บทที่ 2

ทฤษฎีและวารสารปริทรรศน์

2.1 ทฤษฎี

2.1.1 การบีบอัดข้อมูล

ในยุคปัจจุบันที่คอมพิวเตอร์กลายเป็นสิ่งที่มีบทบาทต่อชีวิตประจำวันของมนุษย์เป็นอย่างมาก การทำงานและการสื่อสารต่าง ๆ จึงเกิดขึ้นบนเครื่องคอมพิวเตอร์เป็นหลัก ข้อมูลที่ใช้งานรับและส่งกันไปมาระหว่างเครื่องคอมพิวเตอร์จะเป็นข้อมูลที่อยู่ในรูปแบบของดิจิทัล ไม่ว่าจะเป็นข้อความ ภาพ และเสียง ซึ่งล้วนแต่อยู่ในรูปของดิจิทัลทั้งสิ้น

ข้อมูลภาพจะใช้พื้นที่ในการบรรจุจำนวนมหาศาลเมื่อเทียบกับขนาดของข้อมูลที่เป็นข้อความ นอกจากขนาดที่ใหญ่มาก ๆ แล้ว ยังจะต้องใช้ช่องทางการสื่อสารที่สามารถส่งผ่านข้อมูลได้ในปริมาณมากๆอีกด้วยทำให้เป็นการเสียพื้นที่ในการเก็บรักษา และใช้เวลาในการส่งผ่านเป็นเวลานาน การบีบข้อมูลจึงเริ่มได้รับการพัฒนาเพื่อทำให้ขนาดข้อมูลดิจิทัลเหล่านั้นมีขนาดเล็กลง ซึ่งในขณะเดียวกันก็จะเป็นการลดเวลาการส่งผ่านข้อมูลลงไปด้วย

ในปัจจุบันประชาชนแทบทุกคนจะต้องบริโภคข้อมูลต่าง ๆ เป็นจำนวนมาก ไม่เพียงแต่ข้อมูลที่อยู่ในเครื่องคอมพิวเตอร์เท่านั้นที่จะใช้งานบ่อย ๆ สิ่งรอบตัวหลายอย่างก็เป็นข้อมูลระบบดิจิทัลเช่นกัน และเราทุกคนต่างก็ใช้งานบนข้อมูลอันหลากหลายเหล่านั้นอย่างคุ้นเคย ซึ่งเมื่อพิจารณาลงไปถึงรายละเอียดของข้อมูลแต่ละตัวแล้วพบว่าส่วนใหญ่จะเป็นข้อมูลดิจิทัล บางครั้งก็อยู่ในรูปของการบีบอัด ไม่ว่าจะเป็นเครื่องเล่นซีดีเพลง เครื่องเล่นเอ็มพีสาม เครื่องเล่นวีซีดี เครื่องเล่นวิดีโอเทป โมเด็ม แฟกซ์ โทรศัพท์เคลื่อนที่ โทรทัศน์ ฯลฯ

ข้อมูลดิจิทัลที่อยู่ในรูปแบบต่างๆที่ใช้งานในชีวิตประจำวันเราสามารถคำนวณขนาดข้อมูลได้คร่าว ๆ ซึ่งทำให้ทราบว่าปริมาณข้อมูลสักเท่าไรจึงจะเพียงพอต่อการใช้งาน และในตารางที่ 2-1 บอกถึงระยะเวลาในการส่งผ่านข้อมูลเหล่านั้น

ตารางที่ 2-1 ขนาดข้อมูลลักษณะต่าง ๆ ที่มีการใช้งานอยู่เป็นประจำ โดยจะบอกความละเอียดสำหรับการใช้งานแต่ละอย่างรวมไปถึงบอกระยะเวลาสำหรับการส่งผ่านข้อมูลเหล่านั้นผ่านทางโมเด็ม

Multimedia Data	Size/Duration	Bits/Pixel or Bits/Sample	Uncompressed Size (B for bytes)	Transmission Bandwidth (b for bits)	Transmission Time (using a 28.8K Modem)
A page of text	11" x 8.5"	Varying resolution	4-8 KB	32-64 Kb/page	1.1 - 2.2 sec
Telephone quality speech	10 sec	8 bps	80 KB	64 Kb/sec	22.2 sec
Grayscale Image	512 x 512	8 bpp	262 KB	2.1 Mb/image	1 min 13 sec
Color Image	512 x 512	24 bpp	786 KB	6.29 Mb/image	3 min 39 sec
Medical Image	2048 x 1680	12 bpp	5.16 MB	41.3 Mb/image	23 min 54 sec
SHD Image	2048 x 2048	24 bpp	12.58 MB	100 Mb/image	58 min 15 sec
Full-motion Video	640 x 480, 1 min (30 frames/sec)	24 bpp	1.66 GB	221 Mb/sec	5 days 8 hrs

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

2.1.1.1 พื้นฐานการบีบอัดข้อมูล

การบีบอัดข้อมูลดิจิทัลสามารถแบ่งเป็นประเภทใหญ่ ๆ ได้สองชนิดคือ การบีบอัดข้อมูลแบบไม่มีการสูญเสีย (Lossless Compression) และ การบีบอัดข้อมูลแบบที่มีการสูญเสีย (Lossy Compression) [1]

2.1.1.1.1 การบีบอัดข้อมูลแบบไม่มีการสูญเสีย

เป็นการลดรูปหรือลดขนาดข้อมูลให้เล็กลง ซึ่งสามารถคืนสภาพของข้อมูลต้นฉบับกลับมาได้อย่างสมบูรณ์ เป็นวิธีการที่ไม่มีการทิ้งข้อมูลใด ๆ เลย การบีบข้อมูลลักษณะนี้จะใช้กับงานที่ต้องการความถูกต้องเที่ยงตรงมาก ๆ เช่นการบีบข้อมูลแบบซิพ ซึ่งเป็นลักษณะการบีบอัดที่ได้รับความนิยมในการใช้งานกับไฟล์ข้อมูลทั่วไป

สิ่งที่เป็นตัวกำหนดว่าการบีบอัดข้อมูลจะเป็นแบบไม่สูญเสียก็คืออัลกอริธึมที่เลือกใช้ในการบีบข้อมูล หากอัลกอริธึมที่เลือกใช้เป็นอัลกอริธึมชนิด LZ, LZW, ZIP, Wavelet Transform, RLE, Huffman ข้อมูลที่บีบอัดแล้วก็จะเป็นข้อมูลที่สามารถคืนค่าต้นฉบับกลับมาได้ครบถ้วน

2.1.1.1.2 การบีบอัดข้อมูลแบบที่มีการสูญเสีย

เป็นการลดรูปข้อมูลที่มีประสิทธิภาพสูงมาก สามารถลดปริมาณข้อมูลลงได้หลายเท่าตัว บางครั้งมากกว่าสิบเท่าตัว แต่การบีบข้อมูลลักษณะนี้จะไม่สามารถย้อนกลับขบวนการเพื่อคืนข้อมูลต้นฉบับกลับมาให้เหมือนเดิมได้ เหตุที่สามารถลดปริมาณข้อมูลได้มากเป็นเพราะมีการทิ้งข้อมูลบางส่วนไป ทำให้การคืนรูปข้อมูลทำได้ไม่สมบูรณ์ หรือไม่เหมือนต้นฉบับ

การละทิ้งข้อมูลเกิดจากขบวนการในระหว่างการบีบข้อมูล อย่างเช่นการบีบอัดข้อมูลชนิดเจเป็กจะอาศัยอัลกอริธึม DCT (Discrete Cosine Transform) มาทำการเปลี่ยนลักษณะข้อมูลให้อยู่ในรูปของสัมประสิทธิ์ของ DCT จากนั้นจะทำการ Quantization ซึ่งเป็นการหารด้วยค่าคงที่ค่าหนึ่ง ผลการหารจะปัดเศษทิ้งไป คงเหลือไว้แต่จำนวนเต็ม การปัดเศษในขั้นตอนการแบ่งนัย (Quantization) เป็นสาเหตุที่ทำให้ข้อมูลที่บีบอัดไม่ครบถ้วนเหมือนต้นฉบับ การคืนรูปข้อมูลของ JPEG จึงไม่สามารถทำได้เหมือนต้นฉบับ

2.1.1.2 ประสิทธิภาพการบีบอัดข้อมูล

การหาประสิทธิภาพการบีบอัดข้อมูลจะทำได้โดยการนำปริมาณข้อมูลของไฟล์ต้นฉบับ หารด้วยปริมาณข้อมูลของไฟล์ที่ผ่านการบีบอัดแล้ว [2] ค่าตัวเลขที่ได้จะหมายถึงข้อมูลที่ผ่านการบีบอัดแล้วมีขนาดเล็กลงกี่เท่าเมื่อเทียบกับต้นฉบับ ถ้าประสิทธิภาพมากกว่า 1 แสดงว่าการบีบอัดทำให้ไฟล์มีขนาดเล็กลง ถ้าประสิทธิภาพน้อยกว่า 1 แสดงว่าการบีบอัดไม่ได้ผล ไฟล์ที่ผ่านการบีบอัดมีขนาดใหญ่ขึ้นซึ่งถือว่าการบีบอัดไม่ได้ผล

ยกตัวอย่าง ไฟล์ต้นฉบับมีขนาด 100 กิโลไบต์ ไฟล์ที่ผ่านการบีบอัดแล้วมีขนาดไฟล์ 25 กิโลไบต์

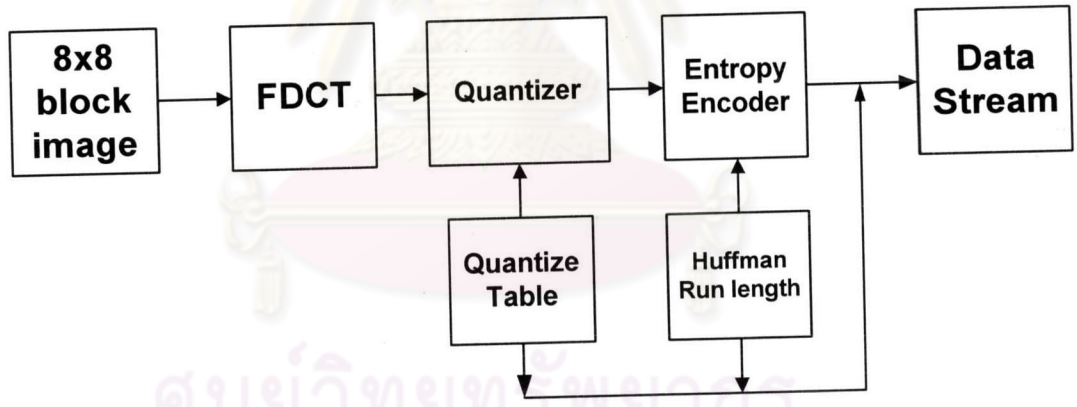
ดังนั้น ประสิทธิภาพการบีบอัดหาได้จาก 100/25 = 4

2.1.2 การบีบอัดข้อมูลแบบต่าง ๆ

2.1.2.1 การบีบข้อมูลแบบเจเป็ก(JPEG)

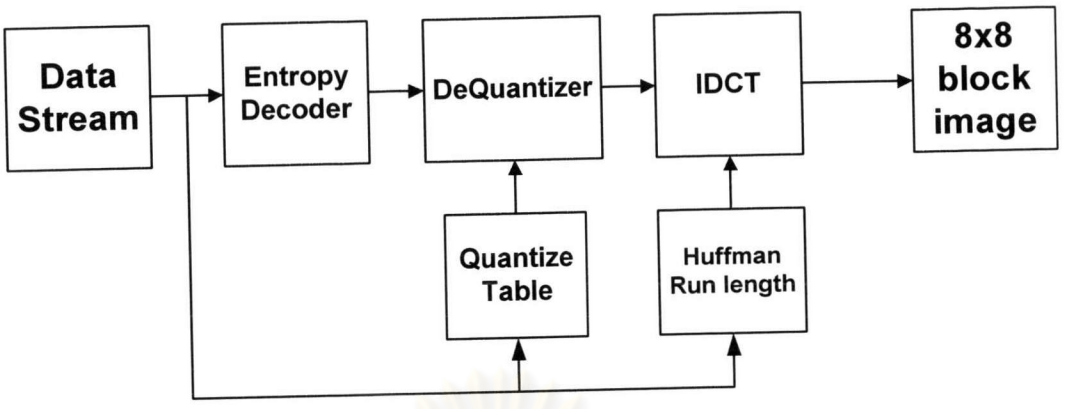
การบีบอัดข้อมูลภาพนิ่งจะใช้ข้อกำหนดของหน่วยงานสากลหน่วยงานหนึ่ง นั่นก็คือ Joint Photographic Experts Group หรือที่เรียกกันว่า JPEG ซึ่งการบีบอัดข้อมูลแบบ เจเป็ก(JPEG) นี้จะเป็นการบีบอัดแบบที่มีการสูญเสีย ภาพที่ได้จากวิธีการนี้จะไม่เหมือนต้นฉบับเดิม หากเปรียบเทียบกันบิตต่อบิต แต่ภาพที่สายตามองเห็นอาจจะมีความคล้ายคลึงกันจนไม่สามารถแยกแยะได้

หลักการคร่าว ๆ ของเจเป็ก ก็คือนำภาพต้นฉบับมาทำการเข้ารหัส ซึ่งข้อมูลดิจิทัลที่ผ่านการเข้ารหัสจะมีขนาดเล็กลง และเมื่อนำข้อมูลที่ผ่านการเข้ารหัสไปใช้งาน ก็จะต้องทำการถอดรหัสกลับมาเสียก่อน ดูขั้นตอนต่าง ๆ ของการเข้ารหัสได้ตามภาพที่ 2-1 และขั้นตอนการถอดรหัสจากภาพที่ 2-2



ภาพที่ 2-1 ขั้นตอนการเข้ารหัสข้อมูลแบบเจเป็ก (JPEG)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



ภาพที่ 2-2 ขั้นตอนการถอดรหัสข้อมูลแบบเจเป็ก

2.1.2.1.1 Discrete Cosine Transform

วิธีการของเจเป็ก จะเป็นการนำการคำนวณทางคณิตศาสตร์ที่ซับซ้อนมากระทำกับชุดข้อมูลดิจิทัลที่เป็นตัวเลข การคำนวณทางคณิตศาสตร์ที่กล่าวถึงนี้คือการทำ Discrete Cosine Transform[3] ซึ่งเป็นสมการทางคณิตศาสตร์ที่มีการค้นพบมานานตั้งแต่ปี ค.ศ. 1974 แต่ได้ถูกนำมาใช้งานกับข้อมูลภาพครั้งแรกในปี ค.ศ. 1984

สมการของ Discrete Cosine Transform จะทำหน้าที่เปลี่ยนลักษณะข้อมูลดิจิทัลที่เป็นตัวเลขซึ่งเป็นตัวแทนของระดับความสว่าง ไปเป็นชุดตัวเลขพิเศษที่เป็นตัวแทนของความถี่ของภาพนั้น ๆ ข้อมูลทั้งสองประเภทนี้สามารถแปลงกลับไปกลับมาโดยที่ยังสามารถคงค่าเดิมเอาไว้ได้โดยไม่มีการเปลี่ยนแปลง และการแปลงด้วยสมการแบบ Forward DCT (FDCT) สามารถย้อนกลับได้ด้วยการทำ Inverse Discrete Cosine Transform (IDCT) ตัวอย่างสมการ FDCT และ IDCT ดูได้จากสมการที่ 2-1 และ 2-2

FDCT $$C(u) = \alpha(u)\sum_{x=0}^{N-1} f(x)\cos\left[\frac{(2x+1)u\pi}{2N}\right]$$ **สมการที่** 2-1

IDCT $$f(x) = \sum_{u=0}^{N-1}\alpha(u)C(u)\cos\left[\frac{(2x+1)u\pi}{2N}\right]$$ **สมการที่** 2-2

$C(u)$ คือ ผลลัพธ์จากการทำ DCT

$f(x)$ คือ ค่าระดับสีเทาของข้อมูล

$\alpha(u)$ คือ สัมประสิทธิ์ของสมการ DCT $\alpha(0) = \sqrt{1/N}$, $\alpha(u) = \sqrt{2/N}$

N คือ จำนวนข้อมูลทั้งหมดที่นำมาประมวลผล DCT

2.1.2.1.2 การแบ่งนัย (Quantization)

การแบ่งนัย (Quantization) จะเป็นการปรับระดับหรือการกำหนดค่าของข้อมูลดิจิทัลด้วยกฎเกณฑ์หนึ่ง ๆ โดยจะต้องทำให้ผลลัพธ์ที่ได้มีค่าอยู่ในช่วงจำกัดหรือเป็นไปตามข้อกำหนดของระบบ[4] เช่นการปัดเศษ การหาร ซึ่งการแบ่งนัย ได้ถูกนำมาใช้ในขั้นตอนของการเข้ารหัส JPEG โดยจะทำการ Quantize หลังจากที่ข้อมูลผ่านขั้นตอน FDCT มาแล้ว ยกตัวอย่างเช่น ตารางแทนค่าข้อมูลดิจิทัลขนาด 8x8 เมื่อผ่านขั้นตอน FDCT แล้วจะยังคงมีโครงสร้างเป็น 8x8 อยู่แต่ตัวเลขภายในตารางจะเปลี่ยนไป จากนั้นก็จะนำตารางที่ได้ไปหารแบบปัดเศษด้วยตารางพิเศษที่กำหนดขึ้นมา เราสามารถเรียกตารางที่กำหนดขึ้นมาใหม่นี้ว่า Quantization Table ผลลัพธ์จากการหารแบบปัดเศษจะทำให้ตัวเลขในตารางมีค่าน้อยลง และบางค่าจะกลายเป็นศูนย์ถ้าผลหารที่ออกมาได้เศษไม่ถึง 0.5 ดูตัวอย่างของ Quantization Table ได้จากภาพที่ 2-3

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	12	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	88	87	103	121	120	101
72	92	95	98	112	100	103	99

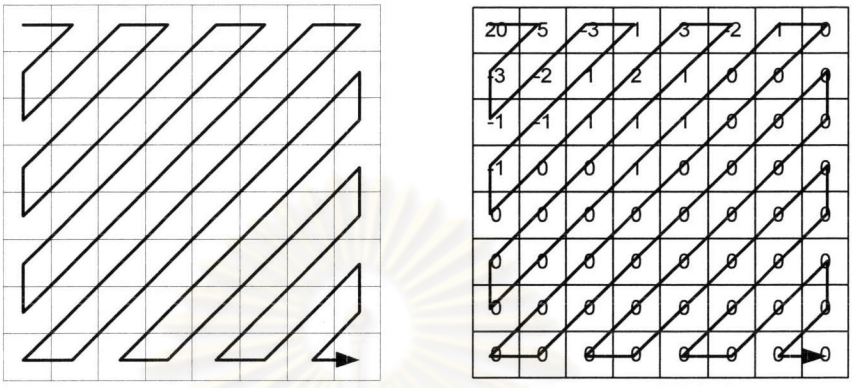
1	2	6	7	15	16	28	29
3	5	8	14	17	27	30	43
4	9	13	18	26	31	42	44
10	12	19	25	32	41	45	54
11	20	24	33	40	46	53	55
21	23	34	39	47	52	56	61
22	35	38	48	51	57	60	62
36	37	49	50	58	59	63	64

ภาพที่ 2-3 ตัวอย่างตารางแบ่งนัย (Quantization Table)

2.1.2.1.3 เอนโทรปี (Entropy)

เป็นขั้นตอนสำหรับการจัดเรียงข้อมูลดิจิทัลโดยมีเจตนาให้มีรูปแบบที่เรียบง่าย หรือมีปริมาณข้อมูลที่ลดน้อยลง ซึ่งอาจจะใช้เทคนิคพิเศษเข้ามาช่วย หรือจะใช้หลายๆเทคนิครวมกันก็ได้ โดยในการเข้ารหัสแบบ JPEG นี้ ในขั้นตอนของการทำเอนโทรปี (Entropy) จะมีเทคนิคที่ใช้รวมกันอยู่ 3 เทคนิคด้วยกัน ประกอบไปด้วย

ก. การกราดแบบซิกแซก(Zigzag Scan) [5] เป็นการเปลี่ยนข้อมูลตารางให้เป็นข้อมูลเรียงแถวไปทีละตัว เริ่มต้นจากมุมด้านหนึ่งแล้วไปจบที่อีกด้านหนึ่งดังภาพที่ 2-4



ภาพที่ 2-4 การกราดแบบซิกแซก

ข. การเข้ารหัสแบบ Runlength [4]

ข้อมูลดิจิทัลที่เรียงตัวกันเป็นแถวจะมีตัวเลขเป็น 0 อยู่เป็นจำนวนมาก การทำ Run length จะทำการเปลี่ยนเลข 0 ที่มักจะมีการวางตัวติดกันหลาย ๆ ตัวให้เป็นรูปแบบที่สั้นลง เช่น 1,0,0,0,0,0,7,0,0,5 จะถูกเปลี่ยนเป็น 0, 1 5, 7 2, 5

ค. การเข้ารหัสแบบ Huffman Coding

เป็นการใช้หลักการของสถิติมาจัดการกับตัวเลข ผลจะทำให้ขนาดข้อมูลลดน้อยลงไปได้อีก

2.1.2.1.4 ตัวอย่างการทำงานของเจเป็ก

ภาพต้นฉบับจะถูกแบ่งให้เป็นตารางขนาด 8x8 หรือ 16x16 เสียก่อน แต่ที่นิยมใช้กันคือขนาด 8x8 ในตัวอย่างนี้จะใช้ขนาด 8x8 ทำการผ่านสมการ FDCT เพื่อเปลี่ยนลักษณะตัวเลข ตามภาพที่ 2-5

183	160	94	153	194	163	132	165
183	153	116	176	187	166	130	169
179	168	171	182	179	170	131	167
177	177	179	177	179	165	131	167
178	178	179	176	182	164	130	171
179	180	180	179	183	169	132	169
179	179	180	182	183	170	129	173
180	179	181	179	181	170	130	169

f(n,m) 8x8 image

FDCT

313	56	-27	18	78	-60	27	-27
-38	-27	13	44	32	-1	-24	-10
-20	-17	10	33	21	-6	-16	-9
-10	-8	9	17	9	-10	-13	1
-6	1	6	4	-3	-7	-5	5
2	3	0	-3	-7	-4	0	3
4	4	-1	-2	-9	0	2	4
3	1	0	-4	-2	-1	3	1

c(k,l) 8x8 DCT coefficients

ภาพที่ 2- 5 ด้านซ้ายคือภาพต้นฉบับ ด้านขวาคือผลจากการผ่านสมการ FDCT

ก. ทำการแบ่งนัยโดยการหารแบบปัดเศษทิ้งด้วยตารางแบ่งนัย ดังภาพที่ 2-6

20	5	-3	1	3	-2	1	0
-3	-2	1	2	1	0	0	0
-1	-1	1	1	1	0	0	0
-1	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

313	56	-27	18	78	-60	27	-27
-38	-27	13	44	32	-1	-24	-10
-20	-17	10	33	21	-6	-16	-9
-10	-8	9	17	9	-10	-13	1
-6	1	6	4	-3	-7	-5	5
2	3	0	-3	-7	-4	0	3
4	4	-1	-2	-9	0	2	4
3	1	0	-4	-2	-1	3	1

=

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	12	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	88	87	103	121	120	101
72	92	95	98	112	100	103	99

ภาพที่ 2-6 การหารด้วยตารางแบ่งนัยซึ่งเป็นการหารแบบปัดเศษ

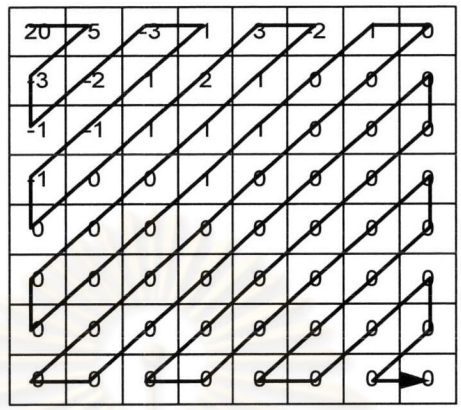
$$\hat{C}(k,l) = NINT\left[\frac{C(k,l)}{T(k,l)}\right]$$

สมการที่ 2-3

C คือ สมาชิกในตารางข้อมูลที่ผ่านการทำ DCTมา, T คือสมาชิกในตารางแบ่งนัย

$\hat{C}$ คือ ผลการหารระหว่างสมาชิก DCT และ สมาชิกในตารางแบ่งนัย

ข.นำผลหารที่ได้มาทำการกราดแบบซิกแซก โดยจะทำการเปลี่ยนเลขในตารางให้เรียงตัวกันเป็นแถวยาว ดังภาพที่ 2-7



[20,5,-3,-1,-2,-3,1,1,-1,-1,0,0,1,2,3,-2,1,1,1,0,0,0,0,0,0,1,1,0,1 EOB]

ภาพที่ 2-7 การกราดแบบซิกแซกจะทำการเรียงตัวเลขให้เป็นแถวยาวเพียงแถวเดียว

ค.เมื่อได้ตัวเลขเรียงแถวมาแล้วก็จะนำไปผ่านขบวนการ Run length และ Huffman ต่อไป จากนั้นก็จะได้ไฟล์ข้อมูลที่เป็นแบบ JPEG

2.1.2.1.5 สรุปการทำงานของเจเป็ก

การบีบอัดข้อมูลภาพนิ่งแบบ JPEG เป็นการบีบข้อมูลแบบที่มีการสูญเสีย โดยภาพต้นฉบับจะผ่านขั้นตอนต่าง ๆ ดังนี้

ก. การ Transform ด้วย Discrete Cosine Transform จะเป็นการเปลี่ยนตัวเลขของข้อมูลให้อยู่ในรูปแบบใหม่ แต่ยังไม่มีการสูญเสียใด ๆ เกิดขึ้น

ข. การแบ่งนัย จะเป็นการหารปัดเศษซึ่งเป็นขบวนการปรับค่าข้อมูล ที่ขั้นตอนนี้จะทำให้เกิดการสูญเสีย ไม่สามารถคืนค่าเดิมของข้อมูลกลับมาได้อีก ทำได้เพียงใกล้เคียงเท่านั้น

ค. การทำ กราดแบบซิกแซก, Run length และ Huffman เป็นการจัดรูปแบบข้อมูลที่ไม่มีการสูญเสีย เพียงแต่ทำให้ปริมาณข้อมูลลดน้อยลง

จากหลักการทั้งหมดของการเข้ารหัสแบบเจเป็ก และตัวอย่างที่แสดงให้เห็นทุกขั้นตอนจะทำให้เราเข้าใจหลักการทำงานทั้งหมด และทราบว่าขั้นตอนใดเป็นสาเหตุที่ทำให้เกิดการสูญเสีย การจะทำให้ข้อมูลของเจเป็ก เกิดการสูญเสียให้น้อยลงนั้นสามารถทำได้โดยการเลือกค่าของตารางแบ่งนัยให้ผลหารเกิดการปัดเศษทิ้งให้น้อยที่สุด แต่ก็จะมีผลทำให้ปริมาณ

ข้อมูลไม่ลดน้อยลงเท่าที่ควร การเลือกใช้งานเจเป็ก ที่ระดับการสูญเสียต่างๆนั้น จะต้องพิจารณาให้รอบคอบว่าเราต้องการคุณภาพของภาพมาก ๆ หรือเน้นเพียงขนาดข้อมูลที่เล็กที่สุด

2.1.2.2 **การบีบข้อมูลแบบพรีฟิกส์** (PREFIX COMPRESSION)

การบีบอัดข้อมูลภาพได้รับการพัฒนากันในหลายแนวทาง โดยทุกวิธีที่คิดค้นขึ้นมานั้นสามารถช่วยลดขนาดของข้อมูลลงได้แต่แตกต่างกันในด้านประสิทธิภาพและคุณภาพของภาพต้นฉบับ การบีบอัดข้อมูลที่ได้รับความนิยมอย่างกว้างขวางที่สุดเป็นการบีบอัดตามมาตรฐานของเจเป็ก ซึ่งเหมาะสมกับลักษณะภาพที่เป็นภาพถ่าย มีสีสันเป็นองค์ประกอบสามสี คือแดง เขียว และน้ำเงินเป็นหลัก ซึ่งหากนำวิธีการของเจเป็ก ไปบีบอัดภาพที่เป็นเอกสารตัวหนังสือก็มักจะลดขนาดได้เพียงเล็กน้อยเท่านั้น

เนื่องจากเอกสารที่ต้องการเก็บบันทึกไว้ในรูปแบบดิจิทัลมักจะมีลักษณะภาพที่เป็นภาพสีเดียว หรือภาพขาวดำ จึงมีการคิดค้นวิธีการบีบอัดวิธีการใหม่ขึ้นมาโดยเฉพาะ ซึ่งจะทำให้ประสิทธิภาพการบีบข้อมูลอยู่ในระดับที่สูงมาก (บีบอัดได้มาก เหลือขนาดข้อมูลน้อย) นั่นคือวิธีการที่มีชื่อว่าการบีบอัดแบบพรีฟิกส์

ข้อกำหนดเบื้องต้นของการบีบอัดแบบพรีฟิกส์ นั้นจะต้องทำงานกับภาพดิจิทัลที่มีลักษณะสองสี คือ ขาวกับดำเท่านั้น ซึ่งเป็นลักษณะของตัวหนังสือเป็นส่วนใหญ่ และสีดำและขาวนั้นจะต้องมีความเข้มคงที่ ก็คือ ขาว กับ ดำ ไม่มีสีเทา ไม่มีการไล่โทนสี ซึ่งจะเป็นลักษณะเหมือนกับไฟล์ข้อมูลชนิด Bitmap 1 บิต แต่ละจุดภาพจะมีค่าเพียงขาวหรือดำเท่านั้น

แรงจูงใจและแนวทางการพัฒนาของการบีบข้อมูลแบบพรีฟิกส์ จะมีอยู่ว่าเอกสารที่เป็นขาวดำ ไม่ว่าจะอยู่บนกระดาษขนาดใดก็ตาม ถ้าเป็นข้อความเดิมก็ควรจะบันทึกเก็บเป็นดิจิทัลด้วยขนาดที่คงเดิม เช่น ลายเซ็นชื่อของคน ๆ หนึ่งบนกระดาษขนาดนามบัตร กับลายเซ็นของคน ๆ เดิมบนกระดาษขนาด A4 ควรจะมีขนาดข้อมูลที่เท่ากัน เพราะเป็นลายเซ็นคนเดิม หรือเป็นภาพเนื้อหาสาระเดียวกันแต่อยู่บนกระดาษที่มีขนาดแตกต่างกันเท่านั้น ซึ่งหากนำลายเซ็นบนนามบัตรและบนกระดาษ A4 มาจัดเก็บด้วยวิธีการของเจเป็กก็จะได้ขนาดไฟล์ที่ใหญ่ไม่เท่ากัน นั่นก็คือภาพลายเซ็นบนนามบัตรจะมีขนาดเล็ก ไฟล์ที่ได้ก็จะมีขนาดเล็ก แต่ภาพลายเซ็นบนกระดาษ A4 จะมีขนาดใหญ่ ไฟล์เจเป็กก็จะใหญ่กว่ากันหลายเท่าตัว นั่นเป็นสาเหตุที่การบีบอัดแบบเจเป็ก ไม่เหมาะกับภาพลายเส้นขาวดำ และทำให้เกิดการพัฒนาวิธีการของการบีบอัดแบบพรีฟิกส์ขึ้นมา

2.1.2.2.1 ลักษณะภาพที่เหมาะกับการบีบอัดแบบพรีฟิกส์

ภาพที่มีลักษณะเป็นสองสี ขาวกับดำ และเนื้อหาสาระของสีดำหรือพื้นที่สีดำน้อยกว่าสีขาวมาก ๆ เช่นภาพของเอกสารที่สแกนเป็นดิจิทัล มีพื้นที่ตัวหนังสือสีดำน้อยกว่าพื้นที่ของกระดาษทั้งใบหลายสิบเท่า ภาพเหล่านี้เหมาะสมอย่างยิ่งที่จะใช้วิธีการบีบอัดแบบ Prefix Compression ตัวอย่างได้แก่ เอกสารขาวดำ กระดาษแฟกซ์

2.1.2.2.2 หลักการทำงานของพรีฟิกส์

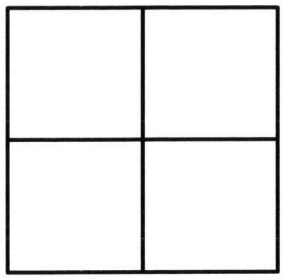
ภาพลักษณะขาวและดำจะถูกแทนค่าเป็นข้อมูลดิจิทัลซึ่งจะมีค่าเป็น 0 และ 1 เท่านั้น ซึ่งตัวเลข 0 จะเป็นการแทนภาพที่ไม่มีสี หรือเป็นสีขาว ส่วนตัวเลข 1 จะเป็นการแทนค่าของส่วนที่เป็นสีดำ ดังนั้นในเอกสารหน้าหนึ่งๆเมื่อถูกสแกนและเปลี่ยนข้อมูลให้เป็นดิจิทัลแล้ว ส่วนที่เป็นสีดำจะถูกกำหนดค่าให้เป็น 1 ส่วนพื้นที่สีขาวจะถูกกำหนดค่าให้เป็น 0 ซึ่งกระบวนการบีบอัดจะทำการเก็บข้อมูลเฉพาะส่วนที่เป็นสีดำหรือค่า 1 เท่านั้น ส่วนพื้นที่สีขาวหรือค่า 0 จะไม่สนใจ ทำให้เราสามารถลดขนาดข้อมูลลงได้เป็นจำนวนมาก

เริ่มจากภาพต้นฉบับที่มีอยู่ดังนี้สมมุตให้พื้นที่กระดาษหรือพื้นที่ทั้งหมดของภาพเป็นสี่เหลี่ยมดังภาพที่ 2-8

ภาพที่ 2-8 ภาพต้นฉบับ

จะทำการแบ่งโซนในภาพใหญ่ให้เป็นพื้นที่เล็ก ๆ ซึ่งจะแบ่งมากหรือน้อยจะขึ้นอยู่กับความละเอียดที่ต้องการ โดยการแบ่งพื้นที่จะแบ่งเป็นสี่เหลี่ยมจัตุรัสยาวด้านละ 2 ยกกำลัง N ไปเรื่อย ๆ เช่นภาพตัวอย่างถูกแบ่งเป็นด้านละสองส่วน หรือ 2 ยกกำลัง 1 ก็คือด้านละ 2 หน่วย หรือมีพื้นที่คิดเป็น 4 หน่วยดังภาพที่ 2-9

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

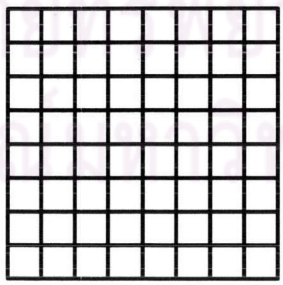


ภาพที่ 2-9 ภาพตัวอย่างถูกแบ่งเป็นด้านละสองส่วน

ถ้าแบ่งมากขึ้นเป็น 2 ยกกำลัง 2 หรือด้านละ 4 หน่วย พื้นที่ของภาพก็จะเป็น 16 หน่วยดังภาพที่ 2-10

ภาพที่ 2-10 ภาพตัวอย่างถูกแบ่งเป็นด้านละสี่ส่วน

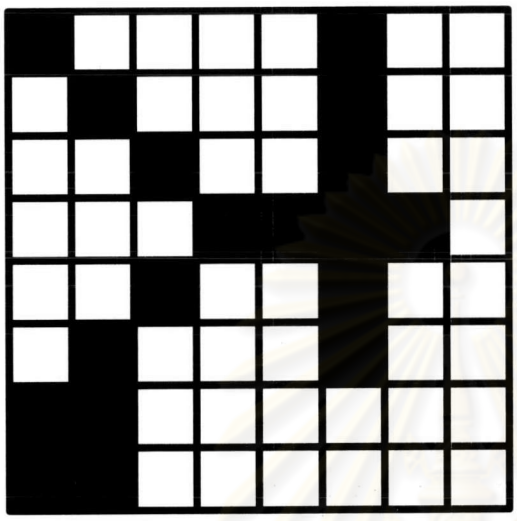
ถ้าแบ่งเป็น 2 ยกกำลัง 3 หรือด้านละ 8 หน่วย พื้นที่ของภาพทั้งหมดก็จะเป็น 64 หน่วยดังภาพที่ 2-11



ภาพที่ 2-11 ภาพตัวอย่างถูกแบ่งเป็นด้านละแปดส่วน

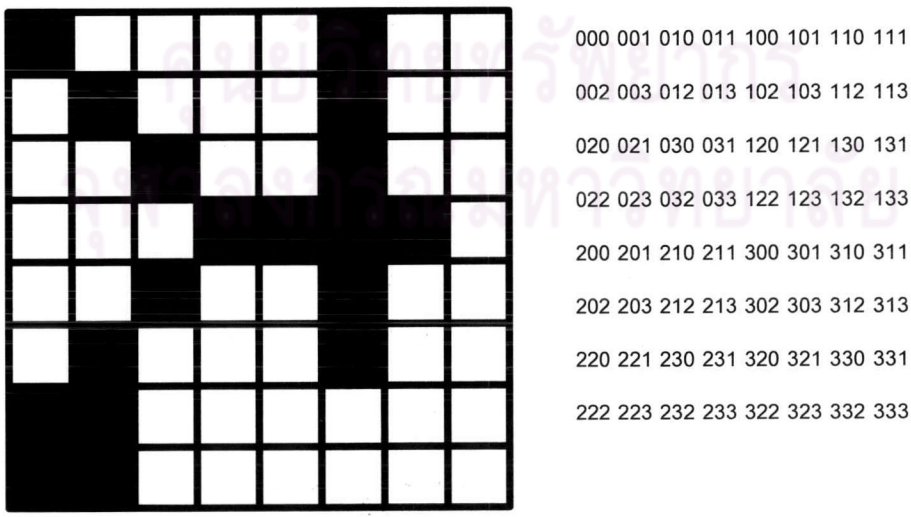
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ยกตัวอย่างภาพที่ได้รับการแบ่งพื้นที่เป็นด้านละ 8 หน่วย สีดำแทนตัวหนังสือ สีขาวแทนกระดาษ เมื่อแบ่งพื้นที่เป็นตารางแล้วแทนค่าสีดำด้วย 1 และสีขาวแทนด้วย 0 (ภาพที่ 2-12)



ภาพที่ 2-12 ภาพที่ถูกแบ่งเป็นช่อง ๆ และกำหนดค่าให้เป็นดิจิทัล

จากนั้นตำแหน่งของตารางจะถูกกำหนดขึ้นมาให้เป็นตัวเลขจำนวน N บิต ซึ่ง N ก็คือตัวเลขจำนวนที่ยกกำลัง จากตัวอย่างที่ใช้ 2 ยกกำลัง 3 นั่นคือ N=3 ตำแหน่งของแต่ละช่องเล็ก ๆ ในตารางจะถูกแทนที่ด้วยเลขจำนวน 3 บิต



ภาพที่ 2-13 ใช้ตัวเลข 3 บิตแทนค่าภาพในแต่ละช่อง

จากนั้นก็ทำการนำตัวเลขของตำแหน่งที่เป็นสีดำออกมาประมวลผลต่อ โดยแยกเขียนออกมาเป็นชุด 000, 101, 003, 103, 030, 121, 033, 122, 123, 132, 210, 301, 203, 303, 220, 221, 222, 223 ถ้าสังเกตให้ดี ตัวเลขนำหน้า หรือ Prefix จะมีเป็นกลุ่มๆ นั่นคือกลุ่มที่ขึ้นต้นด้วย 00 10 03 12 13 21 30 20 22 ตามลำดับ ซึ่งจะทำการเรียบเรียงใหม่ให้อยู่ในรูป **Prefix | counter | Suffix**

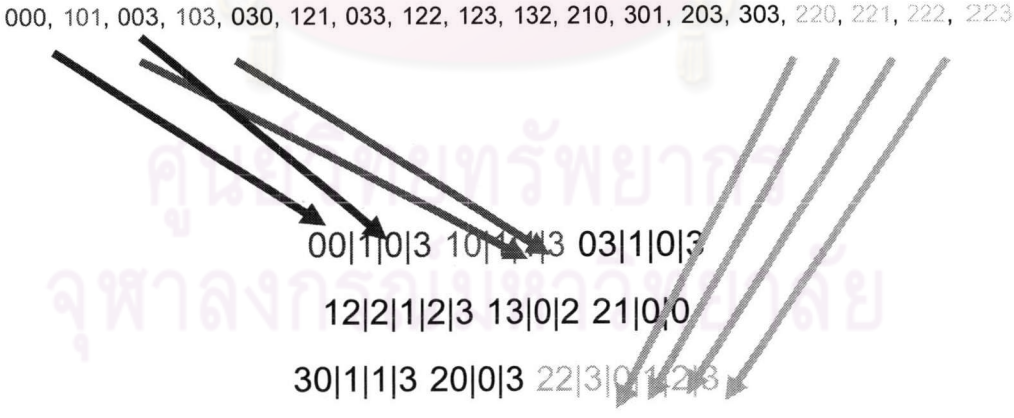
Prefix หมายถึงตัวเลขที่เป็นตัวแทนกลุ่มด้านหน้าสุดจำนวนหลักตามข้อกำหนด ในตัวอย่างนี้จะเป็นสองหลัก

Counter เป็นตัวเลขแทนจำนวนสมาชิกที่มี Prefix เหมือนกันลบด้วย 1

Suffix คือตัวเลขที่ตามหลัง Prefix ของสมาชิกทุกตัวที่ใช้ prefix ร่วมกัน

ตัวอย่าง 000, 003 จะมี Prefix ร่วมกันคือ 00 ดังนั้น Prefix คือ 00 มีจำนวนสมาชิก 2 ตัว ดังนั้น counter คือ 2-1 มีตัวเลขที่ตามหลัง prefix จากสมาชิกทุกตัว ดังนั้น suffix คือ 0 และ 3 เขียนใหม่ได้เป็น 00|1|0|3

ดังนั้นตัวเลขทั้งกลุ่มก็จะเขียนออกมาได้ดังภาพที่ 2- 14



ภาพที่ 2-14 การนำข้อมูลมาจัดเรียงตามกลุ่ม Prefix

จากนั้นเปลี่ยนตัวเลขทั้งชุดนี้ให้เป็นเลขฐานสอง

เลข	0	จะถูกแทนที่ด้วย	00
เลข	1	จะถูกแทนที่ด้วย	01
เลข	2	จะถูกแทนที่ด้วย	10
เลข	3	จะถูกแทนที่ด้วย	11

ดังแสดงในภาพที่ 2-15

00|1|0|3 10|1|1|3 03|1|0|3
12|2|1|2|3 13|0|2 21|0|0
30|1|1|3 20|0|3 22|3|0|1|2|3

⬇

0000010011 0100010111 0011010011
011010011011 01110010 10010000
1100010111 10000011 0101100011011

ภาพที่ 2-15 เปลี่ยนเลขฐานสิบเป็นเลขฐานสอง

จะเห็นว่าข้อมูลส่วนที่เป็นสีดำจะถูกเปลี่ยนให้เป็นตัวเลขดิจิทัลที่สามารถจัดเก็บได้โดยใช้พื้นที่เพียงเล็กน้อยเท่านั้น ไม่มีการเก็บข้อมูลของส่วนที่เป็นสีขาวเลย ทำให้ขนาดของไฟล์ข้อมูลดิจิทัลที่ได้มีขนาดเล็กมากเมื่อเทียบกับไฟล์ดิจิทัลแบบอื่น ๆ โดยเฉพาะกับแบบของ JPEG เนื่องจาก JPEG จะต้องเก็บทุกอย่างนั่นเองไม่ว่าจะเป็นสีดำหรือสีขาว

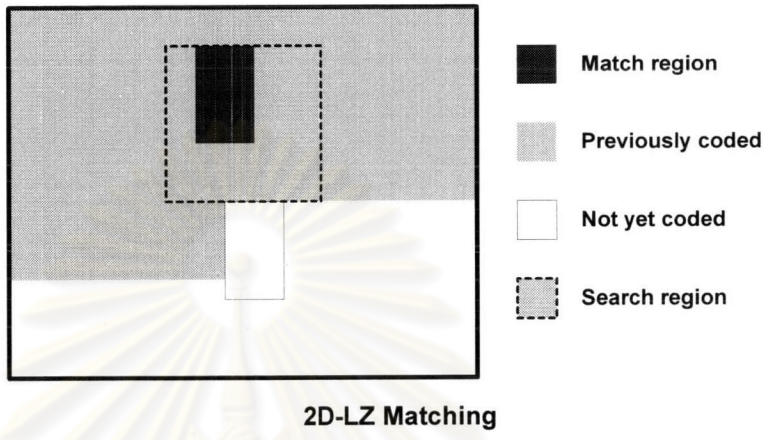
2.1.2.2.3 สรุปการบีบอัดแบบพรีฟิกส์

การบีบอัดข้อมูลภาพพรีฟิกส์ เป็นการบีบอัดข้อมูลชนิดหนึ่งซึ่งจะต้องใช้งานกับภาพต้นฉบับที่เป็นสีขาวและดำเท่านั้น ไม่สามารถใช้งานกับภาพที่มีการไล่ระดับสีเทาได้ และจะมีประสิทธิภาพการบีบอัดที่สูงมาก (ทำให้ไฟล์ข้อมูลเหลือขนาดเล็กลง) เมื่อใช้งานกับต้นฉบับมีปริมาณของสีดำน้อยมาก เมื่อเทียบกับสีขาว

2.1.2.3 **การบีบอัดข้อมูลแบบ** 2D-LZ

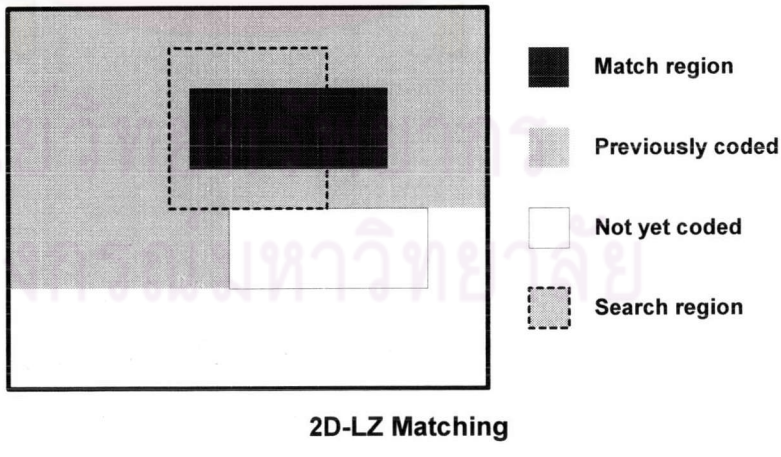
การทำงานของ 2D-LZ จะเริ่มจากการตรวจสอบข้อมูลปัจจุบันที่กำลังพบอยู่ว่าเป็นข้อมูลที่เคยพบมาหรือไม่ โดยจะทำการย้อนกลับไปดูข้อมูลเก่าที่เคยผ่านมาแล้ว ถ้าพบว่า

มีข้อมูลเก่าที่เคยประมวลผลผ่านมาแล้วก็จะทำการบันทึกตำแหน่งของข้อมูลเก่าที่พบแทน แต่ถ้าไม่ตรงกับข้อมูลที่เคยประมวลผลมาก็จะบันทึกข้อมูลใหม่เก็บไว้ การย้อนกลับไปตรวจสอบข้อมูลเก่าจะกระทำภายในขอบเขตที่กำหนดประมาณ 256x256 จุด ดูภาพที่ 2-16 ประกอบ



ภาพที่ 2-16 การทำงานของ 2D-LZ

การปรับปรุงการทำงานของ 2D-LZ ให้มีประสิทธิภาพการบีบอัดสูงขึ้นจะทำการแก้ไขในขั้นตอนการตรวจสอบข้อมูลเก่าโดยในวิธีเดิมจะตรวจสอบย้อนกลับไปในขอบเขตที่กำหนดคือ 256x256 จุดภาพ แต่ในวิธีการที่ปรับปรุงแล้วจะขยายขอบเขตการตรวจสอบให้กว้างขึ้นถ้ามีข้อมูลที่ซ้ำกันและอยู่รวมกันมีพื้นที่ใหญ่กว่าขอบเขตที่กำหนด ดูภาพที่ 2-17 ประกอบ



ภาพที่ 2-17 การขยายขอบเขตการตรวจสอบข้อมูลเก่าของ 2D-LZ

จากการปรับปรุงการทำงานของ 2D-LZ ทำให้วิธีการใหม่นี้สามารถเพิ่มอัตราส่วนการบีบอัดข้อมูลได้สูงขึ้น โดยผลการทดลองกับข้อมูลภาพตัวอย่างกับวิธีการบีบอัดแบบต่าง ๆ ดังภาพที่ 2-18 ช่วยยืนยันได้ว่าการปรับปรุงการบีบอัดแบบ 2D-LZ นั้นได้ผลการบีบอัดที่ดีขึ้น

Data	JBIG	ZIP	2D-LZ	2D-LZ ZIP
Sample1	58.7	88	94.6	171
Sample2	9.77	47.9	22.9	43.8
Sample3	12.4	50.7	35.8	64.9

ภาพที่ 2-18 ผลการทดลองกับภาพตัวอย่างกับวิธีการบีบอัดหลาย ๆ วิธี

จากภาพที่ 2-18 ตัวเลขที่เขียนไว้ในตารางเป็นอัตราส่วนการบีบอัดที่คำนวนจากข้อมูลต้นฉบับหารด้วยข้อมูลภาพที่ผ่านการบีบอัดแล้ว เป้าหมายของการวิจัยคือการพัฒนาการบีบอัดให้ได้อัตราส่วนการบีบอัดมากกว่า 25 เท่า ตัวเลขในตารางที่ต่ำกว่า 25 หมายถึงยังไม่ดี วิธีการ JBIG ได้ผลดีกับบางภาพแต่กับภาพส่วนใหญ่แล้วยังทำได้ไม่ดี ส่วน ZIP จะได้ผลดีกับภาพโดยส่วนใหญ่ ส่วนวิธีการ 2D-LZ จะได้ผลกับภาพส่วนใหญ่เช่นกัน แต่จะมีปัญหากับภาพเพียงบางภาพ ซึ่งหมายถึงจะต้องมีการพัฒนาการบีบอัดของ 2D-LZ ให้ดีขึ้นไปกว่านี้อีก ส่วนวิธีการ 2D-LZ ZIP ซึ่งเป็นวิธี 2D-LZ แบบที่ใช้งานร่วมกับ ZIP ด้วยนั้นจะได้ผลดีมากกับภาพทุกภาพ แต่ด้วยเงื่อนไขของเวลาการถอดรหัสที่จำเป็นจะต้องทำให้เสร็จสิ้นอย่างรวดเร็วทำให้วิธีการ 2D-LZ ZIP ไม่สามารถใช้กับระบบของ imagesetter นี้ได้ เนื่องจากใช้เวลาการคลายข้อมูลที่นานเกินไป ส่วนวิธีการ ZIP ก็ใช้เวลาการคลายข้อมูลที่นานเช่นกัน

2.1.2.3.1 สรุป 2D-LZ

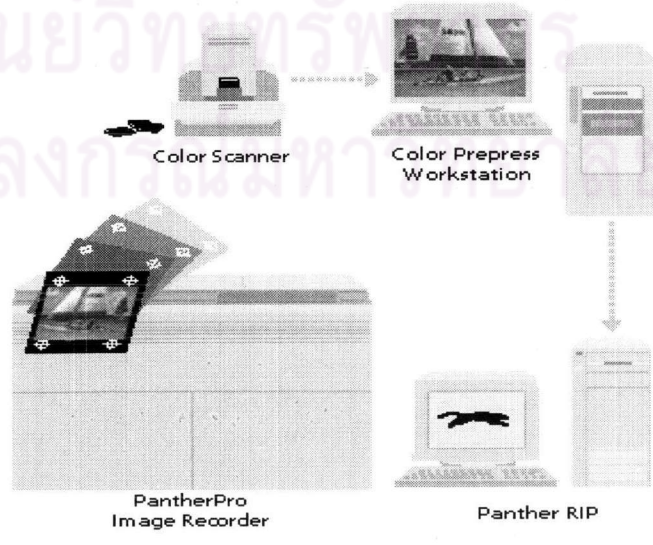
สรุปได้ว่าวิธีการบีบอัดและคลายข้อมูลแบบ 2D-LZ เป็นวิธีการที่ใช้งานได้ผลกับภาพเกือบทุกลักษณะ มีเพียงบางตัวอย่างภาพเท่านั้นที่ยังบีบอัดให้เล็กลงตามอัตราส่วน 25 เท่าไม่ได้ จำเป็นต้องมีการพัฒนาต่อไป

2.1.3 องค์ประกอบของภาพฮาล์ฟโทน

ในโปรแกรมสำหรับจัดหน้าเราสามารถที่จะเลือกวางไฟล์ภาพต่าง ๆ ลงในตัวงานได้ ไฟล์ภาพเหล่านี้อาจจะได้จากการสแกนฟิล์ม หรืออาจจะเป็นภาพที่ได้จากกล้องดิจิทัลโดยตรง การเลือกวางรูปภาพลงในชิ้นงานสามารถจะวางคู่ไปกับตัวหนังสือและสัญลักษณ์อื่น ๆ ได้ตามต้องการ เมื่อจัดวางสิ่งต่าง ๆ ได้เรียบร้อยแล้วเราก็บันทึกข้อมูลนี้ลงในสื่อบันทึกข้อมูล ซึ่งจะถูกนำไปสร้างเป็นแผ่นฟิล์มสำหรับทำเพลตต่อไป หรือในบางแห่งที่ไม่มีการใช้แผ่นฟิล์มแล้วก็จะนำข้อมูลนี้ไปทำเพลตทันที เพลตที่สร้างเสร็จแล้วจะเป็นแม่พิมพ์สำหรับระบบการพิมพ์ออฟเซต ซึ่งสุดท้ายก็จะได้เป็นชิ้นงาน เป็นเอกสาร เป็นหนังสือ เป็นแผ่นพับตามต้องการ

ภาพที่นำมาจัดวางในโปรแกรมจัดหน้าจะเป็นภาพที่มีลักษณะเหมือนรูปถ่าย กล่าวคือเป็นภาพที่มีสีสัน มีการไล่ระดับสีเข้มไปอ่อนและอ่อนไปเข้มอย่างต่อเนื่อง เราจะเรียกภาพเหล่านี้ว่าเป็นภาพชนิดน้ำหนักสีต่อเนื่อง (continuous tone) อย่างเช่นภาพของผลแอ็ปเปิ้ลจะเป็นภาพที่ประกอบไปด้วยองค์ประกอบของสีแดงเป็นหลัก และมีสีอื่น ๆ ปะปนอยู่บ้างเล็กน้อย เช่นสีแดงเข้ม แดงอ่อน แดงอมเขียว ส่วนผลไม้อย่างมะเขือเทศ ก็มีสีแดงเป็นสีหลัก แต่จะเป็นแดงคนละแบบกับแอ็ปเปิ้ล และสีอื่น ๆ ที่ปะปนอยู่ในผลมะเขือเทศก็อาจจะมีสีแดงอมเหลือง แดงอมส้ม ส่วนภาพถ่ายที่เป็นภาพขาวดำ ก็จะเป็นภาพที่มีการไล่ระดับสีดำไปถึงสีเทาและไปสีขาวอย่างต่อเนื่อง

ขั้นตอนการเตรียมพิมพ์จะเป็นไปดังภาพ2-19 โดยเริ่มจาก การสแกนรูปด้วยเครื่องสแกนเนอร์ เครื่องสแกนเนอร์จะกราดแสงไปตกกระทบกับฟิล์มหรือรูป แล้วเปลี่ยนค่าแสงสะท้อนจากฟิล์มให้เป็นสัญญาณสี แดง เขียว น้ำเงิน เพื่อให้สามารถแสดงผลบนจอภาพได้



ภาพที่ 2-19 ขั้นตอนในการเตรียมพิมพ์

เมื่อได้ภาพบนจอแล้ว เราสามารถปรับแต่งภาพต้นฉบับให้ได้ผลลัพธ์ที่ต้องการโดยโปรแกรมจัดการกับภาพ ซึ่งอาจจะเป็นโปรแกรม Adobe Photoshop จากนั้นก็จะใช้โปรแกรมสำหรับวางหน้าเอกสาร ซึ่งอาจจะเป็นโปรแกรม Page Maker ทำการจัดวางรูปภาพและตัวหนังสือเข้าไว้ด้วยกัน เมื่อได้รูปแบบที่ต้องการแล้วก็ทำการพิมพ์เพื่อให้โปรแกรมจัดหน้าทำการแยกสีออกมาเป็นสีของการทำแม่พิมพ์ โปรแกรมจัดหน้าก็จะทำการแปลงสีจากแม่สีหลัก แดง เขียว น้ำเงิน ไปเป็นสีสำหรับงานพิมพ์ก็คือ ฟ้า ชมพู เหลือง ดำ หรือ CMYK (**C**yan **M**agenta **Y**ellow **B**lack) ซึ่งเป็นขั้นตอนของการแยกสี

เมื่อแยกสีออกมาได้เป็นไฟล์สี่สีที่เป็น CMYK แล้ว ไฟล์ข้อมูลเหล่านี้จะถูกส่งไปยังเครื่องริบ หรือ RIP (Raster image Processor) ซึ่งที่เครื่องนี้จะทำการเปลี่ยนข้อมูลสี CMYK ที่เป็นภาพชนิดโทนสีต่อเนื่อง ให้เป็นภาพหนึ่งบิตทิฟฟ์ ซึ่งเป็นภาพที่มีลักษณะเป็นจุดดำเล็ก ๆ หลายจุดประกอบกันจนเป็นภาพ แต่ละจุดภาพจะแทนด้วยข้อมูลดิจิทัลแบบ 0 และ 1 เท่านั้น จากนั้นข้อมูลที่ผ่านเครื่องริบนี้จะถูกส่งไปยังเครื่องสร้างแม่พิมพ์ ซึ่งอาจจะเป็นเครื่องยิงฟิล์ม หรือเครื่องยิงเพลต ข้อมูล 0 และ 1 จะแทนคำสั่งในการเปิดปิดแสงเลเซอร์เพื่อทำการสร้างภาพบนแผ่นฟิล์มหรือเพลต ซึ่งเมื่อนำแม่พิมพ์มาทำการพิมพ์ เราจะได้ภาพพิมพ์ที่มีความเหมือนภาพต้นฉบับกลับคืนมา

2.1.3.1 **การสร้างภาพฮาล์ฟโทน**

ภาพถ่ายที่เป็นภาพขาวดำจะเป็นภาพที่มีการไล่ระดับความดำได้อย่างต่อเนื่อง มีเฉดสีที่เริ่มต้นจากขาวไล่ไปเป็นสีเทาหลายระดับ และสุดท้ายกลายเป็นสีดำ หากจะแบ่งแยกย่อยออกมา เราอาจจะแบ่งระดับสีเทาของภาพถ่ายขาวดำได้ละเอียดถึง 3 ล้านระดับ แต่ในงานพิมพ์กลับมีหมึกสีให้ใช้กันเพียง 4 สีเท่านั้น คือ ฟ้า (Cyan) ชมพู (Magenta) เหลือง (Yellow) ดำ(Black)

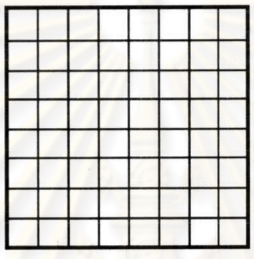
ภาพถ่ายขาวดำต้องการเพียงหมึกสีดำเท่านั้นสำหรับการพิมพ์ การสร้างระดับสีเข้มอ่อนจะอาศัยการสร้างเม็ดสกรีนลงไปในภาพ การสร้างเม็ดสกรีนนี้จะเป็นการแบ่งภาพให้เป็นภาพเล็กๆหลายๆภาพ แต่ละภาพเล็กจะมีจุดภาพเล็ก ๆ ประกอบอยู่ภายในหลาย ๆ จุด ขนาดของจุด และรูปร่างของจุดที่เกิดขึ้นอยู่ในภาพเล็ก ๆ นี้จะเป็นตัวควบคุมระดับความเข้มอ่อนของภาพ

ถ้าจุดภาพเล็ก ๆ เหล่านี้มีขนาดที่ใหญ่โต และมีการวางตำแหน่งไว้ใกล้ ๆ กันมาก ๆ ก็จะได้ภาพที่มีสีเข้มหรือดำ จุดภาพที่อยู่ห่างกันจะให้ภาพที่เป็นสีเทา ส่วนจุดภาพที่มีขนาดเล็กและจัดวางอยู่ห่างกันมาก ๆ จะทำให้ภาพเป็นสีขาว

การสร้างภาพฮาล์ฟโทนจะเป็นหน้าที่ของเครื่องอิมเมจเซ็ตเตอร์ โปรแกรมที่สั่งการสร้างภาพฮาล์ฟโทนจะทำหน้าที่สร้างรูปแบบเม็ดสกรีน ซึ่งอยู่ในรูปแบบของอิเล็คทรอนิกส์ ภาพฮาล์ฟโทนที่ได้จะมีองค์ประกอบภายในดังต่อไปนี้

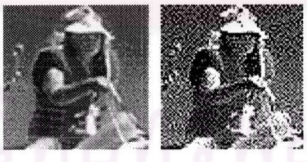
2.1.3.1.1 ความถี่เม็ดสกรีน

ความถี่เม็ดสกรีนจะมีหน่วยวัดเป็นเส้นต่อนิ้ว (line per inch) หรือ lpi ความถี่เม็ดสกรีนนี้จะเกิดจากการแบ่งภาพใหญ่ให้เป็นช่องเล็ก ๆ หลาย ๆ ช่องต่อกัน ลักษณะเหมือนเป็นตารางแบ่งภาพ แต่ละช่องในตารางเรียกว่า ฮาล์ฟโทนเซลล์ (halftone cell) แต่ละฮาล์ฟโทนเซลล์จะประกอบไปด้วยจุดเล็ก ๆ หรือฮาล์ฟโทนด็อต (halftone dot)



ภาพที่ 2-20 เส้นตารางที่แบ่งภาพใหญ่ให้เป็นภาพเล็กๆหลายๆภาพ

ยิ่งมีความถี่เม็ดสกรีนมาก ๆ ภาพก็ยิ่งมีรายละเอียดมาก ในขณะที่ความถี่เม็ดสกรีนน้อย ๆ ภาพก็มีความละเอียดน้อย



ภาพที่ 2-21 ภาพตัวอย่างของการใช้ความถี่เม็ดสกรีนมาก (ซ้าย) และความถี่เม็ดสกรีนน้อย (ขวา)

ภาพตัวอย่างทางซ้ายมือเป็นภาพที่มีความถี่เม็ดสกรีนที่สูงกว่าภาพทางขวา เราจะเห็นได้ว่าภาพทางซ้ายมีรายละเอียดที่เยอะกว่าอย่างเห็นได้ชัด

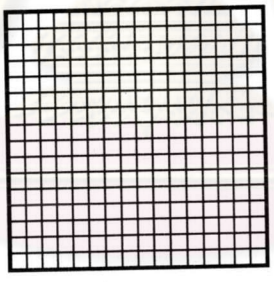
ความถี่เม็ดสกรีนจะเป็นเท่าไร ขึ้นอยู่กับการนำไปใช้งาน โดยเฉพาะวัสดุพิมพ์หรือกระดาษจะเป็นปัจจัยหลักที่เกี่ยวข้องกับความถี่เม็ดสกรีน หากจะทำงานพิมพ์บนกระดาษหนังสือพิมพ์ทั่วไป เราอาจจะใช้ภาพที่มีความถี่เม็ดสกรีนเพียง 85-100 เส้น(lpi) ก็

เพียงพอ สำหรับนิตยสารที่พิมพ์บนกระดาษผิวมันก็อาจจะใช้ความถี่เม็ดสกรีนที่มากขึ้นสัก 133-150 เส้น เพื่อแสดงรายละเอียดให้มากยิ่งขึ้น และสำหรับงานพิมพ์ที่เน้นคุณภาพมากๆ ก็อาจจะใช้ความถี่เม็ดสกรีนที่สูงขึ้นไปถึงระดับ 180 -200 เส้น

2.1.3.1.2 ความละเอียดของเครื่องอิมเมจเซ็ตเตอร์และขนาดของเม็ดสกรีน

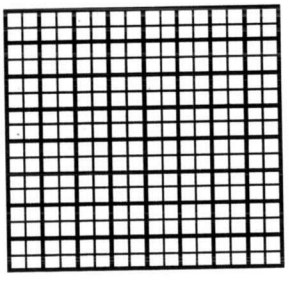
ขนาดของฮาล์ฟโทนเซลล์จะถูกกำหนดขึ้นทันทีว่ามีขนาดเท่าไร เมื่อเราเลือกสร้างเม็ดสกรีนที่ความถี่เม็ดสกรีนค่าหนึ่งบนเครื่องอิมเมจเซ็ตเตอร์ตัวหนึ่ง ๆ ซึ่งมีความละเอียด (resolution) ระดับหนึ่ง ๆ ขนาดของฮาล์ฟโทนเซลล์จะขึ้นอยู่กับความสัมพันธ์ของความถี่เม็ดสกรีนและความละเอียดของเครื่องอิมเมจเซ็ตเตอร์ กล่าวคือ ความละเอียดของเครื่องอิมเมจเซ็ตเตอร์จะเป็นตัวกำหนดว่าตัวเครื่องสามารถสร้างจุดภาพเล็ก ๆ หรือเม็ดสกรีนได้มากเพียงใดในความยาว 1 นิ้ว จุดภาพเหล่านี้จะเกิดจากการยิงแสงเลเซอร์ไปตกกระทบบนผิววัสดุหรือแผ่นฟิล์ม เมื่อแผ่นภาพหรือฟิล์มได้รับแสงเลเซอร์ก็จะเกิดเป็นภาพสีดำหรือจุดดำ ยิ่งจุดดำอยู่ใกล้ ๆ กันมากเท่าไรก็ยิ่งแสดงถึงความละเอียดของเครื่องอิมเมจเซ็ตเตอร์ที่มากขึ้นเท่านั้น

ความละเอียดของเครื่องอิมเมจเซ็ตเตอร์จะหมายถึงปริมาณจุดภาพเล็ก ๆ ที่สามารถสร้างขึ้นมาได้ในความยาว 1 นิ้ว ซึ่งจะมีหน่วยวัดเป็นจุดต่อนิ้ว (dpi) และเรียกจุดเล็ก ๆ เหล่านั้นว่าฮาล์ฟโทนด็อต



ภาพที่ 2-22 ภาพแสดงฮาล์ฟโทนด็อต

เมื่อเราวางฮาล์ฟโทนเซลล์ลงไปประกบกับความละเอียดทั้งหมดของอิมเมจเซ็ตเตอร์ ก็จะเป็นการแบ่งความละเอียดทั้งหมดให้เป็นกลุ่ม ๆ แต่ละกลุ่มจะประกอบไปด้วยจุดเล็ก ๆ จำนวนเท่ากับฮาล์ฟโทนเซลล์



ภาพที่ 2-23 ฮาล์ฟโทนเซลล์วางเทียบลงไปบนจุดภาพความละเอียดทั้งหมด

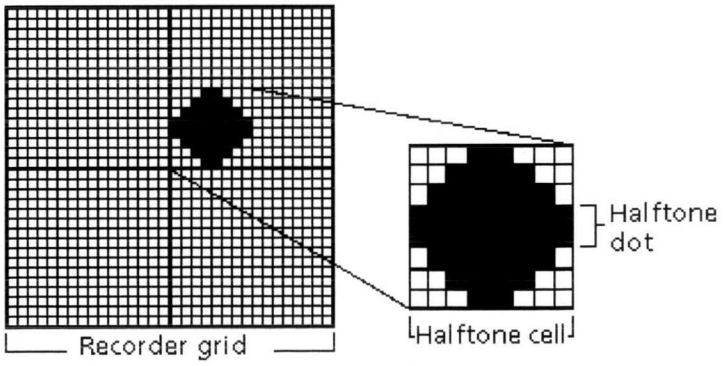
เราสามารถคำนวณหาค่าจำนวนจุดภาพในฮาล์ฟโทนเซลล์ได้ด้วยสมการต่อไปนี้

จำนวนจุดภาพต่อหนึ่งหน่วยฮาล์ฟโทนเซลล์ = (ความละเอียดของอิมเมจเซ็ตเตอร์ / ความถี่เม็ดสกรีน) ^{2} **สมการที่** 2-4

จากรูปตัวอย่าง เราจะคำนวณตัวเลขต่างๆได้ดังนี้ ภาพใหญ่ทั้งภาพแทนความละเอียดสูงสุดของเครื่องอิมเมจเซ็ตเตอร์ จะเห็นได้ว่ามีชุดภาพในแนวนอนอยู่ทั้งหมด 16 จุด นั่นหมายถึงความละเอียดของอิมเมจเซ็ตเตอร์ในภาพนี้มีค่า 16 dpi ส่วนความถี่เม็ดสกรีนก็นับได้จากจำนวนฮาล์ฟโทนเซลล์ที่เรียงตัวต่อเนื่องตลอดความละเอียดของอิมเมจเซ็ตเตอร์ นั่นก็คือ 8 ฮาล์ฟโทนเซล หรือความถี่เท่ากับ 8 lpi ดังนั้นจะคำนวณหาจำนวนจุดภาพในหนึ่งฮาล์ฟโทนเซลลได้เท่ากับ $(16/8)^2 = 4$ ในทางปฏิบัติ จำนวนจุดภาพต่อหนึ่งฮาล์ฟโทนเซลล์จะมีค่าอยู่ในระดับหลักร้อยต่อภาพ เช่น ถ้าเครื่องอิมเมจเซ็ตเตอร์มีความละเอียด 2400dpi และเลือกใช้ความถี่เม็ดสกรีนที่ 100 lpi จำนวนจุดภาพต่อหนึ่งหน่วยฮาล์ฟโทนเซลล์จะเท่ากับ $(2400/100)^2 = 576$

2.1.3.1.3 ขนาดและรูปร่างของเม็ดสกรีน

จุดภาพเล็ก ๆ ในฮาล์ฟโทนเซลล์หนึ่งๆจะสามารถแสดงผลเป็นจุดสีดำหรือเป็นจุดสีขาวก็ได้ การแสดงผลเป็นจุดสีดำจะทำให้เกิดเป็นเม็ดสกรีนขึ้น ซึ่งหากมีจุดภาพสีดำอยู่เป็นจำนวนมากเม็ดสกรีนก็จะใหญ่ ถ้ามีจุดภาพสีดำอยู่น้อยเม็ดสกรีนก็เล็ก ถ้าไม่มีจุดที่เป็นสีดำเลยก็หมายถึงไม่มีเม็ดสกรีนหรือไม่มีภาพนั่นเอง นอกจากปริมาณจุดดำแล้ว การเกิดจุดดำที่อยู่ใกล้ ๆ กันก่อให้เกิดเป็นรูปร่างต่าง ๆ ซึ่งจะเรียกว่ารูปร่างเม็ดสกรีน ซึ่งอาจจะเป็นชนิดรูปวงกลม รูปวงรี รูปเพชร หรืออาจจะเป็นรูปเส้นตรงก็ได้



ภาพที่ 2-24 ภาพแสดงฮาล์ฟโทนเซลล์ และฮาล์ฟโทนด๊อต

ถ้าต้องการให้เม็ดสกรีนมีขนาดที่ใหญ่ขึ้น ก็จะต้องสั่งการให้เครื่องอิมเมจเซ็ตเตอร์สร้างจุดสีดำในฮาล์ฟโทนเซลล์ให้มากขึ้น ถ้าต้องการเม็ดสกรีนขนาดเล็ก ก็ให้อิมเมจเซ็ตเตอร์สร้างจุดสีดำในฮาล์ฟโทนเซลล์ให้น้อยลง หรือหากต้องการเม็ดสกรีนที่มีรูปทรงในแบบที่ต้องการ ก็ต้องสั่งให้เครื่องอิมเมจเซ็ตเตอร์ทำการสร้างจุดดำด้วยรูปแบบที่แตกต่างกันไป ซึ่งรูปแบบของเม็ดสกรีนต่าง ๆ จะเป็นการกำหนดด้วยสมการทางคณิตศาสตร์ สมการนี้จะเรียกว่าเป็น spot function เมื่อเปลี่ยนสมการการสร้างภาพฮาล์ฟโทนก็มีผลทำให้รูปแบบของเม็ดสกรีนเปลี่ยนไป

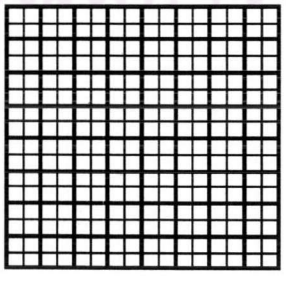
2.1.3.1.4 ขนาดและรูปร่างของเม็ดสกรีนระดับสีเทา

ในมาตรฐานเครื่องพิมพ์แบบโพสสคริปต์ต้องการข้อมูลดิจิทัลที่มีค่าระดับสีเทา 256 ระดับเพื่อสร้างภาพ ดังนั้นเครื่องอิมเมจเซ็ตเตอร์ต่าง ๆ จึงพัฒนาให้รองรับระดับสีเทาได้ 256 ระดับเช่นเดียวกัน

ในฮาล์ฟโทนเซลล์ที่มีขนาดจุดภาพเป็นปริมาณมาก ๆ ยิ่งจะทำให้เม็ดสกรีนสามารถแสดงระดับสีเทาได้มากระดับยิ่งขึ้น ซึ่งจะทำให้ภาพพิมพ์สามารถถ่ายทอดออกมาได้ใกล้เคียงกับต้นฉบับมากยิ่งขึ้น การคำนวณหาระดับสีเทาที่สามารถให้ได้จะหาได้จากสมการ

$$\#\ \text{gray levels} = (\text{dpi} / \text{lpi})^2 + 1 \qquad \textbf{สมการ 2-5}$$

ในภาพตัวอย่างแสดงให้เห็นระดับความละเอียดของเครื่องอิมเมจเซ็ตเตอร์ 16x16 จุด ขนาดฮาล์ฟโทนเซลล์ 2x2 จุด



ภาพที่ 2-25 ภาพแสดงความละเอียดของอิมเมจเซ็ตเตอร์ และฮาล์ฟโทนเซลล์ 2x2

ระดับสีเทาที่ตัวอย่างในรูปนี้สามารถให้ได้จะหาได้จากสมการ $(16 / 8)^2 + 1 = 5$ และในแต่ละฮาล์ฟโทนเซลล์จะมีระดับสีเทาดังนี้

ก. ระดับสีเทา 0 % คือ ไม่มีจุดภาพสีดำเลย หมายความว่าภาพเป็นสีขาว

ข. ระดับสีเทา 25% คือ มีจุดดำ 1 จุด จาก 4 จุดในฮาล์ฟโทนเซลล์

ค. ระดับสีเทา 50 % คือ มีจุดดำ 2 จุด จาก 4 จุดในฮาล์ฟโทนเซลล์

ง. ระดับสีเทา 75 % คือ มีจุดดำ 3 จุด จาก 4 จุดในฮาล์ฟโทนเซลล์

จ. ระดับสีเทา 100 % คือ มีจุดดำ 4 จุด จาก 4 จุดในฮาล์ฟโทนเซลล์ หมายความว่าภาพเป็นสีดำ

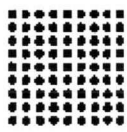
ก ข ค ง จ

ภาพที่ 2-26 ระดับสีเทาที่เกิดขึ้นทั้งหมดในฮาล์ฟโทนเซลล์ขนาด 2x2 ช่อง

การเลือกทำงานให้มีความถี่เม็ดสกรีนสูงๆเพื่อให้สามารถถ่ายทอดรายละเอียดได้คมชัดใกล้เคียงต้นฉบับ ต้องแลกมาด้วยระดับสีเทาที่น้อยระดับ เนื่องจากความละเอียดของเครื่องอิมเมจเซ็ตเตอร์เป็นสิ่งที่จำกัด การกำหนดให้ความถี่เม็ดสกรีนสูงๆจะเป็นการแบ่งความละเอียดทั้งหมดให้กับฮาล์ฟโทนเซลล์จำนวนมาก ผลทำให้จุดภาพขนาดเล็กที่สุดในแต่ละฮาล์ฟโทนเซลล์มีปริมาณน้อยลง ซึ่งเป็นไปตามสมการ 2-5 ผลก็คือระดับสีเทาจะน้อยลง

2.1.3.1.5 รูปแบบเม็ดสกรีน

เมื่อแบ่งภาพพิมพ์ออกเป็นส่วนเล็กๆ จะเห็นว่าเกิดจากการรวมตัวกันของเม็ดสกรีนจนเกิดเป็นภาพ เม็ดสกรีนที่จัดวางตัวในแนวดิ่งหรือแนวนอน หรือทำมุมในระดับ 0 องศา หรือ 90 องศา จะมีผลทำให้เกิดภาพลวงตาขึ้น เป็นภาพที่มีลักษณะเป็นตาราง หรือตาเสือ



ภาพที่ 2-27 มุมสกรีน 90 องศา

วิธีที่จะหลีกเลี่ยงการเกิดตาเสื่อก็คือการมุนเม็ดสกรีนเหล่านี้ให้ทำมุมกัน 45 องศา ซึ่งจะทำให้ลดอาการภาพตาเสื่อลงได้ โดยสายตาจะไม่สังเกตเห็นผลของตาเสื่อนั่นเอง และในมุมสกรีนที่ 45 องศาจะทำให้ภาพที่เห็นดูเหมือนเป็นภาพมากกว่าจะเป็นกลุ่มเม็ดสกรีน

ภาพที่ 2-28 **มุมสกรีน** 45 **องศา**

2.1.3.1.6 การแยกสี

ในขั้นตอนการแยกสี งานพิมพ์จะถูกแบ่งออกเป็น 4 สี ซึ่งเป็นแม่สีสำหรับงานพิมพ์สอดสี แต่ละสีจะต้องทำให้เป็นภาพฮาล์ฟโทน นั่นคือจะทำให้เกิดเป็นเม็ดสกรีน และเมื่อนำภาพฮาล์ฟโทนของแต่ละสีมาพิมพ์ซ้อนทับกัน การรวมตัวของเม็ดสกรีนของแต่ละสีจะทำให้เกิดภาพตาเสื่อขึ้นมาด้วย แต่ก็สามารถแก้ไขได้ด้วยการหมุนมุมสกรีนให้แต่ละสีมีค่าแตกต่างกันดังต่อไปนี้

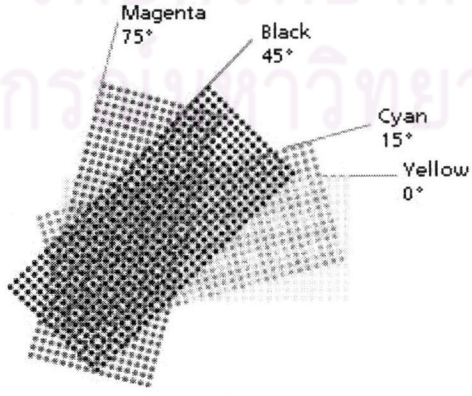
ก. สีดำ ใช้มุมสกรีน 45 องศา

ข. สีชมพู ใช้มุมสกรีน 75 องศา

ค. สีฟ้า ใช้มุมสกรีน 15 หรือ 105 องศา

ง. สีเหลือง ใช้มุมสกรีน 0 หรือ 90 องศา

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



ภาพที่ 2-29 ภาพมุมสกรีนสำหรับภาพฮาล์ฟโทนทั้ง 4 สี

2.2 วารสารปริทรรศน์

2.2.1 การบีบอัดแบบเจเป็ก (JPEG Compression)

การบีบอัดข้อมูลที่แพร่หลายที่สุดในปัจจุบัน(พ.ศ. 2548) คือการบีบอัดข้อมูลแบบ JPEG ซึ่งเป็นเทคโนโลยีที่ได้รับการคิดค้นและพัฒนามาอย่างต่อเนื่อง คำว่า JPEG เป็นคำย่อมาจากชื่อเต็ม Joint Photographic Expert Group ซึ่งเป็นกลุ่มความร่วมมือกลุ่มหนึ่งที่จัดตั้งขึ้นโดย ISO หรือ International Standard Organization ในปี ค.ศ. 1982 [6]

ในครั้งแรก การรวมตัวของกลุ่มนักวิจัยเพื่อพัฒนาการบีบอัดภาพ จะรวมกลุ่มกันในชื่อ PEG หรือ Photographic Expert Group ซึ่งมีวัตถุประสงค์เพื่อพัฒนาการบีบอัดข้อมูลภาพถ่าย ภาพเคลื่อนไหว ภาพนิ่งต่างๆ และข้อมูลตัวหนังสือ เพื่อให้สามารถส่งผ่านระบบเครือข่ายได้ ซึ่งระบบเครือข่ายในเวลานั้นคือ ISDN เป้าหมายเบื้องต้นคือการกำหนดมาตรฐานการสื่อสารข้อมูลขึ้นเพื่อให้วงการสื่อสารทั้งระบบสามารถติดต่อแลกเปลี่ยนข้อมูลภาพซึ่งกันและกันได้โดยมีมาตรฐานเดียว

ในขณะที่ PEG กำลังพัฒนาการบีบอัดข้อมูลตัวใหม่อยู่นี้ ก็มีกลุ่มความร่วมมือกันอีกกลุ่มหนึ่งได้พยายามพัฒนาการบีบอัดภาพสีและขาวดำเพื่อที่จะส่งผ่านระบบโทรสาร กลุ่มวิจัยกลุ่มใหม่นี้คือ CCITT ซึ่งได้ริเริ่มพัฒนากันในปี ค.ศ. 1986 โดยแนวทางการพัฒนาของ PEG และ CCITT ทั้งสองกลุ่มนี้จะมีความคล้ายคลึงกันอย่างมาก หลายฝ่ายจึงมีความเห็นว่าทั้งสองกลุ่มควรจะร่วมมือกันพัฒนา แบ่งปันความรู้และทรัพยากรการทำงานต่าง ๆ ร่วมกัน เพื่อให้การวิจัยและพัฒนาบรรลุวัตถุประสงค์เร็วขึ้น และในที่สุดในปี ค.ศ. 1987 ทาง ISO และ CCITT ก็ได้รวมกลุ่มการวิจัยของทั้งคู่เข้าด้วยกัน จึงเป็นที่มาของคำว่า JPEG ซึ่งตัวย่อ J ตัวแรกมาจากคำว่า JOINT

การบีบอัดข้อมูลภาพแบบ JPEG ได้รับการต้อนรับเป็นอย่างดีจากผู้ผลิตฮาร์ดแวร์และซอฟต์แวร์ เนื่องจากในอดีตก่อนที่จะมี JPEG เทคโนโลยีการบีบอัดข้อมูลที่มีอยู่ในขณะนั้นไม่สามารถทำงานได้ดีกับภาพถ่าย ประสิทธิภาพการบีบอัดข้อมูลที่ทำได้ยังไม่สูงเพียงพอ เป็นเพราะภาพถ่ายเป็นภาพที่มีการไล่ระดับสีที่ต่อเนื่อง ซึ่งในแต่ละภาพจะมีองค์ประกอบเป็นจำนวนสีนับร้อยนับพันสี และที่สำคัญก็คือ มาตรฐานไฟล์ข้อมูลภาพในขณะนั้นมีไม่กี่ตัวเท่านั้นที่สามารถรองรับข้อมูลภาพได้ถึง 24 บิต

อย่างเช่นภาพ GIF เป็นมาตรฐานหนึ่งของการเก็บข้อมูลภาพ พื้นฐานของ GIF จะใช้จำนวนบิตข้อมูลเพียง 8 บิตต่อ 1 จุดข้อมูล หมายความว่าภาพ GIF จะถ่ายทอดสีได้เพียง 256 สีเท่านั้น ส่วนการบีบอัดแบบ LZW ก็ไม่เหมาะกับภาพถ่าย เนื่องจากความซ้ำกันของข้อมูลในภาพ

จะมีเพียงเล็กน้อยเท่านั้น การบีบอัดภาพถ่ายด้วย LZW จึงยังมีประสิทธิภาพการบีบอัดที่ไม่ดีเพียงพอ

ไฟล์ภาพชนิด TIFF และ BMP เป็นไฟล์ภาพที่สามารถเก็บข้อมูลภาพสีได้ 24 บิต และสามารถใช้เก็บข้อมูลภาพถ่ายได้ แต่การบีบอัดข้อมูลที่สามารถใช้กับ BMP และ TIFF ได้ก็มีเพียง LZW และ RLE เท่านั้น ไฟล์ภาพชนิด TIFF จึงไม่ได้รับความนิยมในการใช้งานเหมือนดังเช่นที่ RLE และ LZW เคยประสบมา

การบีบอัดข้อมูลภาพโดย JPEG มีประสิทธิภาพการทำงานที่สูง เป็นทางเลือกที่ดีสำหรับการใช้งาน ตัว JPEG สามารถทำงานกับไฟล์ภาพได้ตั้งแต่การใช้ข้อมูลนาด 6 บิตต่อ 1 จุดภาพไปจนถึง 24 บิตต่อจุดภาพ หรือที่นิยมเรียกกันว่ามีความลึกของบิต (bit depth) เท่ากับ 6 ไปจนถึง 24 บิตสี [7] การกำหนดขนาดไฟล์ข้อมูลที่บีบอัดแล้วจึงเป็นสิ่งที่ทำได้และอำนวยความสะดวกต่อการใช้งาน ถ้าอยากให้ประสิทธิภาพการบีบอัดสูง ก็บีบอัดด้วยคุณภาพในระดับต่ำ ถ้าต้องการคุณภาพสูงเป็นสำคัญ ก็เลือกใช้ประสิทธิภาพการบีบอัดที่น้อยลง แต่ขนาดไฟล์ก็จะมากขึ้น

เมื่อการบีบอัดแบบ JPEG สามารถเลือกได้ว่าต้องการคุณภาพระดับต่ำหรือสูงได้ตามต้องการ นั่นก็หมายความว่าการบีบอัดแบบ JPEG เป็นการบีบอัดที่มีการสูญเสีย โดยไฟล์ JPEG ที่บีบอัดมากๆจะมีการทิ้งข้อมูลภาพบางส่วนออกไป ทำให้ขนาดการบีบอัดเหลือเล็กลง ข้อมูลภาพที่ทิ้งไปนี้มักจะเป็นรายละเอียดที่ไม่สามารถมองเห็นได้ง่าย ๆ หรือเป็นข้อมูลที่ไม่มีความสำคัญต่อภาพ การบีบอัดภาพที่มีการละทิ้งข้อมูลบางส่วนจะทำให้ไม่สามารถคืนรูปข้อมูลกลับไปให้เหมือนต้นฉบับได้ ซึ่งการบีบอัดแบบมีการทิ้งข้อมูลนี้จะเรียกว่าการบีบอัดแบบมีการสูญเสีย (Lossy Compression) [5]

การบีบอัดแบบ JPEG ไม่ได้เป็นการบีบอัดที่ดีที่สุดสำหรับภาพถ่ายทั่วไปแม้ว่าจะมีข้อดีในด้านประสิทธิภาพการบีบอัดที่สูงมาก เนื่องจากผลจากการบีบอัดจะเกิดความเพี้ยนที่สามารถมองเห็นได้ชัดเจน ยิ่งการบีบอัดที่มีขนาดภาพที่เล็กมากเท่าใด ความเพี้ยนที่เกิดขึ้นก็มีมากยิ่งขึ้นเท่านั้น การบีบอัดแบบ JPEG จึงยังไม่ใช่การบีบอัดที่เป็นอุดมคติสำหรับข้อมูลภาพ ความเพี้ยนที่เกิดขึ้นจะมีชื่อเรียกว่า Artifact ซึ่งบางครั้งมีมากจนกลายเป็นจุดด้อยของการบีบอัด JPEG

การใช้งานไฟล์ภาพ JPEG ที่แพร่หลายมากยิ่งขึ้นเรื่อยๆ ทำให้เกิดการพัฒนา JPEG จากการบีบอัดแบบ Lossy ให้เป็น Lossless ก็คือทำให้ JPEG สามารถบีบอัดภาพในแบบไม่สูญเสียได้ เพื่อขจัดความเพี้ยนหรือ artifact จากการบีบข้อมูล การพัฒนาดังกล่าวเป็นการพัฒนาไปสู่มาตรฐาน JPEG2000 โดยการเปลี่ยนขั้นตอนภายในที่สำคัญขั้นตอนหนึ่งคือ เปลี่ยนจากการใช้ Discrete Cosine Transform มาเป็น Wavelet Transform การพัฒนาในครั้งนี้ประกาศใช้เป็นมาตรฐานในปีค.ศ. 2000 แต่ในปัจจุบันยังไม่มีการใช้งานที่แพร่หลายเนื่องจากยังใหม่ต่อผู้ใช้ทั่วไป

อยู่มาก แต่ในความเห็นของผู้เขียน เชื่อว่าการใช้งาน JPEG2000 จะยังไม่ได้รับความนิยมไปจนกว่าจะถึงปี ค.ศ. 2010 ซึ่งเป็นปีที่ JPEG2000 มีอายุครบ 10 ปี การคุ้มครองด้านสิทธิบัตรจะสิ้นสุดลง ทำให้ผู้ผลิตฮาร์ดแวร์และซอฟต์แวร์สามารถใช้งานมาตรฐาน JPEG2000 ได้โดยไม่ต้องเสียค่าลิขสิทธิ์ ดังนั้นในช่วงระยะเวลาระหว่างนี้(ค.ศ.2005) จะยังไม่มีการใช้งานไฟล์ JPEG2000 อย่างแน่นอน

2.2.2 การบีบอัดแบบพรีฟิกส์ (Prefix Compression)

ภาพที่มีลักษณะเป็นภาพขาวดำ มีพื้นที่สีดำเพียงเล็กน้อยเมื่อเทียบกับสีขาว หากใช้การบีบอัดแบบ JPEG ก็จะไม่สามารถลดขนาดลงได้อย่างมีประสิทธิภาพ จึงได้มีแนวคิดที่จะเก็บข้อมูลภาพลักษณะนี้ด้วยการเก็บบันทึกเฉพาะข้อมูลส่วนที่เป็นสีดำ และไม่บันทึกข้อมูลส่วนที่เป็นสีขาว ภาพที่มีข้อมูลสีดำบางส่วนและมีสีขาวเป็นส่วนใหญ่มักจะเป็นภาพเอกสาร ข้อมูลตัวหนังสือในหน้าเอกสารมักจะมีน้อยกว่าบริเวณที่เป็นพื้นสีขาวอยู่หลายเท่า อัลกอริธึมการบีบอัดที่พัฒนาขึ้นมาเพื่อเก็บข้อมูลเฉพาะสีดำนี้มีชื่อเรียกว่า การบีบอัดแบบพรีฟิกส์

การบีบอัดโดยอาศัยหลักการของการบีบอัดแบบพรีฟิกส์ ได้รับการพัฒนาขึ้นในปี 1985 [8] ก่อนที่จะมีการบีบอัดแบบ JPEG หลักการของการบีบอัดแบบพรีฟิกส์ จะเหมาะสมกับภาพขาวดำที่มีระดับสีเพียงสองระดับ นั่นก็คือภาพบิตแม็บ แบบ 1 บิต เมื่อพิจารณาแล้วการบีบอัดด้วยวิธีของพรีฟิกส์ นี้น่าจะเหมาะสมกับเอกสารแฟกซ์ แต่ในหลายปีต่อมาเครื่องโทรสารหรือแฟกซ์ก็ไม่ได้นำอัลกอริธึมนี้มาใช้งานเนื่องจากได้มีการค้นพบวิธีการบีบอัดข้อมูลภาพขาวดำที่ให้ผลการทำงานได้ดีกว่า นั่นคือการบีบอัดแบบ JBIG

แม้ว่าผลการทำงานของอัลกอริธึมพรีฟิกส์ นี้ทำให้เหลือข้อมูลที่เก็บบันทึกน้อยลง ภาพที่เหมาะกับการบีบอัดนี้จะต้องเป็นภาพที่มีปริมาณสีดำน้อยกว่าสีขาวมาก ๆ ถ้าหากพื้นที่สีดำมีมากพอ ๆ กับสีขาวอย่างเช่นภาพฮาล์ฟโทนการบีบอัดข้อมูลด้วยอัลกอริธึมพรีฟิกส์นี้จะไม่สามารถทำได้ เนื่องจากผลการบีบอัดจะได้ข้อมูลภาพที่มีขนาดใหญ่ขึ้นกว่าต้นฉบับก็เป็นได้ นั่นจึงเป็นสาเหตุที่ยังไม่มีการนำการบีบอัดแบบพรีฟิกส์ นี้มาใช้งานจริง และตั้งแต่ปี 1985 เป็นต้นมาก็ยังไม่พบการพัฒนาของอัลกอริธึมนี้อีกเลย

หลังจาก Prefix Compression ที่มีหลักการแบ่งพื้นที่ข้อมูลออกเป็น 4 ส่วนย่อย และแบ่งแต่ละส่วนออกเป็นสี่ส่วนย่อย ๆ ไปเรื่อย ๆ ก็มีอีกอัลกอริธึมหนึ่งที่มีหลักการคล้ายกัน นั่นก็คือการบีบอัดแบบ Quadtree [9] แต่ในที่สุดก็ยังไม่มีการใช้งานที่เป็นรูปธรรมในปัจจุบัน

2.2.3 การบีบอัดแบบ LZ (LZ Compression)

การบีบอัดข้อมูลที่ได้รับความนิยมในการนำมาใช้งานกับภาพมากที่สุดคือการบีบอัดที่มีชื่อว่า LZW หรือ Lempel-Ziv-Welch ซึ่งเป็นการบีบอัดที่ไม่มีการสูญเสีย (lossless) สามารถคืนรูปข้อมูลต้นฉบับกลับมาได้ครบถ้วนไม่ตกหล่น ไม่มีความเพี้ยนใดๆ การบีบอัดด้วยวิธีของ LZW ได้ถูกนำไปใช้ในไฟล์ข้อมูลภาพชนิด GIF และ Tiff และนอกจากไฟล์ภาพแล้ว ยังมีการใช้งาน LZW อยู่ในบางส่วนของการสื่อสารภายในโมเด็มอีกด้วย

ในปีค.ศ. 1977 Abraham Lempel และ Jakob Ziv ได้ร่วมกันพัฒนาอัลกอริธึมการบีบอัดข้อมูลขึ้นมาชนิดหนึ่ง นั่นก็คือการบีบอัดแบบ LZ และได้กำหนดเป็นมาตรฐานการบีบอัดสำหรับข้อมูลชนิดตัวหนังสือ มาตรฐานนี้ชื่อว่า LZ77 [10] ประสิทธิภาพการบีบอัดข้อมูลตัวหนังสือของ LZ77 นี้มีค่าสูงมาก ทำให้ได้รับความนิยมนำไปพัฒนาใช้งานเป็นโปรแกรมอย่าง Zoo , Lha , PkZIP และ arj หลักจากนั้น LZ77 ก็ได้รับการพัฒนาต่อจนกลายเป็น LZ78 [11] เพื่อให้สามารถใช้งานได้ดีกับข้อมูลชนิดไบนารีและภาพแบบบิตแม็บได้ด้วย

ในปีค.ศ.1984 Terry Welch ได้นำ LZ78 มาพัฒนาต่อเพื่อให้สามารถจัดการกับการบีบอัดข้อมูลได้มากยิ่งขึ้นเพื่อนำไปใช้งานกับการควบคุมฮาร์ดดิสก์ ผลการพัฒนาของ Welch ทำให้เกิดเป็น LZW ในที่สุด [12] และความสามารถของ LZW ที่ได้ก็เป็นความยอดเยี่ยมครั้งหนึ่งของวงการคอมพิวเตอร์ นั่นเป็นเพราะการบีบอัดแบบ LZW เป็นวิธีการบีบอัดที่สามารถใช้งานได้กับข้อมูลที่หลากหลาย ไม่ว่าจะเป็นตัวหนังสือ หรือภาพถ่าย นอกจากประสิทธิภาพการบีบอัดที่สูงแล้วยังทำงานได้รวดเร็วมาก ทั้งการบีบอัดข้อมูลและการคืนรูปข้อมูล เนื่องจากการบีบอัดของ LZW ไม่มีการคำนวณที่ซับซ้อน ทำให้ไม่มีความจำเป็นต้องใช้หน่วยประมวลผลที่มีความสามารถในการคำนวณจุดทศนิยมที่สูงเป็นพิเศษ (floating point unit) LZW จะจัดการกับข้อมูลในระดับ byte ซึ่งเป็นโครงสร้างข้อมูลที่คอมพิวเตอร์สามารถเข้าถึงได้เร็วที่สุดนั่นเอง

การนำ LZW ไปใช้งานนอกเหนือไปจากการบีบข้อมูลไฟล์เป็นชิ้นๆแล้ว ยังมีการนำ LZW ไปเป็นส่วนหนึ่งของระบบการเก็บข้อมูลของดิกชั่นนารี การทำงานของดิกชั่นนารีจะบันทึกข้อมูลตัวหนังสือส่วนที่ซ้ำ ๆ กันด้วยข้อมูลชุดเดียว ทำให้สามารถลดปริมาณตัวหนังสือในดิกชั่นนารีลงไปได้อย่างมหาศาลเนื่องจากไม่ต้องเก็บบันทึกคำทั้งคำ อาศัยเก็บตัวหนังสือเป็นชุดๆที่ซ้ำกับคำที่เก็บบันทึกไปแล้วและเก็บข้อมูลใหม่จากตัวหนังสือที่ยังไม่ซ้ำเท่านั้น

ส่วนการนำ LZW ไปใช้กับภาพ Tiff ก็จะอาศัยหลักการเก็บข้อมูลภาพ byte เอาไว้และข้อมูลตัวใหม่ที่ซ้ำกับข้อมูลที่เคยเก็บไปแล้วก็จะอาศัยเพียงอ้างอิงตำแหน่งเท่านั้น ข้อมูลที่เก็บจริงก็จะมีจำนวนน้อยลง นอกจากไฟล์ภาพแบบ Tiff แล้ว ภาพแบบ Gif ก็อาศัยการบีบอัดแบบ

LZW ได้เช่นเดียวกัน [13] พื้นฐานของ Gif จะมีค่าสีเพียง 256 ระดับ ซึ่งก็ใช้จำนวนบิตแทนค่าสีเพียง 8 บิตหรือ 1 byte เท่านั้น ซึ่งก็อยู่ในขอบเขตที่ LZW สามารถทำงานได้โดยไม่มีปัญหา

Terry Welch ได้ตีพิมพ์ความสำเร็จของการบีบอัดข้อมูลด้วยวิธีการ LZW ในวารสารของ IEEE เป็นครั้งแรกและหลังจากนั้นในปี 1985 การบีบอัดแบบ LZW ก็ได้รับการจดสิทธิบัตรโดยกรรมสิทธิ์เป็นของบริษัท Sperry Corporation ซึ่งเป็นบริษัทที่ Welch ทำงานอยู่ ต่อมาในปี 1986 ก็ได้มีการควบรวมกิจการ ทำให้เจ้าของสิทธิบัตรเปลี่ยนเป็น Unisys [14]

การถือกำเนิดของไฟล์ชนิด Gif ซึ่งเป็นผลผลิตของบริษัท CompuServe นั่นได้อาศัยการทำงานของ LZW ในการบันทึกข้อมูล การพัฒนา GIF โดยใช้ LZW เป็นส่วนประกอบนั้นเป็นความผิดพลาดครั้งหนึ่งของ CompuServe ที่ไม่ตรวจสอบความเป็นเจ้าของว่าผู้ใดเป็นผู้ถือลิขสิทธิ์ของ LZW เมื่อไฟล์ชนิด GIF ได้รับความแพร่หลายจากการโปรโมทอย่างเต็มที่ การใช้งานไฟล์ GIF ในขณะนั้นจะต้องมีการจ่ายเงินให้กับ CompuServe ซึ่งเงินบางส่วนจะต้องแบ่งให้เจ้าของ LZW อีกต่อหนึ่ง [14]

บริษัท Aldus ได้พัฒนาไฟล์ภาพชนิด Tiff เวอร์ชั่นที่ 5 และได้ผลิตโปรแกรมจัดหน้าเพื่อเตรียมสิ่งพิมพ์ การทำงานของโปรแกรมได้เปิดโอกาสให้ผู้ใช้งานได้บันทึกข้อมูลภาพเป็นไฟล์ชนิด Tiff เวอร์ชั่น 5 ซึ่งมีตัวเลือกว่าจะบันทึกแบบเป็นข้อมูลดิบไม่มีการบีบอัดเลย หรือจะเลือกใช้การบีบอัดก็ได้ ซึ่ง Aldus เลือกการบีบอัด LZW มาใช้ใน Tiff ของตน ดังนั้น Aldus จึงต้องจ่ายเงินให้กับ Unisys อีกครั้ง และหลังจากนั้นเป็นต้นมาบริษัทใดก็ตามที่มีส่วนเกี่ยวข้องที่จะต้องใช้งานไฟล์ชนิด Tiff ก็มีอันต้องจ่ายค่าลิขสิทธิ์ให้กับ Unisys ทั้งสิ้น หากนับไปแล้วจะมีบริษัทมากกว่า 100 แห่งตั้งแต่ปี ค.ศ. 1990 เป็นต้นมาที่จ่ายเงินค่าลิขสิทธิ์ให้กับ Unisys [14]

ความนิยมในการใช้งานการบีบอัดข้อมูลภาพแบบ LZW บนไฟล์ภาพชนิด Tiff เกิดขึ้นอย่างต่อเนื่อง แต่ LZW ก็ยังไม่สามารถทำงานกับภาพบางลักษณะได้ดีเพียงพอ จึงเกิดการพัฒนาไฟล์ภาพมาตรฐานอื่น ๆ ขึ้นมาอีก ไม่ว่าจะเป็น JPEG , JBIG และ JPEG2000 ที่จะเป็นคู่แข่งของ Tiff ต่อไป

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย