIG, keyimage">
</HEAD>
```

2.3.5 ตัวเอกสาร

อาจจะใช้เนื้อที่หน้าต่างทั้งหมดของบราวเซอร์ในการแสดงเนื้อความของเอกสารโดยใช้ BODY element หรือแบ่งหน้าต่างของบราวเซอร์ออกเป็นเฟรมย่อยๆโดยใช้ FRAMESET element

BODY element ใช้ระบุส่วนที่เป็นเนื้อความของเอกสาร โดยมีแอตทริบิวต่อไปนี้

background = uri
   ค่าของแอตทริบิวคือชื่อไฟล์ภาพที่ใช้เป็นพื้นเอกสาร

text = color
   แอตทริบิวนี้จะเซ็ตสีของตัวหนังสือ

link = color
   แอตทริบิวนี้จะเซ็ตสีของตัวหนังสือที่เป็นไฮเปอร์ลิงค์ที่ยังไม่ได้ไป

vlink = color
   แอตทริบิวนี้จะเซ็ตสีของตัวหนังสือที่เป็นไฮเปอร์ลิงค์ที่ยังได้ไปมาแล้ว

alink = color
   แอตทริบิวนี้จะเซ็ตสีของตัวหนังสือที่เป็นไฮเปอร์ลิงค์ที่กำลังอ่าน

bgcolor = color
   ค่าของแอตทริบิวคือสีของพื้นเอกสาร

FRAMESET element ใช้ระบุส่วนที่เป็นเฟรมของเอกสาร โดยมีแอตทริบิวต่อไปนี้

name = cdata
   กำหนดชื่อให้แก่เฟรม ชื่อนี้จะเป็น target ของลิงค์ในภายหลัง

src = uri
   แอตทริบิวนี้จะระบุที่อยู่และไฟล์เริ่มต้นของเฟรม

noresize
   เป็นบูลีนที่กำหนดว่าไม่ให้มีการเปลี่ยนขนาดหน้าต่างของเฟรม

scrolling = auto/yes/no
   auto: แสดง scroll bar เมื่อจำเป็น (เป็นค่าโดยปริยาย)
   yes: แสดง scroll bar ตลอด
   no: ไม่ให้แสดง scroll bar

frameborder = 1/0

marginwidth = pixels

กำหนดปริมาณขอบซ้ายขวาของเฟรม มีหน่วยเป็นจุดภาพและมากกว่าหรือเท่ากับ 1

marginheight = pixels

กำหนดปริมาณขอบบนล่างของเฟรม มีหน่วยเป็นจุดภาพและมากกว่าหรือเท่ากับ 1

2.4 คอร์รีเลชัน และเทมเพลตแมตชิง

คอร์รีเลชัน(correlation) เป็นกลยุทธ์ในการทำแมตชิง (matching) ที่ง่ายและนิยมใช้กัน มันใช้ได้กับสัญญานเวกเตอร์ของสตริง (signal vector string) และเซ็ต (set)

เซ็ตของแพทเทอร์น(pattern) อ้างอิงเรียกว่าเทมเพลต (template) จะถูกใช้ร่วมกับแพทเทอร์นที่ไม่รู้ค่า

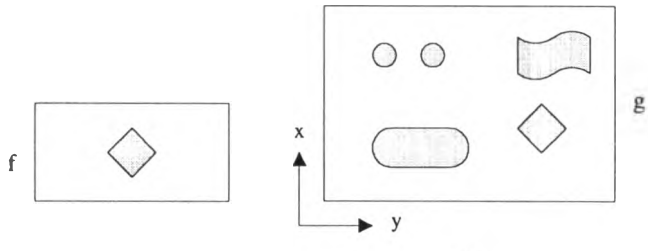
ค่าคอร์รีเลชันของเทมเพลตกับแพทเทอร์นที่ไม่รู้ค่า จะได้จากการเลื่อนเทมเพลตไปยังตำแหน่งต่างๆ และใช้เมตริกซ์แมตชิง (matching matrix) ที่เหมาะสมในการคำนวณฟังก์ชันคอร์รีเลชัน

จากตัวอย่างรูป 2.6 นิยาม

g: เป็นแพทเทอร์นเข้ามา

f: เป็นแพทเทอร์นอ้างอิง หรือเทมเพลต

R: เป็นเนื้อที่ทั่วทั้ง g



รูปที่ 2.6 (a) เทมเพลต $f(x_1,x_2)$ (b) ภาพ $g(x_1,x_2)$

ความต่างหาได้จาก

$$m_1 = \sum_R |f - g| \qquad 2.4.1$$

$$m_2 = \sum_R (f - g)^2 \qquad 2.4.2$$

จากสมการ 2.4.2 กระจายเทอมยกกำลังสอง จะได้

$$m_2 = \sum f^2 - 2\sum fg + \sum g^2 \qquad 2.4.3$$

จะสังเกตุเห็นว่า

1. เทอม $\sum f^2$ และ $\sum g^2$ คือความยาวของเวกเตอร์ยกกำลังสอง สำหรับแพทเทอร์นที่เข้ามา $\sum g^2$ ซึ่งมีค่าคงที่ตลอดการทำแมตชิง จึงไม่มีความหมายมากนัก

2. เทอม $\sum fg$ มีความสำคัญ เพราะแสดงความสัมพันธ์ระหว่างสองเวกเตอร์

นิยาม

$$c_{un} = \sum_R fg \qquad 2.4.4$$

เนื่องจากสัมประสิทธิ์ของเทอมนี้ในสมการที่ 2.4.3 เป็นลบ ดังนั้นเมื่อเทอมนี้ใหญ่ มาตรวัด m_2 จะเล็ก m_2 จึงเป็นมาตรวัดความต่าง และ $\sum fg$ เป็นมาตรวัดความเหมือน และเรียกกระบวนการนี้ว่าคอร์รีเลชันที่ยังไม่นอร์มอไลซ์ (unnormalized correlation) ของ f และ g (บน R)

และคอร์รีเลชันที่นอร์มอไลซ์แล้ว (normalized correlation)

$$c_n = \left(\frac{1}{E}\right)\sum_R fg \qquad 2.4.5$$

โดยที่

$$E = \sum_R f^{1/2} \sum_R g^{1/2} = \|f\| \cdot \|g\| \qquad 2.4.6$$

2.5 ปลักอินของเน็ตสเคปนาวิเกเตอร์ (Netscape Navigator plug-in)

Netscape Navigator plug-in คือ มอดูล(module) อันหนึ่ง ซึ่งเมื่อใช้งานมันจะถูกโหลดและกลายเป็นส่วนหนึ่งของบราวเซอร์ เทคนิคนี้จะทำให้การทำงานเป็นไปได้อย่างรวดเร็วเพราะมอดูลนี้จะอยู่ที่เครื่องคอมพิวเตอร์ที่เป็นบราวเซอร์ จึงไม่จำเป็นต้องเอาจากเซอร์เวอร์ทุกครั้งที่เรียกใช้

จุดประสงค์ของ plug-in คือทำให้โปรแกรม Netscape Navigator ซึ่งเป็นเว็บบราวเซอร์สามารถแสดงข้อมูลและทำงานในรูปแบบพิเศษที่นอกเหนือจากรูปแบบพื้นฐานที่โปรแกรมสามารถแสดงได้

2.5.1 โมดของ plug-in

Netscape Navigator plug-in มีอยู่ด้วยกัน 3 โมดคือ

1. embedded plug-in เป็นส่วนหนึ่งของเอกสาร HTML โดยที่ plug-in นี้จะเห็นเป็นกรอบสี่เหลี่ยมในหน้าเอกสาร HTML ซึ่งคล้ายกับการแสดงรูปภาพชนิด GIF หรือ JPEG

2. full-Page plug-in เป็นตัวแสดงข้อมูลที่ไม่ได้เป็นส่วนหนึ่งของเอกสาร HTML ในโมดนี้ plug-in จะอยู่ในเฟรมของโปรแกรม Netscape ตัวอย่างเช่น Adobe Acrobat viewer

3. hidden plug-in จะทำงานอยู่เบื้องหลัง ตัวอย่างเช่น MIDI player

2.5.2 The Plug-in Application Programming Interface (API)

เนื่องจาก plug-in เป็นโคดมอดูลชนิดไดนามิกลิงค์ (dynamically linked code module) สำหรับระบบปฏิบัติการวินโดวส์ Windows แล้ว มอดูลนี้ก็คือ dynamic link library (DLL) นั่นเอง เทคนิคนี้จะทำให้ประหยัดการใช้ทรัพยากร(resource) ของวินโดวส์ ซึ่งมันจะเรียกโคดเมื่อต้องการใช้

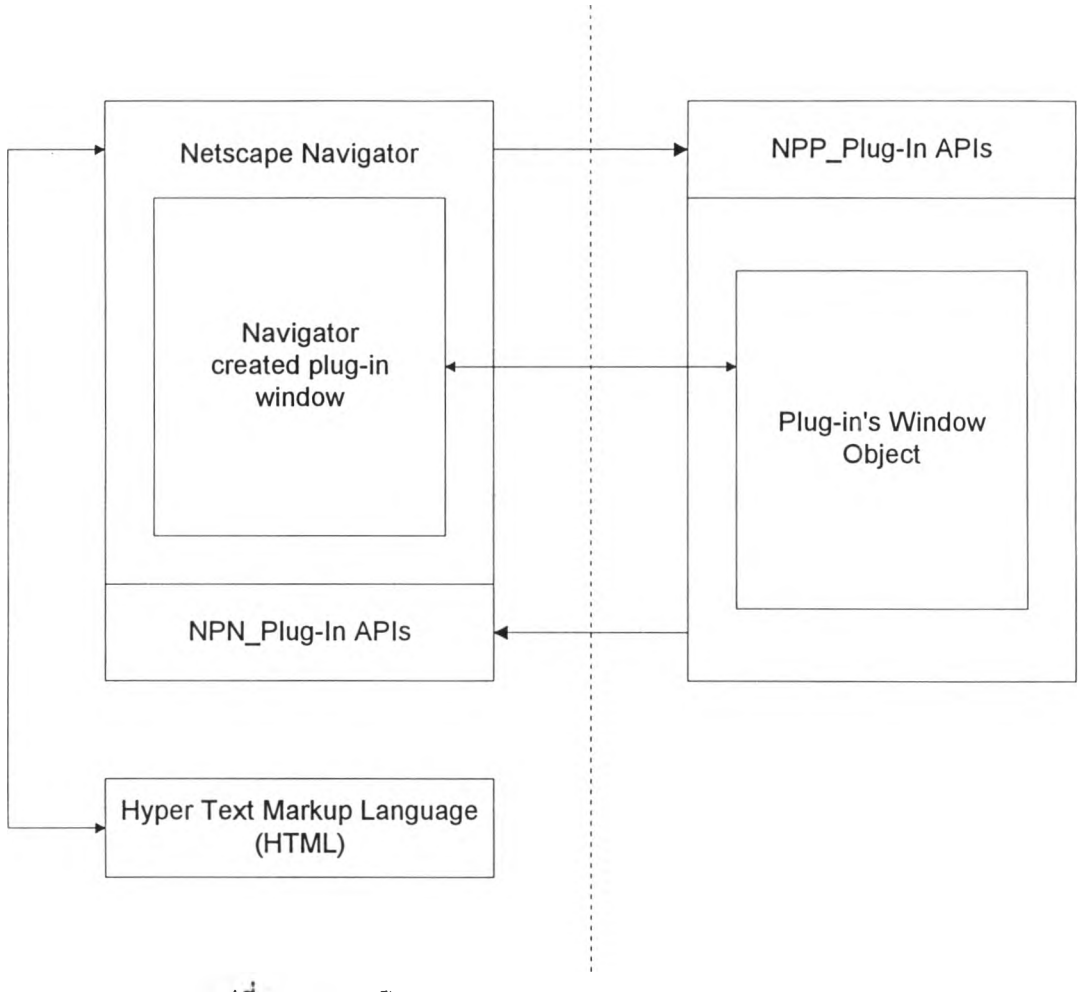
Application Programming Interface (API) ของ plug-in มีด้วยกันสองแบบคือ

1. มีชื่อขึ้นต้นด้วย NPP_ รูทีน(routine) นี้เป็นส่วนของ plug-in (ผู้ทำ plug-in เขียนขึ้น) และมันจะถูกเรียกจาก Navigator

หมายเหตุ NPP ย่อมาจาก Netscape Plug-in : Plug-in defined

2. มีชื่อขึ้นต้นด้วย NPN_ รูทีนนี้เป็นส่วนของ Navigator และให้ plug-in เรียกใช้

หมายเหตุ NPN ย่อมาจาก Netscape Plug-in : Navigator defined



รูปที่ 2.7 สถาปัตยกรรมของ Netscape's Navigator plug-in

รูทีนทั้งหลายที่ถูกเรียกภายใน DLL ต้องผ่าน entry point ดังต่อไปนี้ NP_Initialize NP_GetEntryPoints และ NP_Shutdown

NP_Initialize

ทันทีที่โหลด plug-in รูทีน NP_Initialize จะถูกเรียก

รูทีนนี้มีพารามิเตอร์เพียงตัวเดียว คือ ตำแหน่งของ NPNetscapeFuncs structure

รูทีนนี้จะจบด้วยการเรียก NPP_Initialize (เป็น plug-in API)

NP_GetEntryPoints

หลักจากเรียกรูทีน NP_Initialize รูทีน NP_GetEntryPoints จะถูกเรียก

รูทีนนี้มีพารามิเตอร์เพียงตัวเดียว คือ ตำแหน่งของ NPPluginFuncs structure

ตำแหน่งของฟังก์ชั่นพอยเตอร์ใน NPPluginFuncs structure จะถูกเติมโดย plug-in

หลังจากจบรูทีน NP_GetEntryPoints ฟังก์ชั่นพอยเตอร์จะถูกเรียกโดยตรงจาก Navigator และจะไม่มีการเรียก DLL entry point อื่นๆอีก จนกระทั่งไม่ต้องการโหลด plug-in จึงเรียกรูทีน NP_Shutdown

NP_Shutdown

รูทีนนี้จะเรียก NPP_Shutdown(เป็น plug-in API)

ตารางที่ 2.1 API ของ plug-in ซึ่งเรียกจาก Netscape

ชื่อของ API	รายละเอียด
NPP_Destroy	ลบ instance ของ plug-in
NPP_DestroyStream	ถูกเรียกเมื่อรับข้อมูลจาก stream จนครบ
NPP_GetJavaClass	จะส่งค่าของ plug-in ที่สัมพันธ์กับ Java class
NPP_HandleEvent	handler สำหรับระบบปฏิบัติการ Macintosh
NPP_Initialize	เซ็ตค่าเริ่มต้นให้แก่ตัวแปรชนิด Global
NPP_New	สร้าง instance ใหม่แก่ plug-in
NPP_NewStream	ถูกเรียกเมื่อ stream ใหม่สร้างขึ้นมา
NPP_Print	จัดการเกี่ยวกับการพิมพ์
NPP_SetWindow	ถูกเรียกขณะหน้าต่างของ plug-in ทำงาน
NPP_Shutdown	สิ้นสุดการทำงานของตัวแปรชนิด Global
NPP_StreamAsFile	ให้ชื่อไฟล์ของ stream ที่รับมา
NPP_URLNotify	บอกให้รู้ว่า คำขอสำหรับเรียก URL สมบูรณ์หรือไม่
NPP_Write	เขียนข้อมูลให้แก่ plug-in
NPP_WriteReady	พิจารณาว่า plug-in พร้อมที่จะรับข้อมูลแล้วหรือยัง

ตารางที่ 2.2 API ของ Netscape ซึ่งเรียกจาก plug-in

ชื่อของ API	รายละเอียด
NPN_DestroyStream	สิ้นสุด data stream
NPN_GetJavaEnv	ส่งค่าสภาวะแวดล้อมของ Java
NPN_GetJavaPeer	ส่งค่าของ plug-in ที่สัมพันธ์กับ Java object
NPN_GetURL	สร้าง stream ใหม่ และอ่านข้อมูลจาก URL
NPN_GetURLNotify	สร้าง stream ใหม่ และอ่านข้อมูลจาก URL พร้อมรายงานผล
NPN_MemAlloc	จองหน่วยความจำ
NPN_MemFree	คืนหน่วยความจำ
NPN_NewStream	สร้าง stream ใหม่สำหรับข้อมูล
NPN_PostURL	ส่งข้อมูลไปที่ URL
NPN_PostURLNotify	ส่งข้อมูลไปที่ URL พร้อมรายงานผล
NPN_RequestRead	ต้องการอ่านข้อมูลจาก stream
NPN_Status	แสดงสถานะผาน status bar ของ Netscape
NPN_UserAgent	อาน user agent ของ Netscape
NPN_Version	อ่านรุ่นของ plug-in
NPN_Write	เขียนข้อมูลลงใน stream