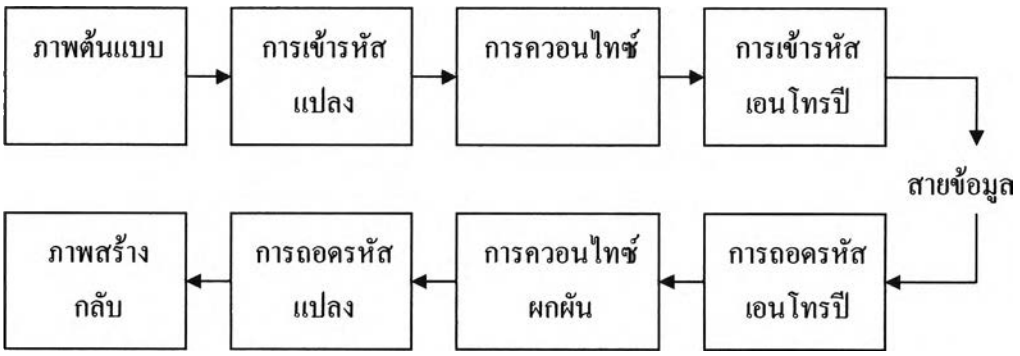
บทที่ 2

ความรู้พื้นฐานและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะอธิบายขั้นตอนการประมวลสัญญาณภาพ ทฤษฎีที่เกี่ยวข้อง และหลักการของมาตรฐาน JPEG2000 ให้ผู้อ่านเข้าใจถึงแต่ละขั้นตอนการทำงาน

2.1 การประมวลผลสัญญาณภาพ

ข้อมูลภาพจะถูกแทนด้วยแถวลำดับ (Array) 2 มิติ สมาชิกในแถวลำดับเรียกว่า จุดภาพ (Pixel) ค่าของจุดภาพใช้แทนความเข้มหรือความสว่างของภาพที่ตำแหน่งนั้น ใช้ตัวเลขที่มีค่าต่ำแทนความสว่างน้อยและตัวเลขที่มีค่าสูงแทนความสว่างมาก ตัวเลขที่ซึ่งใช้แทนค่าความสว่างของจุดภาพนี้เรียกได้ว่าเป็นข้อมูลดิบ (Raw Data) ซึ่งมีขนาดใหญ่ไม่เหมาะสมกับการจัดเก็บข้อมูล เพื่อการจัดเก็บข้อมูลอย่างประหยัดหน่วยความจำก็มีขั้นตอนที่เรียกว่าการบีบอัดข้อมูล และข้อมูลที่ถูกบีบอัดแล้วเมื่อต้องการนำกลับมาใช้ก็จะต้องผ่านขั้นตอนที่เรียกการขยายข้อมูล



รูปที่ 2.1 บล็อกไดอะแกรมของการบีบอัด และการขยายข้อมูลภาพ

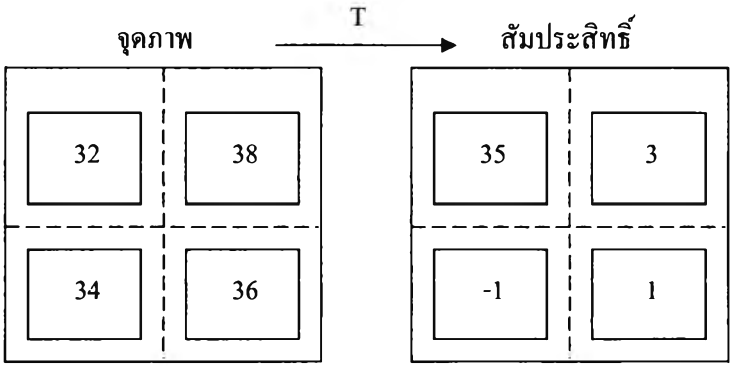
ขั้นตอนในการบีบอัดภาพนิ่งโดยทั่วไปแล้วแบ่งได้เป็น 3 ขั้นตอน ดังรูปที่ 2.1 ได้แก่ การเข้ารหัสแปลง (Transform coding) เพื่อให้การบีบอัดข้อมูลทำได้อย่างมีประสิทธิภาพดียิ่งขึ้น ขั้นตอนถัดไปคือการควอนไทซ์ (Quantization) เป็นการบีบอัดข้อมูลที่ลดปริมาณข้อมูลได้ดีที่สุดแต่ก็มีข้อเสียจากการตัดข้อมูลที่มีความสำคัญน้อยทิ้งไป และขั้นตอนสุดท้ายคือการเข้ารหัสเอนโทรปี (Entropy coding) เป็นการบีบอัดแบบไร้ความสูญเสีย สำหรับขั้นตอนการถอดรหัสแต่ละขั้นตอนจะดำเนินการผกผันกับแต่ละขั้นตอนของการเข้ารหัส และลำดับของการทำงานที่ผกผันนี้จะเรียงตรงกันข้ามกับขั้นตอนการเข้ารหัส

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

2.2 การเข้ารหัสแปลง

โดยทั่วไปในข้อมูลภาพค่าของจุดภาพในบริเวณใกล้เคียงกันจะมีสหสัมพันธ์ (Correlation) สูงระหว่างตัวอย่างจุดภาพ (Sample Pixel) การแปลงที่เหมาะสมจะสามารถลดสหสัมพันธ์ระหว่างจุดภาพได้ เมื่อผ่านการแปลงแล้วจุดภาพจะเรียกว่าสัมประสิทธิ์ (Coefficient) เมื่อแยกพิจารณาสัมประสิทธิ์ออกเป็น 2 กลุ่มคือ กลุ่มสัมประสิทธิ์ที่มีความแปรปรวนสูง และกลุ่มของสัมประสิทธิ์ที่มีความแปรปรวนต่ำ เนื่องจากค่าความแปรปรวนนี้เองเป็นสิ่งที่บอกถึงพลังงานของข้อมูลภาพ ซึ่งพลังงานส่วนใหญ่ของข้อมูลภาพจะอยู่ในกลุ่มของสัมประสิทธิ์ที่มีความแปรปรวนสูง [4] ถ้าการแปลงที่ใช้มีความเหมาะสมต่อชนิดข้อมูลแล้วสัมประสิทธิ์ส่วนมากจะอยู่ในกลุ่มที่มีความแปรปรวนต่ำ ดังนั้นการจัดสรรบิตส่วนใหญ่ให้กับสัมประสิทธิ์ที่มีความแปรปรวนสูงเป็นกลุ่มสัมประสิทธิ์ที่มีจำนวนสัมประสิทธิ์น้อยจะทำให้การบีบอัดมีประสิทธิภาพ

ยกตัวอย่างการแปลงภาพที่แบ่งเป็นบล็อก 2x2 จุดภาพ การแปลงที่ใช้กับจุดภาพทั้ง 4 ทำโดยการหาค่าเฉลี่ยของจุดภาพทั้งหมดเป็นสัมประสิทธิ์ตัวแรก และค่าความแตกต่างระหว่างจุดภาพอีก 3 จุดที่เหลือกับค่าเฉลี่ยหรือสัมประสิทธิ์จะเป็นสัมประสิทธิ์ตัวที่เหลือ ผลจากการแปลงเป็นดังรูปที่ 2.2 ค่าของจุดภาพในบล็อกเป็น 32, 38, 34 และ 36 ค่าเฉลี่ยที่ได้เป็น 35 และความแตกต่างระหว่างอีก 3 จุดที่มีค่า 38, 34 และ 36 คือ 3,-1 และ 1 ตามลำดับ เมื่อทำการแปลงแล้วสัมประสิทธิ์ที่ใช้แทนค่าเฉลี่ยจะเป็น 25% ของสัมประสิทธิ์ทั้งหมดซึ่งมีค่าความแปรปรวนสูง ในขณะที่ 75% ที่เหลือซึ่งเป็นค่าความแตกต่างกับค่าเฉลี่ยมีค่าที่ใกล้เคียงกับศูนย์ ยกตัวอย่างในกรณีที่มีการตัดสัมประสิทธิ์ที่เป็นความแตกต่างทิ้งเป็นการบีบอัดข้อมูลแบบมีความสูญเสีย จุดภาพที่สร้างกลับก็จะได้ค่าเป็น 35 ทั้งหมดซึ่งได้ค่าค่อนข้างใกล้เคียงกับค่าดั้นแบบ การแปลงจุดภาพของแต่ละบล็อกสามารถแทนด้วยสมการที่ (2.1) สำหรับการแปลงผกผันแทนด้วยสมการที่ (2.2)



รูปที่ 2.2 ตัวอย่างการแปลงเพื่อการบีบอัดข้อมูล

$$y(m,n)=\begin{cases}0.25\sum\limits_{i,j\in\{0,1\}}x(m+i,n+i) & m=n=0\\ x(m,n)-y(0,0) & m,n\neq 0\end{cases} \tag{2.1}$$

$$\hat{x}=(m,n)=\begin{cases}y(0,0)+y(m,n) & m,n\neq 0\\ 4y(0,0)-\hat{x}(0,1)-\hat{x}(1,0)-\hat{x}(1,1) & m=n=0\end{cases} \tag{2.2}$$

การเข้ารหัสแปลงที่ใช้ในการบีบอัดข้อมูลภาพส่วนมากจะเป็นการแปลงเชิงเส้นจำกัดมิติ (Finite Dimensional Linear Transforms) ซึ่งการแปลงจะส่งค่าเวกเตอร์ที่มี n มิติไปยังอีกเวกเตอร์ที่มี m มิติ ตามสมการที่ (2.3) ในที่นี้ A คือเมทริกซ์ $n\times m$ มิติ ของสัมประสิทธิ์ที่เป็นจำนวนจริง หรือจำนวนจินตภาพ และ A^* เป็นเมทริกซ์สลับเปลี่ยนสังยุค (Conjugate Transpose) แต่สำหรับการแปลงที่ใช้กับจำนวนจริงแล้ว $A^*=A^t$

$$\vec{y}=A^*\vec{x},\quad \vec{x}=\begin{pmatrix}x_0\\ x_1\\ \vdots\\ x_{n-1}\end{pmatrix},\quad \vec{y}=\begin{pmatrix}y_0\\ y_1\\ \vdots\\ y_{m-1}\end{pmatrix} \tag{2.3}$$

การแปลงผกผันใช้สมการที่ (2.4) ซึ่ง S เป็นเมทริกซ์ $n\times m$ ผกผันด้านซ้ายของ A^* ดังนั้น $SA^*=I$ เมทริกซ์เอกลักษณ์ที่มีขนาด $n\times n$ มิติ แต่เมทริกซ์ S ไม่จำเป็นที่จะมีค่าเพียงรูปแบบเดียวถ้าหากการแปลงเป็นการแปลงที่มีการส่งไม่เท่ากันระหว่างมิติของเวกเตอร์นำเข้า (Input Vector) กับเวกเตอร์นำออก (Output Vector) ไม่เท่ากัน ดังนั้นการแปลงที่ใช้จำกัดให้ $m=n$

$$\vec{x}=S\vec{y} \tag{2.4}$$

การแปลงที่ใช้สำหรับการบีบอัดข้อมูลภาพแบ่งออกเป็น 2 แบบ คือ การแปลงแบบบล็อก (Block Transform) ภาพต้นแบบจะถูกแบ่งออกเป็นบล็อกแล้วทำการแปลงแต่ละบล็อกอย่างอิสระต่อกัน ข้อดีของการแปลงแบบบล็อกคือ ใช้หน่วยความจำในการประมวลผลน้อย แต่มีข้อเสียคือ จุดภาพที่อยู่ติดกันแต่ว่าถูกแบ่งให้อยู่คนละบล็อกเมื่อทำการแปลงแล้วค่าของจุดภาพที่สร้างกลับอาจเกิดความเพี้ยนหรือสังเกตเห็นความไม่ต่อเนื่อง

แบบที่สองเรียกว่าการแปลงแบบแบนด์ย่อย (Subband Transform) ภาพทั้งภาพจะนำไปเข้ารหัสแปลงหรือจะเปรียบว่าภาพทั้งภาพเป็นบล็อกเดียว ซึ่งจะลดสหสัมพันธ์ระหว่างแบนด์ย่อยลง การเข้ารหัสแปลงต้องใช้ตัวกรองความถี่ที่มีมูลฐานที่ตั้งฉากกันในการลดสหสัมพันธ์ เมื่อสัญญาณภาพผ่านเข้าตัวกรองแล้วจะถูกแบ่งออกมาเป็นแบนด์ย่อยของความถี่ในย่านต่าง ๆ สัมประสิทธิ์ที่อยู่ในแต่ละแบนด์ย่อยจะถูกชักตัวอย่างลง (Down-Sampling) เพื่อลดจำนวน

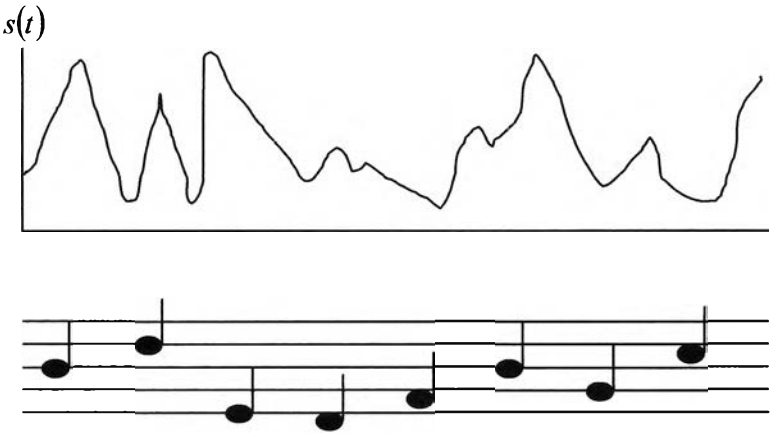
สัมประสิทธิ์ที่เพิ่มขึ้นจากผลที่ได้ของแต่ละแบนด์ย่อย แต่การชักตัวอย่างลงจะก่อให้เกิดการซ้อนทับของสัญญาณ (Aliasing) ซึ่งต้องมีการป้องกันในกระบวนการสร้างกลับ

สัญญาณภาพดิจิทัลนั้นจะอยู่ในรูปของเมทริกซ์ 2 มิติ ซึ่งการแปลงที่ใช้เป็น 1 มิติ ดังนั้นการแปลงข้อมูลภาพจะต้องใช้การแปลง 1 มิติแต่กระทำกับข้อมูลในแนวตั้ง กับแนวนอน ทีละแถวทีละหลัก ซึ่งจะสลับลำดับก่อนหลังของการแปลงแนวนอนกับแนวตั้งก็ได้ สำหรับการแปลงที่ใช้ในมาตรฐาน JPEG2000 เรียกว่าการแปลงเวฟเลต (Wavelet Transform)

2.2.1 การแปลงเวฟเลต

เวฟเลต (Wavelet) [4] หมายความว่าคลื่นลูกเล็ก เวฟเลตเป็นวิธีการวิเคราะห์สัญญาณวิธีหนึ่งโดยแทนสัญญาณว่าเป็นการประกอบกันของคลื่นลูกเล็กหลาย ๆ ลูกให้อยู่ในรูปคลื่นที่ใช้แทนสัญญาณ โดยมีมูลฐานหลักของคลื่นคือมาเธอร์เวฟเลต (Mother Wavelet) ซึ่งกำหนดให้เป็นสัญญาณที่ผลรวมภายในไม่เป็นศูนย์ทั้งหมดและมีค่าอยู่ในช่วงเวลาแคบ วิธีการใช้เวฟเลตวิเคราะห์สัญญาณจะนำมาเธอร์เวฟเลตมาเข้ารหัสแปลงกับสัญญาณ ผลที่ได้จะบอกได้ว่าสัญญาณในช่วงเวลาแคบนั้นมีสหสัมพันธ์กับมาเธอร์เวฟเลตตามค่าสัมประสิทธิ์ เมื่อทำการเลื่อนมาเธอร์เวฟเลตไปที่ช่วงเวลาอื่นจะสามารถพิจารณาว่าสัญญาณมีลักษณะเหมือนกับมาเธอร์เวฟเลตตามสัมประสิทธิ์ จนกระทั่งเลื่อนช่วงเวลาทั้งหมดเท่ากับช่วงของสัญญาณทั้งหมด พารามิเตอร์ที่บอกระดับลักษณะของสัญญาณว่าเหมือนกับมาเธอร์เวฟเลตจะมีอยู่ 2 ตัวคือ การเลื่อนทางแกนเวลาและการสเกล

ถ้าจะเปรียบเทียบเวฟเลตแล้วสามารถยกตัวอย่างที่เข้าใจได้อย่างง่าย ๆ คือ $s(t)$ เป็นสัญญาณเสียงซึ่งมีความต่อเนื่องทางเวลา และเวฟเลตคือตัวโน๊ตหนึ่งตัว โดยที่สัญญาณเสียง $s(t)$ สามารถแทนด้วยตัวโน๊ตหลาย ๆ ตัวประกอบกันเป็นสัญญาณเสียงขึ้นดังรูปที่ 2.3 สัญญาณเสียงสามารถแทนด้วยเวฟเลตที่มีการเลื่อนในทางแกนเวลา และในทางความถี่ ตัวโน๊ตที่อยู่ในบันไดเสียงขั้นที่สูงแสดงให้เห็นว่าสัญญาณส่วนนั้นประกอบด้วยองค์ประกอบความถี่สูง และสำหรับตัวโน๊ต 1 ตัวก็จะแทนค่าให้กับองค์ประกอบของสัญญาณได้ในช่วงเวลาสั้น ๆ หรือเรียกว่าแบ่งตามห้องเสียง ซึ่งแทนช่วงสัญญาณให้กับเวลาที่แตกต่างกัน เรียกได้ว่าการแปลงเวฟเลตเป็นการวิเคราะห์สัญญาณพร้อมกันทั้งทางเวลา



รูปที่ 2.3 สัญญาณ $s(t)$ กับตัวโน๊ตที่แทนสัญญาณ $s(t)$

2.2.2 เวฟเลตกับการวิเคราะห์หลายระดับความละเอียด

ในการวิเคราะห์หลายระดับความละเอียด (Multi-Resolution) ของการแปลงเวฟเลต 1 มิติจะใช้ฟังก์ชันสเกล $\varphi(t)$ และมาเธอร์เวฟเลต $\psi(t)$ สอดคล้องตามตัวกรองความถี่ต่ำและตัวกรองความถี่สูงในตัวกรองการประกอบตามลำดับ ซึ่งฟังก์ชันมูลฐานเวฟเลตทั้งสองได้มาจากการเลื่อนและการสเกลมาเธอร์เวฟเลต [5] มูลฐานของเวฟเลตที่ใช้จะอยู่บนปริภูมิฮิลเบิร์ต (Hilbert Space, $L^2(R)$) สมการที่ (2.5) แทนมาเธอร์เวฟเลต

$$\psi_{a,b}(t)=\frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right) \tag{2.5}$$

โดย a และ b เป็นพารามิเตอร์การสเกล และพารามิเตอร์การเลื่อน ซึ่งพารามิเตอร์ 2 ตัวนี้ส่วนใหญ่แล้วจะมีค่าตามสมการที่ (2.6)

$$a=2^{-m}, b=n\cdot 2^{-m} \tag{2.6}$$

โดย m กับ n เป็นจำนวนเต็มเมื่อแทนค่าสมการที่ (2.5) ด้วยสมการที่ (2.6) ฟังก์ชันเวฟเลตมูลฐานเขียนได้ดังสมการที่ (2.7)

$$\psi_{m,n}(t)=2^{\frac{m}{2}}\psi\left(2^m\cdot t-n\right) \tag{2.7}$$

ฟังก์ชันการสเกล 1 มิติสามารถหาได้จากฟังก์ชั่นสเกลในเชิงเดียวกันได้เป็นสมการที่ (2.8)

$$\varphi_{m,n}(t)=2^{\frac{m}{2}}\varphi\left(2^m\cdot t-n\right) \tag{2.8}$$

ถ้ากำหนดให้ $f(t)$ สามารถกระจายออกเป็นสองพจน์ $f(t)=f^{(m)}(t)+e$ โดยที่ $f^{(m)}(t)$ เป็นเงา (Projection) ที่ได้จากสัญญาณ $f(t)\in L^2(R)$ บนระนาบ $V^{(m)}$ นั่นคือ $f^{(m)}(t)\in V^{(m)}$ และ e เป็นส่วนที่ตั้งฉากกับระนาบ $V^{(m)}$ ผลคูณภายใน (Inner Product) ของสัญญาณที่ได้ทั้งสองจะต้องเท่ากับศูนย์ $\left\langle f^{(m)}(t),e\right\rangle=0$ และกำหนดให้ e อยู่ในระนาบ $W^{(m)}$

$f^{(m)}(t)$ ได้มาจากสัญญาณ $f(t)$ ผ่านตัวกรองความถี่ต่ำและ e ได้จากตัวกรองความถี่สูง สัมประสิทธิ์ของเวฟเลตกับฟังก์ชันสเกลสามารถหาได้จากผลคูณภายใน ดังสมการที่ (2.9) เมื่อ m,n และ k เป็นจำนวนเต็ม $m,n,k\in Z$

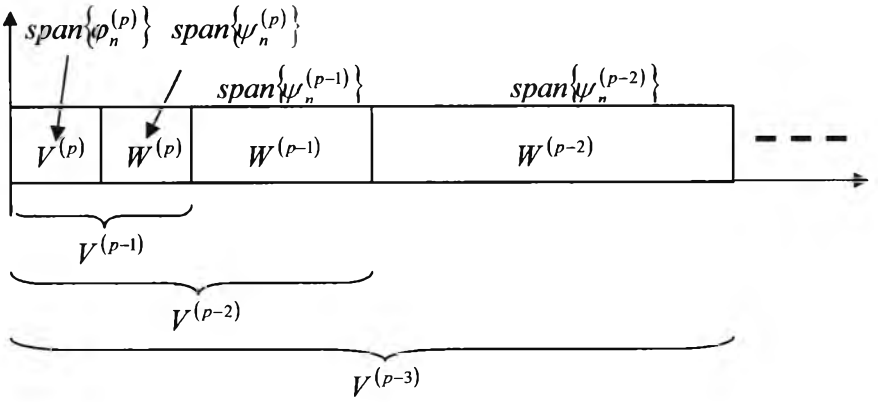
$$\begin{aligned} f(t)&=\sum_{m,n\in Z}\left\langle f,\psi_{m,n}\right\rangle\cdot\psi_{m,n}(t)\\ &=\sum_{n\in Z}\left\langle m,\varphi_{m,n}\right\rangle\cdot\varphi_{m,n}(t)+\sum_{k\le j,n\in Z}\left\langle f,\psi_{k,n}\right\rangle\psi_{k,n}(t)\end{aligned} \tag{2.9}$$

สัญญาณ $f(t)$ จะถูกแยกออกเป็น 2 ส่วน ส่วนแรกเรียกว่าส่วนการประมาณ (Approximation) ซึ่งเป็นส่วนที่ได้จากผลคูณภายในของฟังก์ชันสเกล $\varphi_{m,n}(t)$ ส่วนที่สองเรียกว่าส่วนรายละเอียด (Detail) ได้จากผลคูณภายในของฟังก์ชันเวฟเลตมูลฐาน $\psi_{m,n}(t)$ การใช้เวฟเลตวิเคราะห์หลายระดับความละเอียดจะนำส่วนการประมาณซึ่งอยู่บนระนาบ $V^{(m)}$ มาวิเคราะห์ต่อโดยแยกออกเป็นสองพจน์เช่นเดิม $f^{(m)}(t)=f^{(m+1)}(t)+e$ สัญญาณที่ได้จะแบ่งเป็นส่วนการประมาณของระนาบ $V^{(m)}$ ซึ่งอยู่บนระนาบ $V^{(m+1)}$ กับส่วนรายละเอียดซึ่งตั้งฉากกับระนาบ $V^{(m+1)}$ การวิเคราะห์สัญญาณสามารถทำซ้ำที่ส่วนการประมาณไปได้เรื่อย ๆ จึงเรียกการทำซ้ำ ๆ นี้ว่าการวิเคราะห์หลายระดับความละเอียด

เมื่อ $W^{(m)}$ เป็นส่วนเติมเต็มออโธโกนอล (Orthogonal Complement) ของ $V^{(m)}$ แล้ว ระนาบ $V^{(m)}$ มีมูลฐานที่แผ่ทั่วระนาบคือ $\varphi_n^{(m)}$ และระนาบ $W^{(m)}$ มีมูลฐานที่แผ่ทั่วระนาบคือ $\psi_n^{(m)}$ เมื่อนำมูลฐานของทั้งสองระนาบนี้มารวมกัน จะได้ว่ามูลฐานของทั้งสองระนาบนี้แผ่ทั่วในระนาบ $V^{(m-1)}$ ตามสมการที่ (2.10)

$$V^{(m-1)}=V^{(m)}\oplus W^{(m)} \tag{2.10}$$

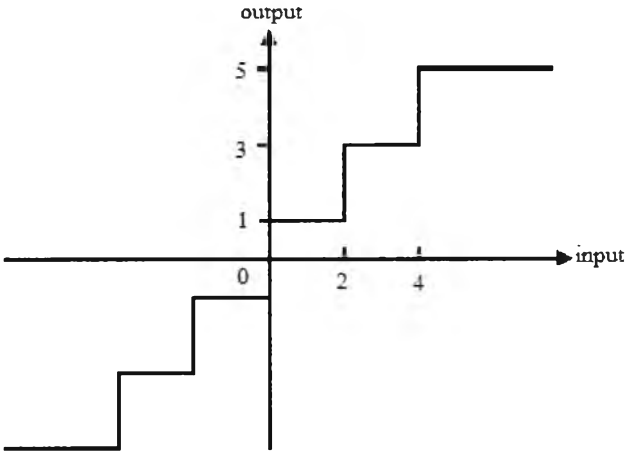
ด้วยคุณสมบัติการเป็นส่วนเติมเต็มออโธโกนอลของส่วนการประมาณและส่วนรายละเอียดแล้ว ทำให้เวฟเลตสามารถสังเคราะห์สัญญาณใด ๆ ที่อยู่ในระนาบ $V^{(p)}\in L^2(R)$ ได้ โดย $p\in Z$ ตามแผนภูมิในรูปที่ 2.4



รูปที่ 2.4 แผนภูมิการแผ่ทั่วของมูลฐานของหลายระดับความละเอียด

2.3 การควอนไทซ์

สัญญาณแอนะล็อกเป็นสัญญาณที่มีความต่อเนื่องของระดับแอมพลิจูด การแทนค่าแอมพลิจูดด้วยบิตข้อมูลแต่ว่าต้องใช้จำนวนบิตเป็นจำนวนอนันต์ในการส่งไปให้ครบทุกระดับในระบบการมองเห็นของมนุษย์นั้นสามารถรับรู้ได้ในขอบเขตที่จำกัด ซึ่งหมายความว่าสัญญาณต้นแบบสามารถแทนได้ด้วยค่าที่ประมาณให้มีความใกล้เคียงจากสัญญาณดิสครีต การควอนไทซ์ที่ใช้ในการบีบอัดสัญญาณนั้นจะเป็นการลดจำนวนบิตที่ใช้แทนข้อมูล ซึ่งเป็นการบีบอัดแบบมีความสูญเสีย เนื่องจากมีการทิ้งข้อมูลบางส่วนที่มีรายละเอียดหรือความสำคัญน้อยไป ดังรูปที่ 2.5 ลักษณะเฉพาะ (Characteristic) ของค่านำเข้า (Input) กับค่านำออก (Output) จากตัวควอนไทซ์ [6]



รูปที่ 2.5 ลักษณะเฉพาะของค่านำเข้ากับค่านำออก

2.4 การเข้ารหัสเอนโทรปี

เอนโทรปี (Entropy) [7] คือปริมาณที่ใช้วัดระดับของข่าวสาร ถ้าข่าวสารมีค่ามาก นั่นหมายความว่าโอกาสที่จะเกิดเหตุการณ์มีความน่าจะเป็นต่ำ เมื่อเหตุการณ์ใดที่คาดไม่ถึงได้เกิดขึ้นเหตุการณ์นั้นจะถือเป็นข่าว และในทางกลับกันข่าวสารที่มีค่าน้อยก็หมายความว่าโอกาสที่จะเกิดเหตุการณ์มีความน่าจะเป็นสูง คนทั่วไปรู้อยู่แล้วว่าจะต้องเกิดขึ้นอย่างแน่นอนจึงไม่เป็นข่าว

Claude E. Shannon ได้ให้นิยามของปริมาณข่าวสารในทางคณิตศาสตร์มีหน่วยเป็นเอนโทรปี สมมติว่าเรามีเหตุการณ์อยู่ในเซต X ซึ่งมีสมาชิกเป็นเหตุการณ์ทั้งหมด โดยที่เหตุการณ์ที่ i แทนด้วยสัญลักษณ์ (Symbol) x_i และความน่าจะเป็นที่จะเกิดเหตุการณ์ที่ i แทนด้วย p_i เอนโทรปีอันดับที่ 0 ของเซตนี้จะหาได้จากสมการที่ (2.11)

$$H(X) = -\sum_i p(x_i) \log_b p(x_i) \tag{2.11}$$

b แทนหน่วยของข่าวสาร สำหรับค่า $b = 2$ หน่วยของข่าวสารเป็นบิต (Bit)

จากทฤษฎีของ Shannon การเข้ารหัสข่าวสารที่แทนด้วยสัญลักษณ์สามารถทำได้ที่ค่าเฉลี่ยของปริมาณเอนโทรปีในกรณีที่ทุก ๆ สัญลักษณ์เป็นอิสระต่อกัน กรณีที่แต่ละสัญลักษณ์ไม่อิสระต่อกัน ดังเช่นการเข้ารหัสข้อมูลภาพปริมาณของเอนโทรปีจะสูงกว่าค่าเฉลี่ยของเอนโทรปีอันดับที่ 0 จึงต้องมีการหาค่าความน่าจะเป็นของสัญลักษณ์ที่มีความสัมพันธ์กับจุดภาพที่บริเวณใกล้เคียงกันในขณะทำการเข้ารหัสเรียกว่าบริบท (Context)

2.4.1 การเข้ารหัสเลขคณิต

ในการเข้ารหัสเลขคณิต (Arithmetic Coding) [7] สัญลักษณ์ทุกตัวของแหล่งข้อมูลจะถูกระบุช่วงเรียงตามความน่าจะเป็น ซึ่งทั้งหมดจะต้องอยู่ในช่วงของจำนวนจริงหนึ่งหน่วย (Unit Interval) $[0,1)$ หรือจะมองว่าการเข้ารหัสเลขคณิตเป็นการแทนค่าสัญลักษณ์ด้วยตัวเลขที่อยู่ในช่วงจำนวนจริงหนึ่งหน่วย ซึ่งเซตนี้เป็นเซตกระชับ (Compact Set) สามารถประกอบขึ้นจากเซตจำนวนอนันต์ ดังนั้นทุกสัญลักษณ์จะสามารถแทนค่าให้อยู่ในช่วงนี้ได้ทั้งหมด

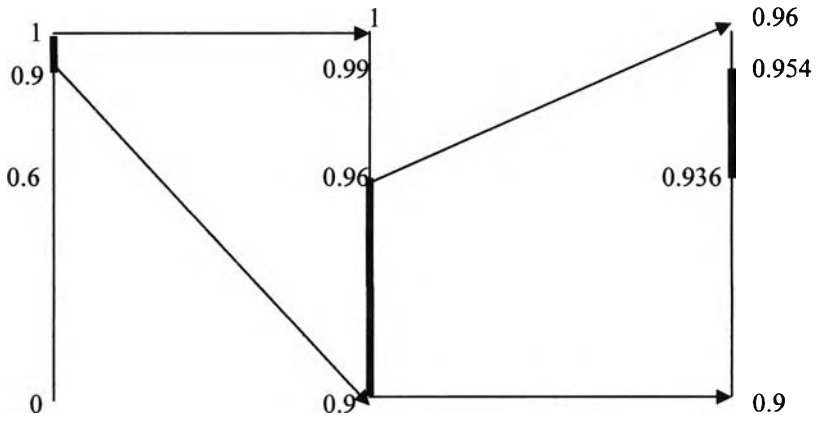
การเข้ารหัสเลขคณิตจะใช้ฟังก์ชันการแจกแจงสะสม (Cumulative Distribution Function, CDF) ในการถ่ายแบบ (Mapping) สัญลักษณ์ให้อยู่ในช่วงของจำนวนจริงหนึ่งหน่วย เรียงกันตามลำดับ กระบวนงานของตัวเข้ารหัสเลขคณิตจะทำงานตามตัวอย่าง กรณีที่มีแหล่งข้อมูลที่ให้

ข่าวสารออกมาแทนเป็นสัญลักษณ์ได้ 3 แบบ s_1, s_2 และ s_3 ซึ่งมีความน่าจะเป็นของสัญลักษณ์เท่ากับ 0.6, 0.3, 0.1 ตามลำดับดังแสดงในตารางที่ 2.1

ตารางที่ 2.1 ความน่าจะเป็น และช่วง ที่ได้สัญลักษณ์ s_1, s_2 และ s_3

สัญลักษณ์	ความน่าจะเป็น	ช่วง
s_1	0.6	[0,0.6)
s_2	0.3	[0.6,0.9)
s_3	0.1	[0.9,1)

ถ้าสัญลักษณ์ที่ต้องการจะส่งเป็น $s_3 s_1 s_2$ ตัวเลขที่ได้จะต้องอยู่ในช่วงของสัญลักษณ์ตัวแรก นั่นคือ [0.9,1) ดังรูปที่ 2.6 สำหรับการเข้ารหัสสัญลักษณ์ตัวถัดไปจะต้องนำช่วง [0.9,1) มาแบ่งตามสัดส่วนของความน่าจะเป็นเช่นเดิมจะได้ช่วงของสัญลักษณ์เป็น [0.9,0.96), [0.96,0.99) และ [0.99,1) สำหรับสัญลักษณ์ตัวที่ 2 คือ s_1 ตัวเลขที่ได้ก็จะต้องอยู่ในช่วง [0.9,0.96) และทำการแบ่งช่วงซ้ำอีกครั้งเพื่อทำการเข้ารหัสสัญลักษณ์ตัวถัดไป ซึ่งช่วงของแต่ละสัญลักษณ์จะเป็น [0.9,0.936), [0.936,0.954) และ [0.954,0.96) ซึ่งสัญลักษณ์ตัวที่ 3 คือ s_2 ดังนั้นค่าตัวเลขที่ได้จะต้องอยู่ในช่วง [0.936,0.954)



รูปที่ 2.6 กระบวนการเข้ารหัสเลขคณิตกับการขยายช่วง

จากกระบวนการเข้ารหัสเลขคณิตจะได้ตัวเลขที่ใช้แทนค่าลำดับของสัญลักษณ์มา ซึ่งการประมวลผลสัญญาณที่ใช้ระบบดิจิทัลจะใช้เลขฐาน 2 แทนในการคำนวณค่าตัวเลขนี้ใน

ขั้นตอนการเข้ารหัส การเข้ารหัสเลขคณิตจะใช้ขั้นตอนวิธีตาม [7] หลักสำคัญของขั้นตอนวิธีนี้คือการสเกลช่วง เมื่อจำนวนสัญลักษณ์ที่จะทำการเข้ารหัสเลขคณิตเป็นลำดับที่มีความยาวมาก ช่วงความน่าจะเป็นที่ค่าจะตกอยู่จะแคบมาก ตัวเลขที่จะใช้แทนค่าในช่วงที่มีความละเอียดสูงต้องใช้จำนวนบิตที่มากในการแทนค่าตัวเลขนี้ การหลีกเลี่ยงปัญหานี้สามารถทำได้โดยการสเกลช่วงในระหว่างการเข้ารหัส ถ้าช่วงของความน่าจะเป็นอยู่ในครึ่งล่างของช่วงหนึ่งหน่วยก็จะทำการส่งบิต 0 ออกมาและจะทำการขยายช่วงเป็น 2 เท่า ดังสมการที่ (2.12) กับ (2.13) ถ้าช่วงของความน่าจะเป็นอยู่ในครึ่งบนของช่วงหนึ่งหน่วยก็จะทำการส่งบิต 1 ออกมาและจะทำการขยายช่วงเป็น 2 เท่าเช่นกัน กรณีที่ทำการขยายช่วงแล้ว ช่วงยังอยู่ในครึ่งบนหรือครึ่งล่างของช่วงหนึ่งหน่วยก็จะทำซ้ำไปเรื่อย ๆ การเข้ารหัสลำดับสัญลักษณ์ $s_3 s_1 s_2$ แสดงเป็นตารางที่ 2.2 ขั้นตอนสุดท้ายใส่ค่าบิตใด ๆ ปิดท้ายสายบิต (Bit-stream) ซึ่งทำให้ตัวเลขอยู่ในช่วงของความน่าจะเป็นของสัญลักษณ์ในอันดับที่สุดท้าย ซึ่งมีช่วงเป็น $[0.936,0.954)$ แต่บิตที่ทำการเข้ารหัสเป็น 0.111 นั่นคือ 0.875 ดังนั้นเพื่อให้ค่าอยู่ในช่วงจึงใส่บิต 1 เข้าไปเป็น 0.1111 ค่าที่ได้จะเป็น 0.9375

$$x' = \{x \in [0,0.5) | \, x' = 2x\} \tag{2.12}$$

$$x' = \{x \in [0.5,1) | \, x' = 2(x - 0.5)\} \tag{2.13}$$

x แทนขอบเขตของช่วง x' แทนขอบเขตของช่วงหลังการขยายช่วง

ตารางที่ 2.2 การเข้ารหัสเลขคณิตของลำดับ $s_3 s_1 s_2$

สัญลักษณ์	การทำงาน	ช่วง	บิตที่เข้ารหัส
s_3	เลือกช่วงความน่าจะเป็นของ s_3	$[0.9,1)$	1
	ขยายช่วง	$[0.8,1)$	1
	ขยายช่วง	$[0.6,1)$	1
	ขยายช่วง	$[0.2,1)$	-
s_1	เลือกช่วงความน่าจะเป็นของ s_1	$[0.2,0.68)$	-
s_2	เลือกช่วงความน่าจะเป็นของ s_2	$[0.488,0.632)$	1

การถอดรหัสจะต้องทำการเลือกขนาดหน้าต่าง n ที่ใช้สำหรับการถอดรหัส ซึ่งจะเริ่มทำการถอดรหัสจาก n บิตแรกของสายบิต ในขั้นตอนการถอดรหัสหน้าต่างจะค่อย ๆ เลื่อนไป บิตที่มีนัยสำคัญสูงจะถูกตัดออกไป และบิตที่มีนัยสำคัญน้อยที่อยู่นอกหน้าต่างจะเลื่อนเข้าเพื่อทำการถอดรหัส ขนาดของหน้าต่างจะต้องมีขนาดใหญ่พอที่จะทำการแทนค่าของตัวเลขช่วงความน่าจะเป็นของสัญลักษณ์ที่มีค่าน้อยที่สุดสำหรับกรณีนี้ช่วงความน่าจะเป็นที่มีค่าต่ำที่สุดเป็นของสัญลักษณ์ s_3 มีค่าความน่าจะเป็น 0.1 ดังนั้นขนาดของหน้าต่างที่ใช้จะต้องเป็นอย่างน้อย $n=4$ การถอดรหัสจะต้องเริ่มจาก 4 บิตแรกนั่นก็คือ 1111 ซึ่งมีค่าสอดคล้องกับ 0.9735 และมีค่าอยู่ในช่วง $[0.9,1)$ แสดงว่าสัญลักษณ์ตัวแรกคือ s_3 ช่วง $[0.9,1)$ นั้นมีค่าอยู่ในครึ่งบนของช่วงหนึ่งหน่วยดังนั้นจึงทำการเลื่อนหน้าต่างไป 1 บิต และต้องทำการขยายช่วงเป็น $[0.8,1)$ สายบิตตอนนี้มีค่าเป็น 1110 ซึ่งมีค่าสอดคล้องกับ 0.875 แต่ช่วงยังอยู่ในครึ่งบนของช่วงหนึ่งหน่วย จึงต้องทำการเลื่อนหน้าต่างไปอีกจนกระทั่งช่วงตกอยู่ระหว่าง $[0.2,1)$ สำหรับตอนนี้ค่าของสายบิตเป็น 1000 ซึ่งมีค่าสอดคล้องกับ 0.5 และอยู่ใน 60% แรกของช่วง $[0.2,1)$ ดังนั้นสัญลักษณ์ตัวถัดมาที่ถอดรหัสได้คือ s_1 แล้วทำการปรับช่วงใหม่เป็น $[0.2,0.68)$ และค่าของสายบิตยังคงเป็น 1000 ซึ่งมีค่าสอดคล้องกับ 0.5 ซึ่งมีค่าอยู่ระหว่าง 60% ถึง 90% ของช่วง $[0.2,0.68)$ ดังนั้นสัญลักษณ์ที่ถอดรหัสออกมาเป็นตัวสุดท้ายคือ s_2 สำหรับตัวถอดรหัสจะทราบว่าจำนวนสัญลักษณ์ที่ถูกเข้ารหัสมามีทั้งหมดจำนวนเท่าใดเพื่อให้ทราบว่าถึงจุดสิ้นสุดของข้อมูลสำหรับกรณีนี้ตัวถอดรหัสจะหยุดการทำงานเมื่อได้สัญลักษณ์ออกมา 3 ตัว ขั้นตอนการถอดรหัสแสดงดังตารางที่ 2.3

ตารางที่ 2.3 การถอดรหัสของลำดับ $s_3s_1s_2$

ช่วง	หน้าต่าง	ค่าตัวเลข	สัญลักษณ์ที่ได้
$[0.9,1)$	1111	0.9375	s_3
$[0.8,1)$	1110	0.875	-
$[0.6,1)$	1100	0.75	-
$[0.2,1)$	1000	0.5	-
$[0.2,0.68)$	1000	0.5	s_1
$[0.488,0.632)$	1000	0.5	s_2

2.5 ภาพรวมของมาตรฐาน JPEG2000

มาตรฐานการบีบอัดภาพนิ่งที่กำหนดโดย ISO/ITU-T ก่อนหน้านี้ได้กำหนดมาตรฐานที่ใช้กันอย่างแพร่หลายคือมาตรฐาน JPEG [8] ขั้นตอนของการเข้ารหัสข้อมูลภาพเริ่มจากแปลงบล็อกดิสครีตโคไซน์ (Block Discrete Cosine Transform, DCT) 2 มิติ จากนั้นทำการควอนไทซ์ค่าของสัมประสิทธิ์เพื่อให้ค่าสัมประสิทธิ์มีขนาดที่เล็กลง และขั้นตอนถัดมาเป็นการจัดรูปแบบข้อมูลให้ง่ายต่อการทำการเข้ารหัสเอนโทรปีซึ่งมาตรฐาน JPEG เรียงค่าสัมประสิทธิ์แบบซิกแซ็ก (Zigzag) และสุดท้ายเป็นการเข้ารหัสเอนโทรปีเป็นการบีบอัดข้อมูลในขั้นตอนสุดท้ายด้วยการบีบอัดแบบไร้ความสูญเสีย การเข้ารหัสฮัฟแมน (Huffman Coding) เป็นการเข้ารหัสเอนโทรปีที่ใช้ในมาตรฐาน JPEG ผลที่ได้ก็ยังมีข้อจำกัดอยู่ที่การแปลงที่ใช้เป็นแบบบล็อกดิสครีตโคไซน์ที่ทำการเข้ารหัสอย่างอิสระต่อกันทำให้ค่าของสัมประสิทธิ์ระหว่างบล็อกมีค่าไม่ต่อเนื่องกัน เมื่อการเข้ารหัสภาพที่อัตราบิตต่อจุดภาพ (Bit per pixel, bpp) ต่ำจะทำให้เกิดความไม่ต่อเนื่องกันของข้อมูลระหว่างบล็อกมากยิ่งขึ้น ส่งผลให้เกิดบล็อกเทียม (Blocking Artifact) หลังจากการถอดรหัส ภาพที่ได้จะมีความแตกต่างของจุดภาพที่ขอบของบล็อกสูง

ISO กับ ITU-T จึงได้ร่วมมือกันออกแบบมาตรฐาน JPEG2000 ซึ่งเป็นมาตรฐานการบีบอัดภาพนิ่งล่าสุดในขณะที่เขียนวิทยานิพนธ์เล่มนี้ มาตรฐาน JPEG2000 พัฒนาขึ้นเพื่อเพิ่มความสามารถของการเข้ารหัสภาพนิ่งในส่วนที่มาตรฐาน JPEG ทำได้ไม่ดี และได้เปลี่ยนแปลงทั้ง 3 ขั้นตอนหลักเปรียบเทียบกับการเข้ารหัส JPEG โดยวิธีที่ใช้ในมาตรฐาน JPEG2000 เป็นการแปลงดิสครีตเวฟเลต (Discrete Wavelet Transform, DWT) [9] แทนการแปลงดิสครีตโคไซน์ ขั้นตอนจัดเรียงค่าสัมประสิทธิ์จะทำการพิจารณาเป็นระนาบบิตและทำการเรียงข้อมูลจากระนาบบิตนัยสำคัญสูงไปยังระนาบบิตนัยสำคัญต่ำ และมีขั้นตอนวิธีที่ใช้สำหรับการจัดเรียงข้อมูลโดยการแบ่งสัมประสิทธิ์ที่จะทำการเข้ารหัสออกเป็นส่วนเพื่อความเหมาะสมในการบีบอัดข้อมูลเรียกว่า Embedded Block Coding with Optimized Truncation (EBCOT) [10] เสนอโดย D. S. Taubman และ A. Zakhor แทนการเรียงข้อมูลแบบซิกแซ็ก และในขั้นตอนสุดท้ายการเข้ารหัสเอนโทรปีใช้การเข้ารหัสเลขคณิต (Arithmetic Coding) ที่เรียกว่าตัวเข้ารหัสเอ็มคิว (MQ Coder) แทนการเข้ารหัสฮัฟแมน

ความสามารถที่เพิ่มขึ้นของมาตรฐานการบีบอัดภาพนิ่ง JPEG2000 เช่น มาตรฐานการบีบอัดภาพนิ่งก่อนหน้าเป็นแบบเข้ารหัส 1 ครั้งถอดรหัสได้ 1 แบบ โดยการเข้ารหัสจะเลือกพารามิเตอร์ที่ฝังตัวเข้ารหัส และตัวถอดรหัสจะสามารถถอดรหัสได้ตามที่ถูกกำหนด แต่สำหรับมาตรฐาน JPEG2000 จะเป็นแบบเข้ารหัส 1 ครั้งสามารถถอดรหัสได้หลายแบบ เนื่องจากสายบิต

ของ JPEG2000 สามารถถอดรหัสได้โดยแบ่งเป็นเลเยอร์ (Layer) ที่ประกอบจากข้อมูลในโค้ดบล็อก (Code-block) ซึ่งตัวถอดรหัสจะสามารถเข้าถึงข้อมูลในแต่ละส่วนด้วยมาร์คเกอร์ (Marker)

2.6 ลักษณะของ JPEG2000

2.6.1 ความสามารถในการบีบอัดสูง

JPEG2000 สามารถเข้ารหัสที่ระดับไร้ความสูญเสียได้ และยังทำงานได้ดีในระดับอัตราบิตต่ำประมาณ 0.25 บิตต่อจุดภาพ และสำหรับภาพระดับเทา (Grey Scale) สามารถทำงานได้ดีแม้อัตราบิตต่ำเพียง 0.05 บิตต่อจุดภาพ [1]

ในการบีบอัดแบบไร้ความสูญเสียมาตรฐาน JPEG2000 สามารถบีบอัดภาพได้ที่ระดับอัตราบิตต่ำกว่ามาตรฐาน JPEG Lossless (JPEG-LS)

2.6.2 การเข้ารหัสหนึ่งครั้ง ถอดรหัสได้หลายแบบ

ตัวถอดรหัสของ JPEG2000 สามารถถอดรหัสภาพจากสายบิตเดียวกันได้หลายแบบขึ้นอยู่กับว่าต้องการถอดรหัสตรงส่วนใดของสายบิต เนื่องจากสัมประสิทธิ์ของเวฟเลตมีโครงสร้างของข้อมูลแยกย่อยให้อยู่ในระดับความละเอียด และการเข้ารหัสที่เป็นแบบระนาบบิต สายบิตของ JPEG2000 สามารถเข้าถึงข้อมูลแบบสุ่ม (Random Access) ทำให้สามารถถอดรหัสตำแหน่งที่ต้องการของภาพได้กล่าวคือ

สามารถถอดรหัสขึ้นกับระดับความละเอียดโดยแบนด์ย่อยที่ได้จากการแปลงเวฟเลตในแต่ละระดับมีข้อมูลของภาพแบ่งตามระดับความละเอียด

การเข้ารหัสสัมประสิทธิ์เป็นระนาบบิตโดยเรียงระนาบบิตจากนัยสำคัญสูงไปนัยสำคัญต่ำ ทำให้ข้อมูลเรียงตามระดับคุณภาพ

ตำแหน่งของภาพสามารถถอดรหัสเฉพาะบริเวณที่ตัวถอดรหัสต้องการจากโค้ดบล็อกที่ตกอยู่ในพื้นที่นั้น ๆ

2.6.3 การถอดรหัสแบบโพรเกรสซิฟ (Progressive)

สายบิตของ JPEG2000 มีมาร์กเกอร์ (Marker) ทำให้ตัวถอดรหัสสามารถเข้าถึงข้อมูลได้อย่างอิสระ ลักษณะนี้ใช้วิธีการเหมือนกับการเข้ารหัสหนึ่งครั้ง ถอดรหัสได้หลายแบบ แต่ที่เรียกว่าการถอดรหัสแบบโพรเกรสซิฟเพราะใช้สำหรับงานที่ใช้ในการส่งข้อมูลและมีการ

แสดงผลตามเวลาจริง เช่นงานในโครงข่ายอินเทอร์เน็ตทำให้การถอดรหัสสามารถกำหนดให้คุณภาพของภาพเพิ่มขึ้นได้ระหว่างระดับความละเอียดของภาพ กับคุณภาพของภาพในขณะที่ข้อมูลรับเข้ามาตลอดเวลาจนกระทั่งครบทั้งภาพ

2.6.4 การเข้ารหัสเน้นบริเวณที่สนใจ (ROI coding)

เป็นการเข้ารหัสเพื่อให้บริเวณที่ถูกเลือกเข้ารหัสที่คุณภาพสูงกว่าบริเวณอื่น สำหรับภาพที่มีการเข้ารหัสบริเวณที่สนใจแล้ว การถอดรหัสจะทำการถอดรหัสบริเวณที่สนใจก่อน โดยการกำหนดพารามิเตอร์ที่ตัวเข้ารหัส ลักษณะนี้เหมาะกับการประยุกต์ใช้สำหรับงานด้านการจัดเก็บข้อมูล และการส่งผ่านช่องสัญญาณที่มีแบนด์วิดท์แคบ

2.7 ขั้นตอนวิธีของ JPEG2000

มาตรฐาน JPEG2000 ถูกกำหนดขั้นตอนวิธีเฉพาะส่วนของการถอดรหัส เพื่อให้เข้าใจการทำงานของ JPEG2000 ได้ง่ายกว่าจะอธิบายตามกระบวนงานการเข้ารหัส

2.7.1 ก่อนการประมวลผล

เนื่องจากข้อมูลภาพมีการจัดเก็บที่แตกต่างกันหลายวิธีรูปแบบของข้อมูลจึงมีความแตกต่างกัน ดังนั้นก่อนการเข้ารหัสตามมาตรฐาน JPEG2000 ต้องมีการจัดรูปแบบตามขั้นตอนก่อนการประมวลผล โดยแบ่งออกเป็น 3 ขั้นตอน

2.7.1.1 การแปลงองค์ประกอบสี

ในจุดภาพของภาพสีจะมีข้อมูลจากองค์ประกอบหลายองค์ประกอบมารวมกัน รูปแบบปริภูมิสี (Colour Space) ที่ใช้กันมากที่สุดคือองค์ประกอบแม่สีของแสงแบ่งเป็นสี แดง เขียว และน้ำเงิน (RGB) องค์ประกอบสีในแต่ละจุดภาพเกิดจากความเข้มของสี แดง เขียว และน้ำเงิน การแปลงองค์ประกอบสีในมาตรฐาน JPEG2000 เพื่อลดสหสัมพันธ์ขององค์ประกอบสี ทำให้เพิ่มประสิทธิภาพของการบีบอัดข้อมูล การแปลงองค์ประกอบสีที่ใช้ในมาตรฐาน JPEG2000 คือการแปลงปริภูมิสี RGB ไปเป็นปริภูมิสีแบบหนึ่งซึ่งมีความใกล้เคียงกับปริภูมิสี YCbCr โดย Y คือองค์ประกอบความสว่าง ส่วน Cb กับ Cr คือองค์ประกอบจากความแตกต่างของสีน้ำเงินกับสีแดงตามลำดับ กรณีที่จุดภาพมีค่าเป็นจำนวนเต็มบวกแล้ว จะทำการเลื่อนระดับไฟตรง (DC Level Shifting) ก่อนที่จะทำการแปลงองค์ประกอบสี ขั้นตอนนี้จะทำการเลื่อนค่าของจุดภาพจากช่วง $[0,2^s-1]$ เป็น $[-2^{s-1},2^{s-1}-1]$, s คือจำนวนบิตที่ใช้แทนค่าจุดภาพ การเลื่อนระดับไฟตรงนี้ทำโดย

การนำค่า 2^{s-1} ลบค่าของจุดภาพทุกจุด การเลื่อนระดับไฟตรงจะทำให้ผลรวมค่าแอมพลิจูดของจุดภาพทั้งหมดลดลง

การแปลงองค์ประกอบสีที่ใช้แบ่งออกเป็น 2 แบบคือแบบไม่สามารถผันกลับได้ (Irreversible Component Transform, ICT) กับแบบสามารถผันกลับได้ (Reversible Component Transform, RCT) [11] เมทริกซ์การแปลงองค์ประกอบสีของ ICT และ RCT ดังสมการที่ (2.14) และ (2.15) ตามลำดับ การแปลงองค์ประกอบสีแบบไม่สามารถผันกลับได้จะใช้กับการบีบอัดแบบมีความสูญเสีย ส่วนการแปลงองค์ประกอบสีแบบสามารถผันกลับได้จะใช้สำหรับการบีบอัดแบบไร้ความสูญเสีย

$$\begin{bmatrix} Y_0 \\ Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.144 \\ -0.16875 & -0.33126 & 0.5 \\ 0.5 & -0.41869 & -0.08131 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2.14}$$

$$\begin{bmatrix} Y_0 \\ Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{4} & \frac{2}{4} & \frac{1}{4} \\ 0 & -1 & 1 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2.15}$$

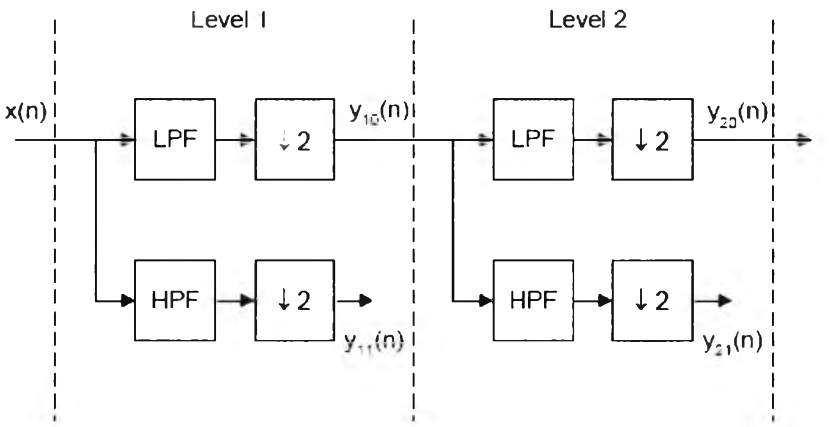
2.7.1.2 ไทล์ (Tile)

หลังจากการแปลงองค์ประกอบสีแล้วแต่ละองค์ประกอบสีจะถูกทำเป็นสี่เหลี่ยมและต้องไม่ซ้อนทับกันยกเว้นไทล์ที่อยู่บริเวณขอบด้านล่างและด้านขวาของภาพ [3] แต่เนื่องจากองค์ประกอบสีแต่ละตัวอาจมีระดับความละเอียดไม่เท่ากัน ดังนั้นขนาดของสี่เหลี่ยมที่ใช้อาจไม่เท่ากันได้ ขนาดของไทล์สามารถเป็นเท่าไหร่ก็ได้จนกระทั่งทั้งภาพเป็นไทล์เดียว สำหรับการแบ่งไทล์ในแต่ละองค์ประกอบสีขนาดของไทล์จะต้องมีขนาดเท่ากัน การแบ่งไทล์นี้ช่วยลดขนาดหน่วยความจำที่ใช้ในการประมวลผลสัญญาณ เนื่องจากการเข้ารหัสจะกระทำทีละไทล์อย่างอิสระต่อกัน การแบ่งไทล์นี้เรียกได้ว่าเป็นการแปลงแบบบล็อกอย่างหนึ่งที่ช่วยในเรื่องขนาดของหน่วยความจำที่ใช้ประมวลผลเท่านั้น แต่การแบ่งไทล์ให้มีขนาดเล็กก็มีข้อเสียเช่นกัน คือขนาดของไทล์ที่เล็กลงจะทำให้การแปลงเกิดความแตกต่างของสัมประสิทธิ์ที่อยู่ติดกันแต่อยู่คนละไทล์ เมื่อทำการถอดรหัสแล้วภาพที่สร้างกลับจะเกิดเป็นบล็อกเทียม

2.7.2 การแปลงดิสครีตเวฟเลต

การแปลงเวฟเลตที่ใช้ในมาตรฐาน JPEG2000 ข้อมูลภาพเป็นข้อมูลที่มีการอ้างถึงตำแหน่งของข้อมูลทั้งแนวตั้ง และแนวนอน วิธีการแปลงที่ใช้สำหรับข้อมูลภาพเป็นการแปลงเวฟเลต 1 มิติที่แนวตั้งจนกระทั่งครบทุกหลัก แล้วจึงทำการแปลงเวฟเลต 1 มิติที่แนวนอนจนกระทั่งครบทุกแถว

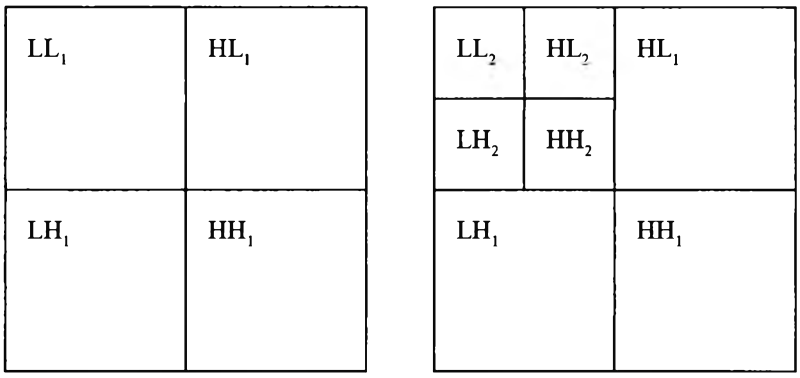
การแปลงเวฟเลต 1 มิติ [7] ถูกแสดงในรูปที่ 2.7 ในแต่ละครั้ง สัญญาณที่แยกได้จะแบ่งออกเป็น 2 แบนด์ย่อย (Subband) โดยใช้ตัวกรองความถี่ต่ำ กับตัวกรองความถี่สูง ตัวกรองทั้ง 2 เป็นตัวกรองแบบผลตอบสนองอิมพัลส์จำกัด (Finite Impulse Response, FIR) สัญญาณที่ได้จากตัวกรองความถี่ต่ำ $y_{10}(n)$ และสัญญาณที่ได้จากตัวกรองความถี่สูง $y_{11}(n)$ จะทำการชักตัวอย่างลงด้วยค่า 2 ลดจำนวนของสัมประสิทธิ์ที่เพิ่มขึ้นจากทุก ๆ แบนด์ย่อยเพื่อหลีกเลี่ยงการซ้อนทับของสัญญาณ (Aliasing) การแปลงเวฟเลตระดับต่อไปจะนำสัญญาณ $y_{10}(n)$ ซึ่งได้จากแบนด์ย่อยที่ผ่านตัวกรองความถี่ต่ำมาทำการแปลงอีกครั้ง สัญญาณที่ได้แบ่งเป็นแบนด์ย่อยที่การแปลงระดับที่ 2 และเป็นเช่นนี้เรื่อยไปจนกระทั่งหยุดที่ระดับการแปลงที่ต้องการ



รูปที่ 2.7 การแปลงเวฟเลต 1 มิติ 2 ระดับ

ในการแปลงเวฟเลต 2 มิติ สัญญาณจะแบ่งออกเป็น 4 แบนด์ย่อย คือ LL, HL, LH และ HH ตัวอักษรแรกแทนแถบความถี่ของแนวนอน และตัวอักษรตัวที่สองแทนแถบความถี่ของแนวตั้ง เช่นแบนด์ย่อย HL เป็นแบนด์ย่อยที่มีความถี่สูงในแนวนอน และแบนด์ย่อยที่มีความถี่ต่ำในแนวตั้ง ดังรูปที่ 2.8 การแปลงเวฟเลต 2 มิติในระดับความละเอียดถัดไปจะนำสัญญาณจากแบนด์ย่อย LL_1 มาทำการแปลงต่อ ได้เป็นแบนด์ย่อย LL_2, HL_2, LH_2 และ HH_2 ตัวเลขห้อยแสดงถึงแบนด์

ย่อยที่ได้จากการแปลงเวฟเลตในครั้งที่นั้น แบนด์ย่อยที่ได้จะมีจำนวนของสัมประสิทธิ์ลดลงเหลือ 1 ใน 4 จากขนาดระดับความละเอียดในรอบก่อน



รูปที่ 2.8 การแปลงเวฟเลต 2 มิติ รูปซ้าย การแปลง 1 ระดับ รูปขวา การแปลง 2 ระดับ

ตัวกรองที่ใช้ในงานประมวลผลสัญญาณภาพมีความแตกต่างกับตัวกรองที่ใช้กับงานประมวลผลสัญญาณทั่ว ๆ ไป ซึ่งให้จำนวนของสัมประสิทธิ์ที่ผ่านตัวกรองมากกว่าจำนวนของตัวอย่างก่อนที่จะเข้าตัวกรอง ในการแปลงเชิงเส้นจำนวนของสัมประสิทธิ์ที่เพิ่มจะขึ้นกับจำนวนของสัมประสิทธิ์จากตัวกรอง หากมีสัญญาณที่มีจำนวนตัวอย่าง n_1 จุดและตัวกรองมีจำนวนของสัมประสิทธิ์ n_2 จุด สัญญาณที่ได้จะมีจำนวนสัมประสิทธิ์ $n_1 + n_2 - 1$ จุด แต่จำนวนสัมประสิทธิ์ที่เพิ่มขึ้นนั้น ในงานด้านอื่นเช่นการประมวลผลสัญญาณเสียง สัญญาณนำเข้าที่เข้าตัวกรองนั้นมีจำนวนตัวอย่างที่เยอะมาก หรือเข้าใกล้อนันต์เมื่อเปรียบเทียบกับจำนวนของสัมประสิทธิ์ของตัวกรองจึงไม่มีปัญหากับค่าบริเวณขอบ แต่สำหรับสัญญาณภาพที่มีการแบ่งออกเป็นไทล์จะเกิดความเพี้ยนขึ้นที่บริเวณขอบของไทล์ด้านที่ติดกัน วิธีการแก้ปัญหานี้ให้กับตัวกรองสำหรับงานประมวลผลสัญญาณภาพ ใช้สังวัตนาการรอบ (Circular Convolution) แทนสังวัตนาการเชิงเส้น (Linear Convolution) และขยายสัญญาณเป็นคาบ ส่วนต้นและส่วนปลายของสัญญาณหรือบริเวณขอบของภาพจะทำการขยายสัญญาณเป็นคาบที่สมมาตรกัน (Periodic Symmetric extension) ดังรูปที่ 2.9 แทนสัญญาณเป็นตัวอักษร “ABCDEFG” เนื่องจากตัวกรองที่ใช้เป็นแบบผลตอบสนองอิมพัลส์จำกัด ที่มีเฟสเป็นเชิงเส้น ดังนั้นค่านำออกของสัญญาณที่ได้จะเป็นคาบที่สมมาตรเช่นกัน [5]

รูปที่ 2.9 การขยายสัญญาณสมมาตรเป็นคาบ

พิมพ์ต้นฉบับบทคัดย่อวิทยานิพนธ์ภายในกรอบสีเขียวนี้เพียงแผ่นเดียว
บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย

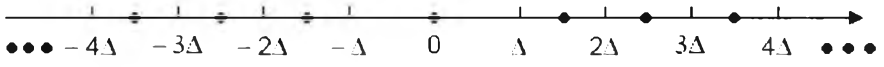
คุณสมบัติของตัวกรองที่มีเฟสของค่านำออกเป็นเชิงเส้นพบได้ในตัวกรองที่เป็นออโธโกนอลเวฟเลต (Orthogonal Wavelet) แต่สำหรับตัวกรองที่ใช้ในมาตรฐาน JPEG2000 จะเป็นตัวกรองแบบไบออโธโกนอลเวฟเลต (Biorthogonal Wavelet) ในมาตรฐาน JPEG2000 มี 2 แบบ แบบที่หนึ่งคือ 9/7 โฟลทติ้งพอยท์เวฟเลต (Floating Point Wavelet) [11] ใช้กับการแปลงเวฟแบบไม่สามารถทำกลับได้เหมือนเดิม กับแบบที่สอง 5/3 อินทิเจอร์ (Integer Wavelet) [11] ใช้กับการแปลงที่สามารถทำกลับได้

2.7.3 การควอนไทซ์

หลังจากการแปลงเวฟเลตแล้ว ขั้นตอนถัดไปคือการควอนไทซ์ ในมาตรฐาน JPEG2000 ใช้สเกลาร์ควอนไทซ์และมีเขตไร้ผลสนอง (Dead Zone) [6] ที่ค่าศูนย์ ดังรูปที่ 2.10 โดยมีช่วงที่จุดกึ่งกลางเป็น 2 เท่าของช่วงอื่นหรือเรียกช่วงนี้ว่าเขตไร้ผลสนอง ค่านำเข้าที่อยู่ในช่วงนี้จะถูกควอนไทซ์ให้มีค่าเป็น 0 ถ้าค่านำเข้ามีค่า Z และใช้ค่าควอนไทซ์เป็น Δ นำออก q หาได้จากสมการที่ (2.16) และตัวถอดรหัสจะทำการผกผันการควอนไทซ์จากสมการที่ (2.17) ค่า $\hat{Z}$ เป็นค่าที่ได้จากการผกผันการควอนไทซ์ซึ่งจะได้ค่าที่มีความใกล้เคียงกับค่า Z เนื่องจากการควอนไทซ์เป็นกระบวนงานทางการบีบอัดที่มีความสูญเสียเว้นแต่การควอนไทซ์จะใช้ค่า $\Delta = 1$ และค่า δ ในสมการที่ (2.17) เป็นพารามิเตอร์แทนขีดแบ่ง (Threshold) ปกติแล้วจะใช้ค่ากลางของช่วง $\delta = \frac{1}{2}$

$$q = sign(Z)\left\lfloor \frac{|Z|}{\Delta} \right\rfloor \tag{2.16}$$

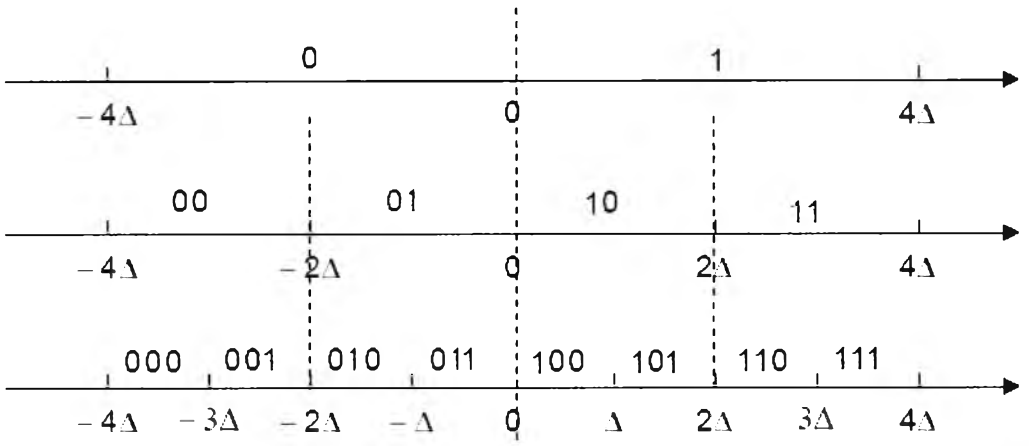
$$\hat{Z} = \begin{cases} 0 & q = 0 \\ sign(q)(|q| + \delta)\Delta & q \neq 0 \end{cases} \tag{2.17}$$



รูปที่ 2.10 การควอนไทซ์แบบยูนิฟอร์มสเกลาร์กับเขตไร้ผลสนอง

ข้อดีของการควอนไทซ์แบบยูนิฟอร์มสเกลาร์กับเขตไร้ผลสนองมี 2 อย่าง อย่างแรกคือคุณสมบัติที่จำเป็นสำหรับงานทางด้านการบีบอัดข้อมูลคือควอนไทซ์กลับมาได้จากการถอดรหัสสายบิต ในกรณีของการควอนไทซ์การประมาณค่าสัมประสิทธิ์อย่างหยาบสำหรับการควอนไทซ์ให้ใกล้เคียงกับค่าสัมประสิทธิ์ก่อนการควอนไทซ์ เมื่อตัวถอดรหัสรับบิตข้อมูลมามากขึ้นแล้ว จะได้ค่าที่ถูกต้องยิ่งขึ้นจนกระทั่งสามารถทำกลับมาได้สัมประสิทธิ์ที่มีค่าเท่ากับก่อน

การควอนไทซ์ คุณสมบัตินี้เรียกว่าการควอนไทซ์ฝัง (Embedded Quantization) ลำดับของสายบิตที่มีแทนค่าควอนไทซ์แต่ละตัวด้วยจำนวน K บิต $Q_0Q_1...Q_{K-1}$ หรืออัตราบิต K บิตต่อสัมประสิทธิ์ ค่าควอนไทซ์สำหรับช่วงของบิตที่ $K-1$ จะฝังอยู่ในช่วงของบิตที่ $K-2$ และในเหตุผลเดียวกัน ก็จะฝังอยู่ในช่วงของบิตที่ $K-3$ เช่นนี้เสมอไป ค่าของตัวควอนไทซ์จะเริ่มจากบิตนัยสำคัญสูง Q_0 ช่วงของบิตนี้จะแบ่งออกเป็น 2 ช่วงย่อย ส่วนของ Q_1 จะแบ่งออกเป็น 4 ช่วงย่อยสำหรับการควอนไทซ์ที่ใช้ค่าไบนารีช่วงย่อยของบิตที่ K จะแบ่งช่วงย่อยเพิ่มขึ้นบิตละสองเท่า หรือบิตที่ K จะแบ่งออกเป็น 2^{K+1} ช่วงย่อย ยกตัวอย่างสำหรับค่านำเข้าที่ตกอยู่ในช่วงย่อยใดจะได้ค่าควอนไทซ์ออกมาตามช่วงนั้น ๆ ของตัวควอนไทซ์ที่อัตรา 3 บิตต่อสัมประสิทธิ์ แสดงดังรูปที่ 2.11 สำหรับการถอดรหัสที่อัตราบิตต่ำก็จะทำการตัดบิตนัยสำคัญถัดไปหรือแทนให้มีค่าเป็นศูนย์เพื่อเป็นการประมาณค่าอย่างหยาบ



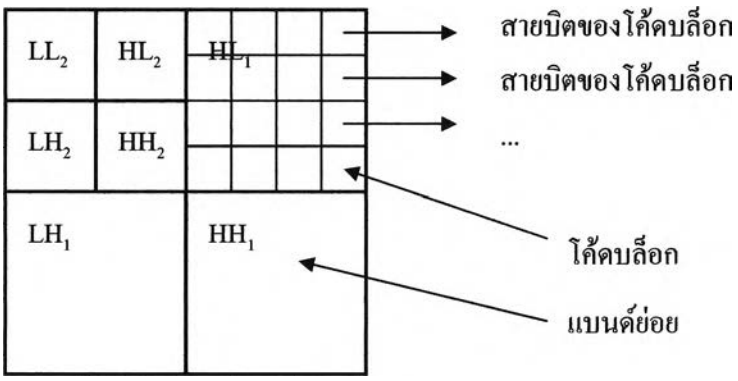
รูปที่ 2.11 ตัวสเกลาร์ควอนไทซ์ฝัง Q_0, Q_1 และ Q_2 ของบิตที่อัตราบิต 1,2 และ 3 บิตต่อสัมประสิทธิ์

ข้อดีอย่างที่สองคือ ส่วนมากแล้วสัมประสิทธิ์เวฟเลตที่ได้จากการแปลงของแบนด์ย่อยความถี่สูงจะมีค่าใกล้เคียงกับศูนย์ ยกเว้นสัมประสิทธิ์ที่เป็นส่วนขอบของวัตถุในภาพ เนื่องจากตัวควอนไทซ์มีเขตไร้ผลสนองกว้าง เมื่อผ่านการควอนไทซ์แล้วสัมประสิทธิ์ส่วนมากของแบนด์ย่อยความถี่สูงจะมีค่าเป็นศูนย์ ส่งผลให้การบีบอัดมีประสิทธิภาพยิ่งขึ้น

2.7.4 โค้ดบล็อกและชั้นคุณภาพ

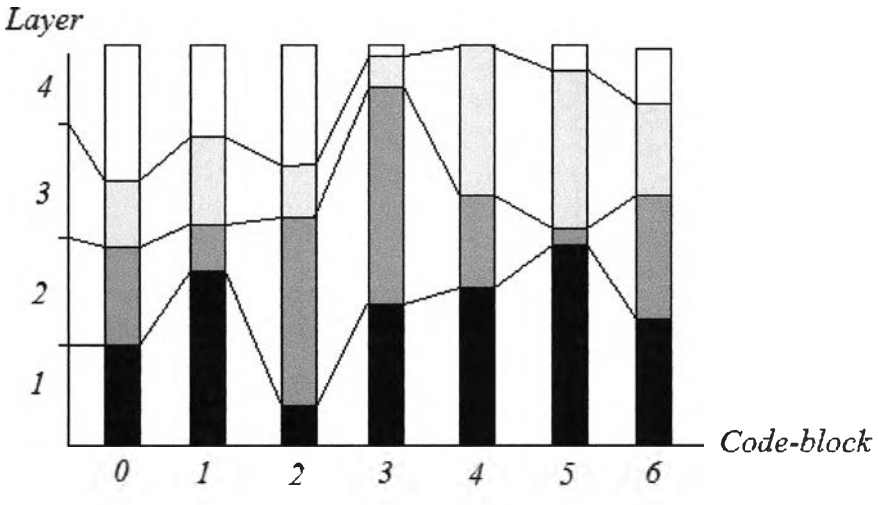
หลังจากขั้นตอนการแปลงเวฟเลตแล้วจะทำการแบ่งแบนด์ย่อยออกเป็นโค้ดบล็อก [10] โดยที่ขนาดของโค้ดบล็อกต้องเป็น $2^{n_1+n_2}$ สัมประสิทธิ์ โดย n_1, n_2 เป็นขนาดของโค้ดบล็อกในแนวนอนและแนวตั้งตามลำดับ โดยที่ $n_1, n_2 \geq 2$ และ $n_1 + n_2 \leq 12$ ยกเว้นโค้ดบล็อกที่อยู่บริเวณขอบของแบนด์ย่อย แต่ทุกโค้ดบล็อกจะต้องมีขนาดไม่เกิน 4096 สัมประสิทธิ์ แต่ละโค้ดบล็อกจะถูกควอนไทซ์และเข้ารหัสเอนโทรปีอย่างอิสระต่อกันดังรูปที่ 2.12 หลังจากผ่านการ

เข้ารหัสเอนโทรปีแล้ว แต่ละโค้ดบล็อกจะถูกเข้ารหัสเป็นสายบิต โดยในการเรียงข้อมูลในสายบิตแบบโพรเกรสซิฟ หมายความว่าในฝั่งของตัวถอดรหัสจะสามารถถอดรหัสแบบโพรเกรสซิฟได้จากชั้นคุณภาพ (Quality Layer) [10]



รูปที่ 2.12 แต่ละโค้ดบล็อกทำการเข้ารหัสเอนโทรปีอย่างอิสระต่อกัน แล้วเรียงข้อมูลเป็นสายบิต และทุกแบนด์ย่อยจะถูกแบ่งเป็นโค้ดบล็อกเหมือนกับแบนด์ย่อย HL_1

สายบิตที่ได้จากแต่ละโค้ดบล็อกยังไม่มีคุณสมบัติแบบโพรเกรสซิฟ จึงต้องมีส่วนที่ทำการตัดสินใจเลือกปริมาณบิตที่เหมาะสมในแต่ละระดับชั้นคุณภาพ (Quality Layer) การจัดสรรบิตจากแต่ละโค้ดบล็อกสำหรับแต่ละชั้นคุณภาพ บริเวณที่อยู่ในพื้นที่สีเดียวกันจะถูกจัดให้เป็นชั้นคุณภาพเดียวกันดังรูปที่ 2.13 เมื่อมีการแบ่งระดับชั้นคุณภาพแล้วตัวถอดรหัสสามารถถอดรหัสจากข้อมูลที่ได้รับเป็นชั้นคุณภาพหรือบางส่วนของชั้นคุณภาพได้



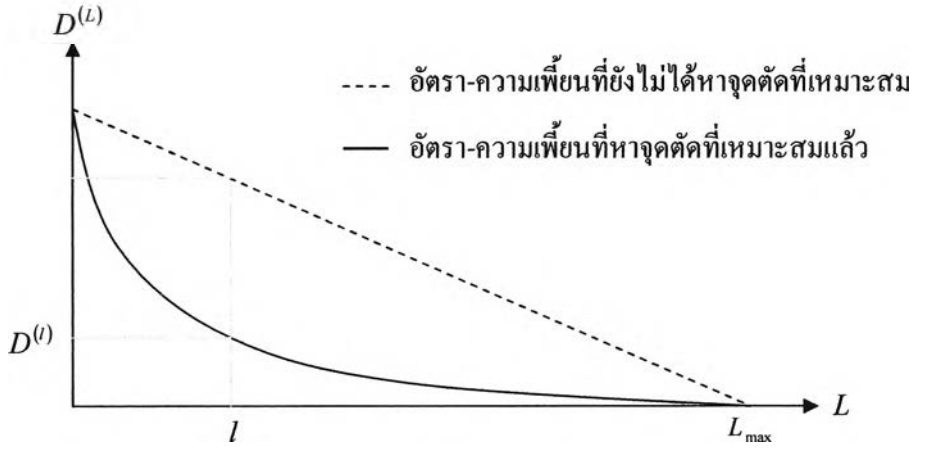
รูปที่ 2.13 ชั้นคุณภาพที่ได้จากการจัดสรรบิตให้กับแต่ละโค้ดบล็อก

โค้ดบล็อกที่มีความสำคัญจะถูกจัดให้มีปริมาณบิตสำหรับเข้ารหัสมากกว่า ดังนั้นจึงเป็นไปได้ที่จะมีโค้ดบล็อกที่ได้ปริมาณบิตสำหรับเข้ารหัสในแต่ละชั้นคุณภาพไม่เท่ากัน ดังเช่นในรูปที่ 2.13 สำหรับชั้นคุณภาพที่ 1 บล็อกที่ 2 มีความสำคัญน้อย ปริมาณบิตที่จัดสรรให้น้อยกว่าบล็อกอื่น และบล็อกที่ 5 มีความสำคัญมากปริมาณบิตที่จัดสรรให้จึงสูงที่สุด ทั้งนี้การตัดสินใจสำหรับการเลือกจุดตัด (Truncation Point) สายบิตของโค้ดบล็อกที่เหมาะสมสามารถหาได้ในขั้นตอนวิธีก่อนการบีบอัดที่เรียกว่าการเข้ารหัสบล็อกฝัง (Embedded Block Coding)

2.7.5 การเข้ารหัสบล็อกฝัง (Embedded Block Coding) [3]

ขั้นตอนการจัดเรียงสายบิตของโค้ดบล็อกใน JPEG2000 โดยรวมแล้วเรียกว่าการเข้ารหัสบล็อกฝัง ซึ่งการเข้ารหัสบล็อกฝังนี้จะทำให้สายบิตมีลักษณะเป็นโพรเกรสซิฟโดยการเลือกจุดตัดของสายบิตที่เหมาะสมดังรูปที่ 2.13 สำหรับจุดตัดที่เหมาะสมหาได้จากเส้นโค้งอัตรา-ความเพี้ยน (Rate-Distortion Curve) โดยอัตราหมายถึงการบีบอัดจะทำที่ระดับอัตราบิตที่กำหนดซึ่งแทนอยู่ในรูปความยาวของสายบิต L และความเพี้ยนจากสายบิตที่มีความยาว l แทนด้วย $D^{(l)}$

กำหนดให้ความยาวของสายบิตทั้งหมดเป็น L^n โดยที่ n เป็นจำนวนของชั้นคุณภาพ ถ้าตัวถอดรหัสทำการถอดรหัสสายบิตที่มีความยาว l ซึ่งมีค่าน้อยกว่าความยาวสายบิตของชั้นคุณภาพทั้งหมด L_{max} แล้ว คุณภาพของภาพสร้างกลับที่ได้จะต้องเกิดความเพี้ยนขึ้น ซึ่งการเข้ารหัสบล็อกฝังจะมีหน้าที่หาจุดตัดที่เหมาะสมเพื่อให้ความเพี้ยนเกิดขึ้นต่ำสุดในความยาวสายบิตที่กำหนดเส้นโค้งอัตรา-ความเพี้ยนแสดงในรูปที่ 2.14



รูปที่ 2.14 เส้นโค้งอัตรา-ความเพี้ยน

เส้นโค้งอัตรา-ความเพี้ยนที่ได้จากการหาจุดตัดที่เหมาะสมจะทำให้ความเพี้ยนลดลงกว่าก่อนยังไม่หาจุดตัดที่เหมาะสม ซึ่งจุดตัดที่เหมาะสมสามารถคำนวณได้

กำหนดให้ L^n คือผลรวมความยาวของสายบิตถึงจุดตัดที่ n และ L_n คือความยาวสายบิตที่จุดตัดที่ n โดยสอดคล้องตามสมการที่ (2.18)

$$L^n = \sum_n L_n \le L_{\max} \tag{2.18}$$

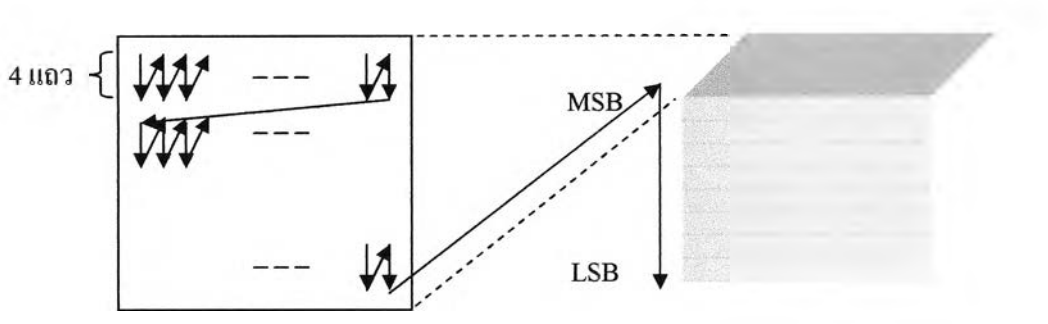
ความเพี้ยนจะเป็นปริมาณที่มีค่าไปในทางตรงกันข้ามกับความยาวของสายบิต กล่าวคือเมื่อสายบิตที่จะทำการถอดรหัสมากขึ้นความเพี้ยนจะลดลง ที่จุดตัดที่ n ผลรวมความเพี้ยนแทนด้วย D^n สามารถหาได้จากสมการที่ (2.19) และ D_n แทนความเพี้ยนของจุดตัดที่ n

$$D^n = \sum_n D_n \tag{2.19}$$

จุดตัดที่เหมาะสมที่จะทำให้ค่าความเพี้ยนลดลงมากที่สุดคือจุดที่ทำให้ความชันระหว่างจุดตัดเดิมและจุดตัดใหม่มีค่าน้อยที่สุด ซึ่งในการหาจุดตัดที่เหมาะสมนั้นความยาวของสายบิต L นั้นจะต้องเป็นความยาวที่ผ่านการเข้ารหัสเอนโทรปีแล้ว [3]

2.7.6 การเข้ารหัสระนาบบิต

สัมประสิทธิ์ในโค้ดบล็อกทั้งหมดทำการเขียนให้อยู่ในรูปของตัวเลขไบนารี โดยนำแต่ละบิตนัยสำคัญที่เท่ากันเขียนอยู่ในระนาบเดียวกัน มาตรฐาน JPEG2000 จะเริ่มการเข้ารหัสสัมประสิทธิ์ทีละระนาบ เริ่มจากสัมประสิทธิ์ของระนาบบิตนัยสำคัญมากสุด (Most Significant Bit-plane, MSB) ทุกตัว แล้วจึงไปเข้ารหัสระนาบบิตนัยสำคัญรองลงไปทีละระนาบจนกระทั่งเข้ารหัสถึงระนาบบิตนัยสำคัญน้อยสุด (Least Significant Bitplane, LSB) การเข้ารหัสในแต่ละระนาบจะกวาดตรวจ (Scan) แบบแรสเตอร์ (Raster) ทีละแถบ แถบละ 4 แถว การกวาดตรวจในแต่ละแถบจะกวาดตรวจในแนวหลักก่อน เมื่อกวาดตรวจครบ 4 สัมประสิทธิ์ในแนวหลักก็จะไปเริ่มที่หลักถัดไป จนกระทั่งครบ 1 แถบ การกวาดตรวจจะไปเริ่มต้นที่แถบถัดไปทีละหลักเช่นกัน เมื่อการกวาดตรวจครบทั้งระนาบแล้ว การกวาดตรวจจะไปเริ่มที่ระนาบบิตนัยสำคัญรองลงไป 1 ระนาบ โดยลำดับจะเหมือนกับในระนาบบิตนัยสำคัญมากสุดดังรูปที่ 2.15 [3]



รูปที่ 2.15 การกวาดตรวจแบบแรสเตอร์บนระนาบบิต

2.7.7 เงื่อนไขในการเข้ารหัสเอนโทรปี

กระบวนการสุ่มมาร์คอฟ (Markov Random Processes) เป็นรูปแบบที่ใช้แทนความน่าจะเป็นของข่าวสารแบบมีเงื่อนไขอย่างมีประสิทธิภาพสำหรับแหล่งกำเนิดทั่ว ๆ ไป ซึ่งรวมไปถึงแหล่งกำเนิดของสัญญาณภาพ การเข้ารหัสเลขคณิตที่สามารถลดเอนโทรปีของข่าวสารให้ลู่เข้าสู่ขอบเขตเอนโทรปีตามทฤษฎีของ Shannon ขึ้นกับเงื่อนไขของความน่าจะเป็นที่เหมาะสมใช้ในการเข้ารหัสด้วย สำหรับการประมวลผลสัญญาณภาพ สัญญาณ 2 มิติ ถูกกวาดตรวจเหลือเพียง 1 มิติ รูปแบบที่เหมาะสมเพื่อการเข้ารหัส การกำหนดพารามิเตอร์ที่ใช้เป็นบริบทสำหรับการเข้ารหัสเอนโทรปีจะกล่าวถึงต่อไป

การเข้ารหัสเอนโทรปีจะเข้ารหัสแต่ละโค้ดบล็อกอย่างอิสระต่อกัน พารามิเตอร์ที่ใช้กำหนดให้อ้างถึงในโค้ดบล็อกสำหรับการเข้ารหัสเอนโทรปี ประกอบด้วยพารามิเตอร์ขนาดของโค้ดบล็อกที่มี จำนวนแถว และจำนวนหลัก n_1, n_2 ตามลำดับ สัมประสิทธิ์ที่ถูกควอนไทซ์แทนด้วย $q[n_1, n_2]$ บิตเครื่องหมายแทนด้วย $\chi[n_1, n_2]$ และค่าแอมพลิจูดของสัมประสิทธิ์ที่ถูกควอนไทซ์แทนด้วย $v[n_1, n_2] = |q[n_1, n_2]|$ โดยที่ $v^p[n_1, n_2]$ แทนบิตที่ p ของเลขไบนารีสำหรับสัมประสิทธิ์ $q[n_1, n_2]$ เริ่มนับจาก $p = 0$ แทนบิตนัยสำคัญน้อยสุด เมื่อค่า p เพิ่มขึ้นนัยสำคัญของบิตก็จะเริ่มมากขึ้น พารามิเตอร์ถัดไปคือสถานะเครื่องหมาย $\sigma[n_1, n_2]$ พารามิเตอร์นี้จะมีค่าเริ่มต้นในการเข้ารหัสสัมประสิทธิ์เป็น 0 และจะเปลี่ยนค่าเป็น 1 หลังจากที่บิตที่ไม่เป็น 0 บิตแรกถูกกวาดตรวจพบ เมื่อพารามิเตอร์สถานะเครื่องหมายเปลี่ยนค่าเป็น 1 ก็จะไม่มีการเปลี่ยนกลับจนกระทั่งการเข้ารหัสเสร็จสิ้นในระนาบบิตสุดท้ายของโค้ดบล็อก

ในขั้นตอนการเข้ารหัสจะแบ่งออกเป็น 3 สถานะตามสถานะเครื่องหมาย ในกรณีที่ $\sigma[n_1, n_2] = 0$ แล้ว หรือสัมประสิทธิ์ข้างเคียงค่าใดค่าหนึ่งเข้ารหัสเครื่องหมายไปแล้ว บิตนั้นจะ

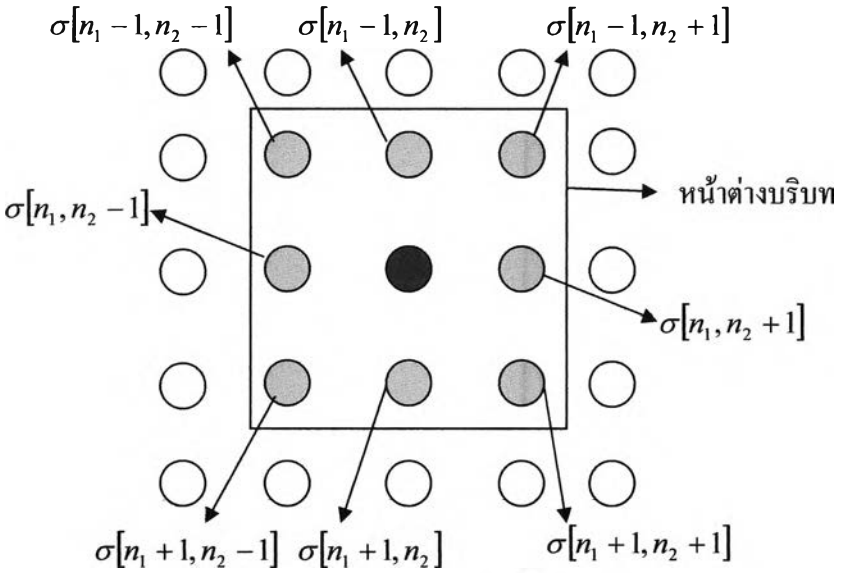
เข้ารหัสในสถานะเครื่องหมาย ถ้าสัมประสิทธิ์อยู่ในสถานะที่เข้ารหัสเครื่องหมายไปแล้วบิตถัดไปของสัมประสิทธิ์นั้นจะเข้ารหัสในสถานะแมคนิจูดรีไฟน์เมนต์ (Magnitude Refinement)

2.7.8 การแพร่ของเครื่องหมาย (Significance Propagation)

การเข้ารหัสสำหรับสัมประสิทธิ์ในโค้ดบล็อกจะเริ่มต้นจากเข้ารหัสในสถานะเครื่องหมาย ซึ่งการเข้ารหัสในสถานะนี้แบ่งออกได้เป็น 2 แบบ แบบแรกคือนอร์แมลโหมด (Normal Mode) และแบบที่สองคือรันโหมด (Run Mode)

2.7.8.1 นอร์แมลโหมด (Normal mode)

ในโหมดนี้จะแบ่งบริบท (Context) ออกเป็น 9 แบบ ซึ่งจะพิจารณาสัมประสิทธิ์ข้างเคียงที่เข้ารหัสในสถานะเครื่องหมายไปแล้วดังรูปที่ 2.16 สัมประสิทธิ์ที่กวาดตรวจถึงอยู่ตำแหน่งตรงกลางของหน้าต่างบริบท (Context Window) ผลรวมของสถานะเครื่องหมายจะแทนด้วยฉลากบริบท (Context Label) $\kappa^{Sig}[n_1, n_2]$ ซึ่งคำนวณได้จากสมการที่ (2.20) โดย κ^h เป็นผลรวมของสถานะเครื่องหมายในแนวนอนของหน้าต่างบริบท κ^v เป็นผลรวมของสถานะเครื่องหมายในแนวตั้งของหน้าต่างบริบท และ κ^d เป็นผลรวมของสถานะเครื่องหมายในแนวทแยงมุมของหน้าต่างบริบท



รูปที่ 2.16 บริบทพิจารณาจาก 8 สัมประสิทธิ์ข้างเคียงในหน้าต่างบริบท

$$\kappa^h[n_1,n_2]=\sigma[n_1,n_2-1]+\sigma[n_1,n_2+1]$$
$$\kappa^v[n_1,n_2]=\sigma[n-1_1,n_2]+\sigma[n+1_1,n_2] \quad (2.20)$$
$$\kappa^d[n_1,n_2]=\sum_{k_1=\pm1}\sum_{k_2=\pm2}\sigma[n_1+k_1,n_2+k_2]$$

โค้ดบล็อกในแต่ละแบนด์ย่อยมีบริบทที่ไม่เหมือนกัน เช่นในแบนด์ย่อย HL ที่ผ่านตัวกรองความถี่สูงแนวนอนมีขอบของวัตถุในแนวนอนปรากฏชัด ในแบนด์ย่อย LH ที่ผ่านตัวกรองความถี่สูงแนวตั้งมีขอบของวัตถุในแนวตั้งปรากฏชัด และแบนด์ย่อย HH ที่ผ่านตัวกรองความถี่สูงทั้งแนวตั้งและแนวนอนมีขอบของวัตถุในแนวทแยงปรากฏชัด ในการกวาดตรวจพบว่าสัมประสิทธิ์ที่เป็นขอบของวัตถุจะพบก่อน แล้วสัมประสิทธิ์ของส่วนที่เป็นขอบจะเข้ารหัสอยู่ในสถานะเครื่องหมายก่อนสัมประสิทธิ์อื่น ค่าสถานะเครื่องหมายที่ได้จะนำมาคำนวณเป็นฉลากบริบทอันมีได้ทั้งหมด 9 แบบ คือ 0 ถึง 8 ดังตารางที่ 2.4

ตารางที่ 2.4 การแบ่งรูปแบบของบริบททั้ง 9 แบบในแบนด์ย่อยต่างกัน

$\kappa^{sig}[n_1,n_2]$	แบนด์ย่อย LL และ LH			แบนด์ย่อย HL			แบนด์ย่อย HH	
	κ^h	κ^v	κ^d	κ^h	κ^v	κ^d	κ^d	$\kappa^h+\kappa^v$
8	2	x	x	x	2	x	≥ 3	x
7	1	≥ 1	x	≥ 1	1	x	2	≥ 1
6	1	0	≥ 1	0	1	≥ 1	2	0
5	1	0	0	0	1	0	1	≥ 2
4	0	2	x	2	0	x	1	1
3	0	1	x	1	0	x	1	0
2	0	0	≥ 2	0	0	≥ 2	0	≥ 2
1	0	0	1	0	0	1	0	1
0	0	0	0	0	0	0	0	0

x หมายถึงค่าจะเป็นค่าใดก็ได้ หรือไม่กำหนดค่าตายตัว

2.7.8.2 รันโหมด (Run-mode)

ที่ระดับอัตราการบีบอัดสูง สัมประสิทธิ์ส่วนใหญ่ในแบนด์ย่อยจะมีค่าเป็น 0 หรือไม่ได้อยู่ในสถานะเครื่องหมายเกือบทุกระนาบบิต แม้ว่าสัมประสิทธิ์ที่มีค่าจะถูกเข้ารหัสให้อยู่ในสถานะเครื่องหมายแต่ก็จะเข้ารหัสให้อยู่ในสถานะเครื่องหมายในระนาบบิตที่มีนัยสำคัญต่ำมาก เมื่อมีจำนวนสัมประสิทธิ์ที่ไม่ได้อยู่ในสถานะเครื่องหมายมาก รันโหมดจะถูกนำมาใช้เมื่อสอดคล้องกับเงื่อนไขทั้ง 3 ข้อ [3]

1. สัมประสิทธิ์ทั้ง 4 ตัวใน 1 หลักต้องไม่อยู่ในสถานะเครื่องหมายสำหรับระนาบบิตนั้น $\sigma[n_1 + r, n_2]$ เมื่อ $0 \le r < 4$

2. สัมประสิทธิ์ทั้ง 4 ตัวใน 1 หลักต้องมีสัมประสิทธิ์ข้างเคียงที่ไม่อยู่ในสถานะเครื่องหมายสำหรับระนาบบิตนั้น $\kappa^{sig}[n_1 + r, n_2] = 0$ เมื่อ $0 \le r < 4$

3. สัมประสิทธิ์ทั้ง 4 ตัวจะต้องอยู่ในแถบการกวาดตรวจเดียวกัน

จำนวนสัมประสิทธิ์ที่ไม่อยู่ในสถานะเครื่องหมายนี้จะใช้ไบนารี 2 บิตในการเข้ารหัสในจุดเริ่มต้นของระนาบบิตนัยสำคัญมากสุด เมื่อพบสัมประสิทธิ์ที่อยู่ในสภาวะเครื่องหมายแล้วก็จะบอกจุดสุดท้ายด้วยสัญลักษณ์รันอินเตอร์รัปชั่น (Run Interruption) หลังจากนั้นสัมประสิทธิ์ที่เหลือจะถูกเข้ารหัสในนอร์แมลโหมดจนกระทั่งพบเงื่อนไขทั้ง 3 ข้อในรันโหมด สัมประสิทธิ์ที่มีการเปลี่ยนสถานะเครื่องหมายในระนาบบิตนั้นจะเข้ารหัสในพาสการแพร่ของเครื่องหมาย (Significance Propagation Pass)

2.7.9 แมคนิจูดรีไฟน์เมนต์พาส (Magnitude Refinement Pass)

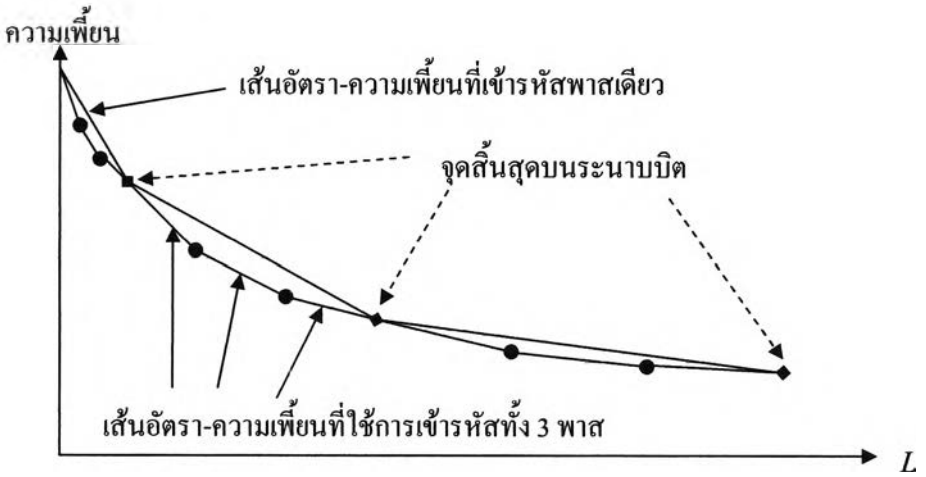
เมื่อสัมประสิทธิ์อยู่ในสถานะเครื่องหมายแล้ว $\sigma[n_1, n_2] = 1$ ในระนาบบิตถัดไปจะทำการเข้ารหัสสัมประสิทธิ์นี้เป็นแมคนิจูดรีไฟน์เมนต์ ซึ่งบริบทที่ใช้ในการเข้ารหัสแมคนิจูดรีไฟน์เมนต์ใช้เงื่อนไขความน่าจะเป็นแบบมาร์คอฟจากตัวควอนไทซ์มีเขตไร้ผลสนอง สัมประสิทธิ์ที่มีสถานะเครื่องหมายแล้วจะเข้ารหัสในแมคนิจูดรีไฟน์เมนต์พาส

2.7.10 คลีนอัพพาส (Cleanup Pass)

สำหรับสัมประสิทธิ์ที่ไม่ได้เข้ารหัสทั้งในสถานะเครื่องหมายและแมคนิจูดรีไฟน์เมนต์จะเข้ารหัสในคลีนอัพพาส หรือสัมประสิทธิ์ที่ยังไม่ได้อยู่ในสถานะเครื่องหมายและไม่มีสัมประสิทธิ์ข้างเคียงที่ยังไม่ได้อยู่ในสถานะเครื่องหมาย

2.7.11 ระนาบบิตที่แตกออก (Fractional Bit-planes)

สายบิตของโค้ดบล็อกมีจุดตัดที่เหมาะสมซึ่งจะพิจารณาจาก เส้นโค้งอัตรา-ความเพี้ยน ในแต่ละระนาบบิตจะมีจุดตัดที่เหมาะสมอยู่หลายจุดได้มาจากการเข้ารหัสสัญลักษณ์ (Symbol) เป็น 3 พาส หมายความว่าในหนึ่งระนาบบิตการกวาดตรวจ 1 รอบจะทำการเข้ารหัสแยกออกเป็น 3 พาส ซึ่งบิตที่ได้จากการเข้ารหัสในแต่ละพาสนั้นส่งผลต่อความเพี้ยนไม่เท่ากัน สำหรับพาสการแพร่ของเครื่องหมายถือว่ามีความสำคัญมากเนื่องจากเครื่องหมายบวก หรือลบ ของสัมประสิทธิ์หากตัวถอดรหัสตีความผิดจะมีความเพี้ยนเกิดขึ้นสูง แต่ในขณะเดียวกันจำนวนเครื่องหมาย (Sign) ของสัมประสิทธิ์ก็มีจำนวนน้อยด้วยดังนั้น พาสการแพร่ของเครื่องหมายถือเป็นพาสที่มีความสำคัญที่สุด ถัดมาแมคนิจูดรีไฟน์เมนต์พาสบิตที่อยู่ในพาสนี้จะมีความสำคัญตามนัยสำคัญของระนาบบิต และสุดท้ายคลีนอัพพาสจะเป็นพาสที่ถูกเข้ารหัสแล้วมีประสิทธิภาพน้อยกว่าพาสอื่น [3] เส้นโค้งอัตรา-ความเพี้ยนของจุดตัดสำหรับแต่ละพาสแสดงดังรูปที่ 2.17 เมื่อการเข้ารหัสใช้ทั้ง 3 พาสแล้วจุดตัดที่เหมาะสมก็จะได้ความชันที่มีค่าน้อยกว่าการเข้ารหัสเพียงพาสเดียว

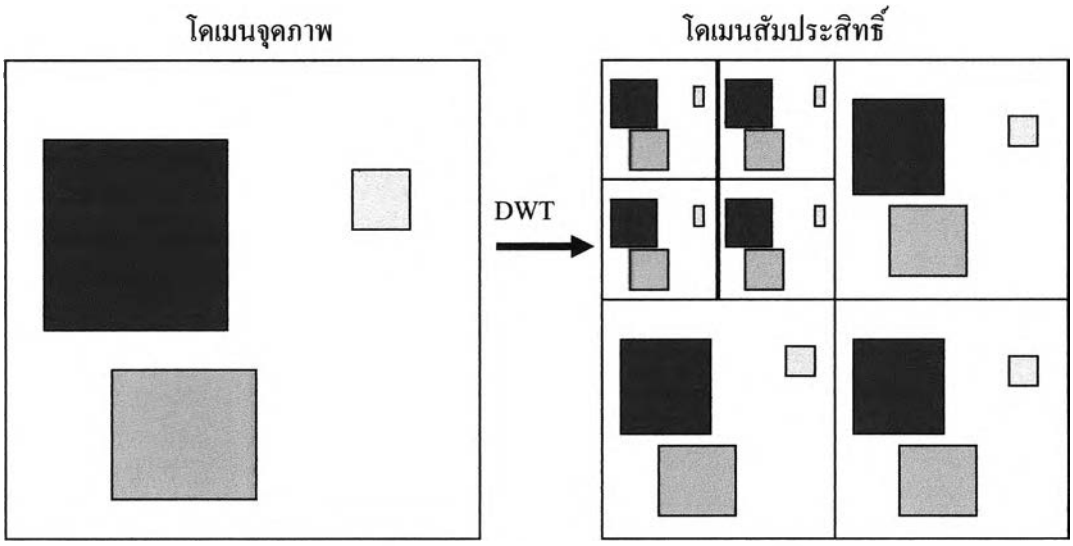


รูปที่ 2.17 เส้นโค้งอัตรา-ความเพี้ยนของสายบิตจากการเข้ารหัสระนาบบิตเป็น 1 พาส และ 3 พาส

2.8 การเข้ารหัสเน้นบริเวณที่สนใจ

ลักษณะประการหนึ่งของมาตรฐาน JPEG2000 ทำให้ตัวถอดรหัสสามารถถอดรหัสได้บริเวณที่ตัวเข้ารหัสกำหนดให้มีความสำคัญมากกว่าได้ก่อน สำหรับในกรณีที่ข้อมูลมีการส่งในอัตราบิตต่ำซึ่งใช้ได้ดีกับภาพที่มีความละเอียดสูงแต่ต้องการบางส่วนที่ภาคส่งอยากให้ภาครับสามารถถอดรหัสได้ก่อน ในการเข้ารหัสบริเวณที่สนใจของมาตรฐาน JPEG2000 การแปลงเวฟเลตมีส่วนสำคัญที่ช่วยให้สามารถเข้ารหัสแบบเน้นบริเวณได้ เนื่องจากภาพที่จะทำการเข้ารหัสนั้นเมื่อผ่านการแปลงเวฟเลตแล้ว ทำให้จุดภาพของภาพแต่ละจุดภาพที่อยู่ใกล้เคียงกันมีสหสัมพันธ์

ลดลงสัมประสิทธิ์ในบริเวณนั้นเองก็จะมีความเป็นอิสระต่อกันมากขึ้น ดังนั้นสัมประสิทธิ์ในบริเวณนั้นเองจึงสามารถทำการประกอบกลับมาเป็นจุดภาพของภาพบริเวณนั้นได้ ดังรูปที่ 2.18 โดเมนจุดภาพที่มีสีเดียวกันเทียบได้กับโดเมนสัมประสิทธิ์ที่มีสีเดียวกัน

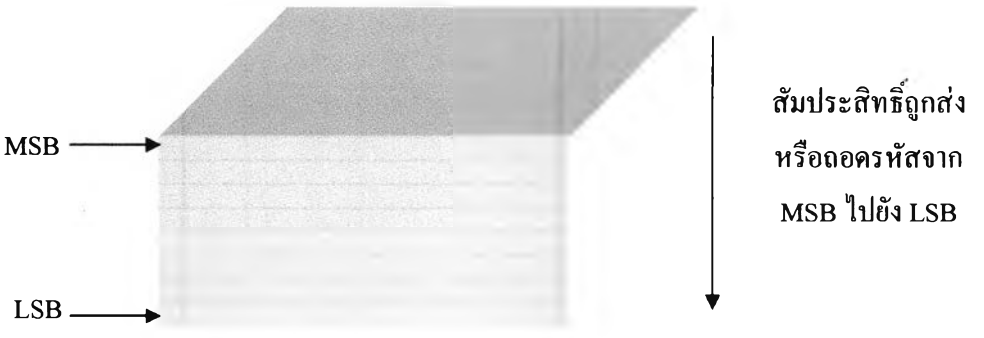


รูปที่ 2.18 ความสัมพันธ์เชิงพื้นที่ของจุดภาพกับสัมประสิทธิ์เวฟเลต

ในการเข้ารหัสเน้นบริเวณที่สนใจ ผู้ใช้ต้องกำหนดบริเวณบนภาพที่ต้องการเข้ารหัสให้ได้คุณภาพสูงกว่าบริเวณอื่น การเข้ารหัสเน้นบริเวณที่สนใจของมาตรฐาน JPEG2000 ใช้วิธีลดความสำคัญของสัมประสิทธิ์เวฟเลตที่ไม่ได้อยู่ในบริเวณที่สนใจหรือเรียกบริเวณที่อยู่นอกบริเวณที่สนใจว่าพื้นหลัง การลดความสำคัญของสัมประสิทธิ์ในพื้นหลังทำได้โดยการเลื่อนระนาบบิตให้อยู่ในระนาบบิตนัยสำคัญต่ำลง เนื่องจากการเข้ารหัสระนาบบิตจะกระทำในโค้ดบล็อกอย่างอิสระต่อกัน ถ้ามีการกำหนดบริเวณที่สนใจแล้ว ตัวเข้ารหัสจะเพิ่มกระบวนการเลื่อนระนาบบิตในโค้ดบล็อกก่อนที่จะทำการเข้ารหัสระนาบบิต

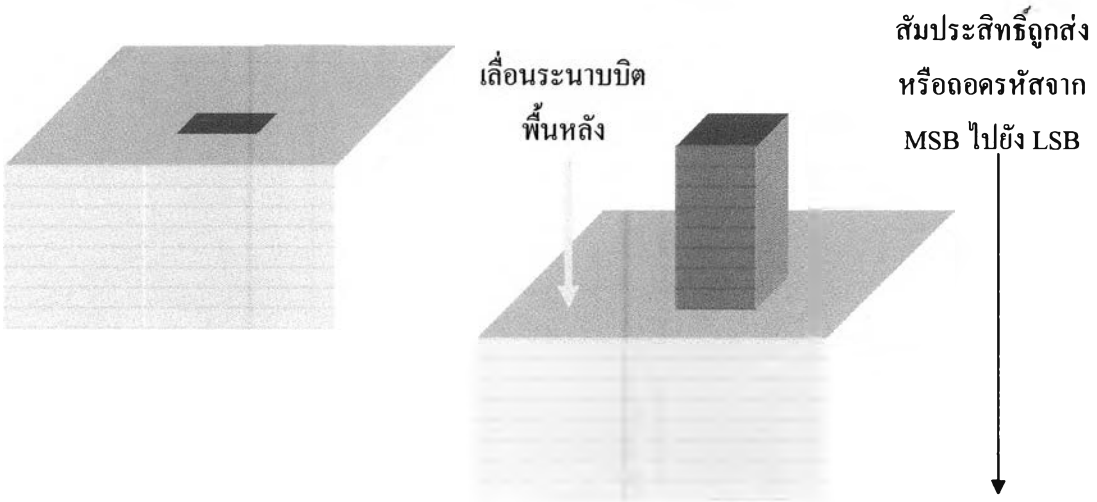
2.8.1 หลักการเน้นบริเวณที่สนใจ

ลำดับการกวาดตรวจสัมประสิทธิ์ในโค้ดบล็อกนั้นเรียงจากระนาบบิตนัยสำคัญมากสุดไปยังระนาบบิตที่มีนัยสำคัญน้อยสุด ดังรูปที่ 2.19



รูปที่ 2.19 ระนาบบิตในโค้ดบล็อก

มาตรฐานได้กำหนดวิธีการเน้นบริเวณที่สนใจอย่างตรง ๆ คือการส่งสัมประสิทธิ์ของบริเวณที่สนใจไปก่อน โดยการลดระดับบิตของสัมประสิทธิ์พื้นหลังดังรูปที่ 2.20



รูปที่ 2.20 การเลื่อนระนาบบิตของสัมประสิทธิ์พื้นหลัง

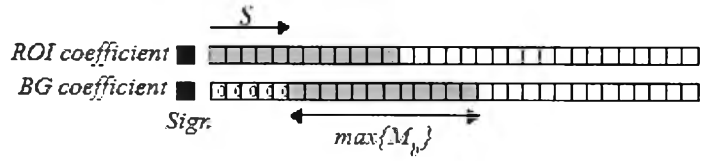
ที่ตัวถอดรหัสเมื่อทำการถอดรหัสสายบิตแล้วพบจุดตัดหรือกระบวนการถอดรหัสถูกหยุดก่อนที่จะทำการถอดรหัสเสร็จสมบูรณ์ ภาพสร้างกลับที่ได้จะมีคุณภาพในบริเวณที่สนใจดีกว่าพื้นหลัง เนื่องจากบริเวณที่สนใจจะถูกสร้างกลับด้วยจำนวนระนาบบิตมากกว่าพื้นหลัง แต่ถ้าสายบิตทั้งหมดถูกถอดรหัสแล้ว ภาพที่สร้างกลับก็จะได้คุณภาพสูงสุดตามแต่ปริมาณบิตที่กำหนดในตอนเข้ารหัส

มาตรฐาน JPEG2000 กำหนดระเบียบวิธีการเลื่อนระนาบบิตไว้อยู่ 2 วิธี เจเนรัลชิฟท์ (General Shift) และแมกซ์ชิฟท์ (MaxShift) ในกระบวนการเลื่อนระนาบบิตของทั้งสองวิธีนี้ ระนาบบิตจะถูกเลื่อนไปตามค่าชิฟท์ (Shift Value) s ซึ่งมีค่าเป็นจำนวนเต็มบวก $s \in Z^+$

2.8.2 เจเนรัล (General Shift) [3]

ระเบียบวิธีเจเนรัลชิฟท์ระนาบบิตส่วนพื้นหลังถูกเลื่อนไป s ระนาบตามสมการที่ (2.21) วิธีเจเนรัลชิฟท์สัมประสิทธิ์ของพื้นหลังจะเลื่อนไปไม่เกินระนาบบิตสูงสุดในโค้ดบล็อก ระนาบบิตสูงสุดในโค้ดบล็อกหมายถึงจำนวนบิตที่ใช้แทนค่าไบนารีของสัมประสิทธิ์ที่มีขนาดของสัมประสิทธิ์สูงที่สุดใช้สัญลักษณ์ $\max\{M_b\}$ หรือ K_max ระนาบบิตจะเลื่อนไปดังรูปที่ 2.21

$$s = \{s \in Z^+ \mid s < M_b\} \tag{2.21}$$

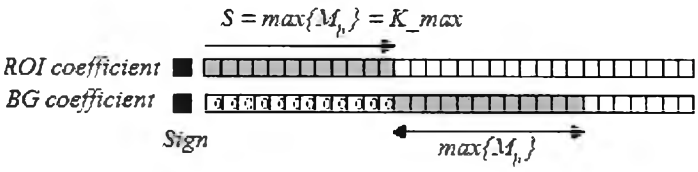


รูปที่ 2.21 ระเบียบวิธีเจเนรัลชิฟท์

2.8.3 แมกซ์ชิฟท์ (Maxshift) [3]

ระเบียบวิธีแมกซ์ชิฟท์ จะพิจารณาค่าสูงสุดของสัมประสิทธิ์ว่าค่าใดเป็นค่าสูงที่สุด และจากนั้นจะทำการหาจำนวนของระนาบบิตที่ใช้แทนค่าสัมประสิทธิ์นั้นว่าต้องใช้จำนวนเท่ากับ M_b เมื่อได้ค่าจำนวนระนาบบิตสูงสุดมาแล้วจะทำการเลื่อนระนาบบิตของสัมประสิทธิ์พื้นหลังไป $\max\{M_b\}$ ค่าชิฟท์ที่ใช้เป็นตามสมการที่ (2.22) ระนาบบิตจะเลื่อนไปดังรูปที่ 2.22

$$s = \{s \in Z^+ \mid s = \max\{M_b\}\} \tag{2.22}$$



รูปที่ 2.22 ระเบียบวิธีแมกซ์ชิฟท์

กระบวนการเลื่อนระนาบบิตค่าของสัมประสิทธิ์พื้นหลังค่าสัมประสิทธิ์จะเปลี่ยนแปลงไป การเลื่อนระนาบบิตลง 1 ระนาบจะทำให้ค่าของสัมประสิทธิ์นั้นลดลงเหลือครึ่งหนึ่ง การเลื่อนระนาบบิตไป s ระนาบจะทำให้ค่าของสัมประสิทธิ์ถูกหารด้วยค่า 2^s แต่ค่าสัมประสิทธิ์ที่ถูกเปลี่ยนไปทำให้กระบวนการสร้างกลับในตัวถอดรหัสไม่สามารถทำกลับได้เป็นภาพที่ถูกต้อง สำหรับวิธีเจเนรัลชิฟท์ ตัวเข้ารหัสจะต้องส่งรูปร่างของบริเวณที่สนใจไปด้วย เพื่อให้ตัวถอดรหัสทราบว่าสัมประสิทธิ์ตัวที่ได้รับมามีสัมประสิทธิ์ตัวใดบ้างที่จะต้องทำการเลื่อนระนาบบิตกลับเท่ากับ s ระนาบ หรือถ้าไม่ส่งรูปร่างบริเวณที่สนใจ ตัวเข้ารหัสจะส่งพารามิเตอร์บอกบริเวณบริเวณที่สนใจ ซึ่งกำหนดเป็นรูปร่างอย่างง่าย ๆ ไว้ 2 แบบคือสี่เหลี่ยม และวงรี

สำหรับวิธีแมกซ์ชิฟท์ตัวถอดรหัสจะทราบว่าสัมประสิทธิ์ที่ได้เป็นสัมประสิทธิ์ของบริเวณที่สนใจ โดยการพิจารณาค่าสัมประสิทธิ์ที่รับมา ถ้าสัมประสิทธิ์ที่รับมามีบิตที่ไม่เป็น 0 อยู่ในระนาบบิตนัยสำคัญสูง s ระนาบ ตัวถอดรหัสจะรู้ได้ว่าสัมประสิทธิ์นี้เป็นสัมประสิทธิ์ของบริเวณที่สนใจ และทำการเลื่อนระนาบบิตของสัมประสิทธิ์พื้นหลังกลับเท่ากับค่าชิฟท์ s

2.9 เอกสารและงานวิจัยที่เกี่ยวข้อง

ในวิทยานิพนธ์นี้อ้างถึงระเบียบวิธีที่เกี่ยวข้องกับการเข้ารหัส JPEG2000 เน้นบริเวณที่สนใจอีก 3 วิธี ซึ่งเป็นการปรับปรุงความสามารถในการเข้ารหัสเน้นบริเวณที่สนใจอย่างต่อเนื่องกันเรื่อยมา

2.9.1 การเลื่อนระนาบบิตระนาบต่อระนาบ (Bit-plane by Bit-plane Shift, BbBShift) [12]

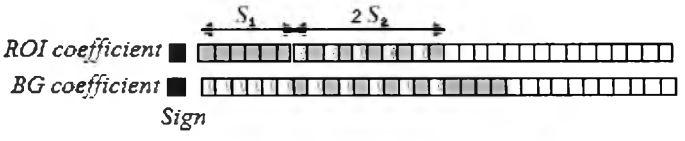
เสนอขึ้นโดย Zhou Wang และ Alan C. Bovik ซึ่งแสดงให้เห็นว่าระเบียบแมกซ์ชิฟท์ของมาตรฐานสามารถเน้นคุณภาพของบริเวณที่สนใจอย่างมาก แต่ภาพในส่วนพื้นหลังจะถูกลดระดับความสนใจลงจนกระทั่งภาพสัมประสิทธิ์ในส่วนบริเวณที่สนใจได้รับค่าจนกระทั่งครบทุกระนาบบิต ภาพในส่วนพื้นหลังจึงจะเริ่มมีการเพิ่มขึ้นของคุณภาพ ซึ่ง Zhou Wang เสนอว่าถ้าทำการเลื่อนระนาบบิตของบริเวณที่สนใจสลับกับระนาบบิตของพื้นหลังแล้ว คุณภาพของพื้นหลังกับของบริเวณที่สนใจจะเพิ่มขึ้นอย่างใกล้เคียงกันด้วย การเลื่อนระนาบบิตที่ Zhou Wang นำเสนอจะเป็นดังรูปที่ 2.23 การเลื่อนระนาบบิตแบบ BbBShift จะต้องส่งพารามิเตอร์ 2 ตัวให้กับตัวถอดรหัส คือ s_1 และ s_2 ซึ่ง $s_1, s_2 = \{s_1, s_2 \in Z^+ \mid s_1 + s_2 = \max\{M_b\}\}$ ระนาบบิตจะเลื่อนตามเงื่อนไขดังนี้หมายเหตุระนาบบิตที่ b หมายถึงระนาบบิตที่มีนัยสำคัญสูงอันดับที่ b แต่ตัวอักษรพร้อมตัวห้อย M_b หมายถึงแมคนิจูดของสัมประสิทธิ์ที่มี b ระนาบบิต

สัมประสิทธิ์ในบริเวณที่สนใจจะทำการเลื่อนระนาบบิตไปตามสมการที่ (2.23)

$$\begin{aligned} b \le s_1 &\Rightarrow b = b \\ s_1 < b < s_1 + s_2 &\Rightarrow b = s_1 + 2(b - s_1) \end{aligned} \tag{2.23}$$

สัมประสิทธิ์ในบริเวณพื้นหลังจะทำการเลื่อนระนาบบิตไปตามสมการที่ (2.24)

$$\begin{aligned} b < s_2 &\Rightarrow b = s_1 + 2b - 1 \\ b \ge s_2 &\Rightarrow b = s_1 + s_2 + b \end{aligned} \tag{2.24}$$



รูปที่ 2.23 การเลื่อนระนาบบิตระนาบต่อระนาบ

2.9.2 การเลื่อนระนาบบิตระนาบต่อระนาบแบบนัยทั่วไป (Generalized Bit-plane by Bit-plane Shift, GBbBShift) [13]

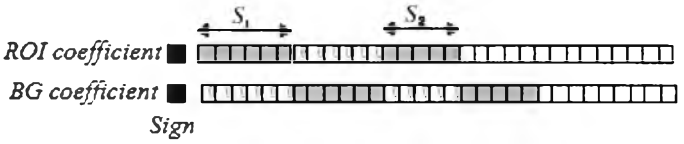
วิธีนี้ยังเป็นของ Zhou Wang และ Alan C. Bovik เช่นกันซึ่งได้กล่าวถึงในระเบียบวิธีเดิมของพวกเขาว่า เมื่อระนาบบิตของบริเวณที่สนใจได้รับมาส่วนหนึ่งแล้วจากนั้นคุณภาพส่วนพื้นหลังจะเพิ่มขึ้นในอัตราเดียวกันกับคุณภาพของบริเวณที่สนใจ แต่เนื่องจากคุณภาพของบริเวณที่สนใจนั้นจะสูงอยู่ในระดับหนึ่งเพราะได้รับระนาบบิตนัยสำคัญสูงมาก่อน เมื่อได้รับระนาบบิตที่มีนัยสำคัญน้อยกว่าคุณภาพของบริเวณที่สนใจจึงไม่จำเป็นต้องเพิ่มขึ้นอีกมากนักจึงทำการปรับปรุงระเบียบวิธี BbBShift ซึ่งหลังจากเน้นบริเวณที่สนใจส่วนหนึ่งแล้วก็จะทำการเน้นพื้นหลังแทนที่การให้ความสำคัญของพื้นหลังเทียบเท่ากับบริเวณที่สนใจ การเลื่อนระนาบบิตแบบ GBbBShift เป็นดังรูปที่ 2.24 การเลื่อนระนาบบิตนี้จะต้องส่งค่าพารามิเตอร์ s_1 และ s_2 ให้กับตัวถอดรหัสด้วยเพื่อให้ตัวถอดรหัสทราบว่าระนาบบิตใดเป็นระนาบบิตของบริเวณที่สนใจ หรือเป็นระนาบบิตของพื้นหลังและการเลื่อนระนาบบิตจะทำตามสมการที่ (2.25) และ (2.26)

สัมประสิทธิ์ในบริเวณที่สนใจ

$$\begin{aligned} b \le s_1 &\Rightarrow b = b \\ s_1 < b \le s_1 + s_2 &\Rightarrow b = b + s_1 \end{aligned} \tag{2.25}$$

สัมประสิทธิ์ในบริเวณพื้นหลัง

$$\begin{aligned} b < s_1 &\Rightarrow b = b + s_1 \\ s_1 < b \le s_1 + s_2 &\Rightarrow b = b + s_1 + s_2 \end{aligned} \tag{2.26}$$



รูปที่ 2.24 การเลื่อนระนาบบิตระนาบต่อระนาบแบบนัยทั่วไป

การเลื่อนระนาบบิตทั้งวิธี BbBShift และ GBbBShift ได้พัฒนาการเข้ารหัสเน้นบริเวณที่สนใจโดยที่สามารถกำหนดระดับความสนใจได้หลายระดับ ซึ่งวิธีแมกซ์ชิฟท์กำหนดระดับความสนใจได้เพียงระดับเดียว โดยที่ทั้งสองวิธีนี้ยังคงไม่จำเป็นที่จะต้องส่งบริเวณที่สนใจให้กับตัวถอดรหัส เนื่องจากสัมประสิทธิ์บริเวณที่สนใจกับสัมประสิทธิ์พื้นหลังถูกระบุแน่ชัดว่าจะตกอยู่ในระนาบบิตที่เท่าใดด้วยพารามิเตอร์เพียงสองตัวคือ s_1 และ s_2

2.9.3 การเลื่อนระนาบบิตส่วนสำคัญ (Partial Significant Bit-plane Shift, PSBShift) [14]

วิธีนี้เสนอขึ้นโดย Lijie Liu และ Guoliang Fan ได้นำวิธี GBbBShift ของ Zhou Wang มาปรับปรุงอีกครั้งโดยวิธีนี้จะเน้นบริเวณที่สนใจจากระนาบบิตส่วนสำคัญ หรือระนาบบิต

I21943552

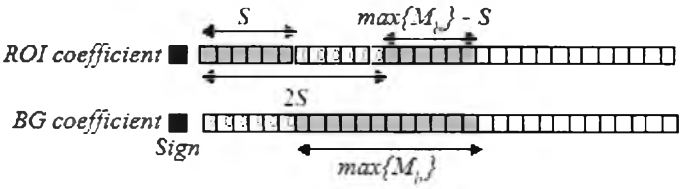
นัยสำคัญ s ระนาบแรกของบริเวณที่สนใจเรียกว่าส่วนสำคัญ แล้วทำการเลื่อนระนาบบิตของส่วนที่เหลือลงไปเท่ากับ s ระนาบ สัมประสิทธิ์พื้นหลังก็เลื่อนระนาบบิตทั้งหมดลงไปเท่ากับ s ระนาบเช่นกันดังรูปที่ 2.25 การเลื่อนระนาบบิตจะทำตามสมการที่ (2.27) และ (2.28) โดยมีพารามิเตอร์ค่าชิฟท์ s เหมือนกับวิธีเจเนรัลชิฟท์

สัมประสิทธิ์ในบริเวณที่สนใจ

$$\begin{aligned} b \le s &\Rightarrow b = b \\ b > s &\Rightarrow b = b + s \end{aligned} \tag{2.27}$$

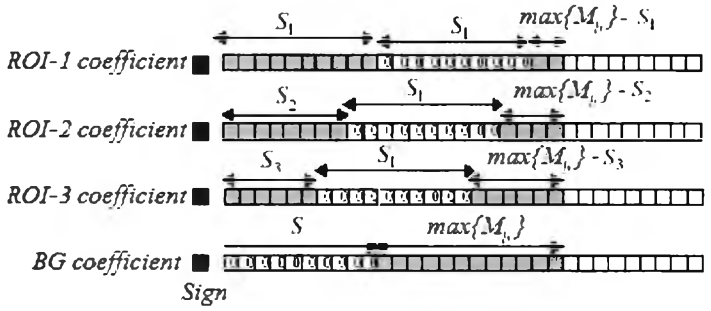
สัมประสิทธิ์ในบริเวณพื้นหลัง

$$b = b + s \tag{2.28}$$



รูปที่ 2.25 การเลื่อนระนาบบิตส่วนสำคัญ

ข้อดีอีกข้อหนึ่งของวิธีนี้คือลักษณะการกำหนดบริเวณที่สนใจได้รับการพัฒนาไปอีกขั้นหนึ่งคือ สามารถกำหนดบริเวณที่สนใจให้มีระดับความสนใจในหลายบริเวณได้ ด้วยการกำหนดบริเวณที่สนใจแยกออกจากกันและทำการเลื่อนระนาบบิตด้วยค่าพารามิเตอร์ s_n ที่ไม่เท่ากัน ค่า n หมายถึงอันดับของบริเวณที่สนใจ ซึ่งการกำหนดบริเวณที่สนใจให้กับภาพโดยที่มีความสำคัญของบริเวณที่สนใจไม่เท่ากันจะต้องทำการเลื่อนระนาบบิตแสดงดังรูปที่ 2.26



รูปที่ 2.26 การเลื่อนระนาบบิตส่วนสำคัญ โดยมีหลายระดับความสนใจ

กำหนดให้มีบริเวณที่สนใจเป็น 3 บริเวณ $n = 3$ และเขียนแทนบริเวณที่สนใจที่ 1, 2 และ 3 ด้วยตัวย่อว่า ROI-1, ROI-2 และ ROI-3 ตามลำดับ บริเวณที่สนใจทั้ง 3 ถูกกำหนดให้มีลำดับความสำคัญไม่เท่ากัน โดยเรียงลำดับความสำคัญจากมากไปน้อยตามลำดับ และทำการเลื่อนระนาบบิตของทุกสัมประสิทธิ์ลงไป $s = s_1$ ระนาบ ซึ่งสัมประสิทธิ์ที่เป็นบริเวณที่สนใจจะทำการเลื่อนตั้งแต่ระนาบบิตที่ s_1, s_2, s_3 ตามลำดับ แต่การเข้ารหัสบริเวณที่สนใจหลายบริเวณโดย

กำหนดระดับความสำคัญของบริเวณที่สนใจยังมีข้อเสียอย่างหนึ่ง ยกตัวอย่างเข้ารหัสเน้นบริเวณที่สนใจด้วยการเลื่อนระนาบบิตส่วนสำคัญดังรูปที่ 2.26 เมื่อตัวถอดรหัสทำการถอดรหัสสายบิตระดับคุณภาพของ ROI-1, ROI-2 และ ROI-3 จะเพิ่มขึ้นจนกระทั่งการถอดรหัสทำไปจนถึง s_1 ระนาบ สัมประสิทธิ์บริเวณพื้นหลังจะเริ่มถูกถอดรหัสตามมา และคุณภาพของบริเวณพื้นหลังจะเพิ่มขึ้นอย่างรวดเร็ว ซึ่งเมื่อทำการถอดรหัสไปจนถึงระนาบบิตที่ $s_1 + s_3$ เป็นไปได้ว่าคุณภาพของบริเวณพื้นหลังจะสูงกว่า ROI-3 หรือถ้ามีการถอดรหัสต่อไปจนถึงระนาบบิตที่ $s_1 + s_2$ คุณภาพของบริเวณพื้นหลังก็อาจสูงกว่า ROI-2 เนื่องจากการพิจารณาของโค้งอัตราความเพี้ยนแล้วคำนวณหาจุดตัดที่เหมาะสมจะพบว่าโค้ดบล็อกที่มีสัมประสิทธิ์พื้นหลังจะได้รับการจัดสรรบิตที่มากกว่าของบริเวณที่สนใจทำให้คุณภาพของพื้นหลังสูงขึ้นอย่างรวดเร็ว

วิทยานิพนธ์นี้ได้เสนอวิธีในการเข้ารหัสเน้นบริเวณที่สนใจ โดยเสนอวิธีการรวมระเบียบวิธีการเลื่อนระนาบบิตเข้าด้วยกัน โดยให้มีลักษณะสามารถกำหนดบริเวณที่สนใจได้หลายบริเวณและสามารถปรับระดับความสนใจได้ละเอียดยิ่งขึ้น