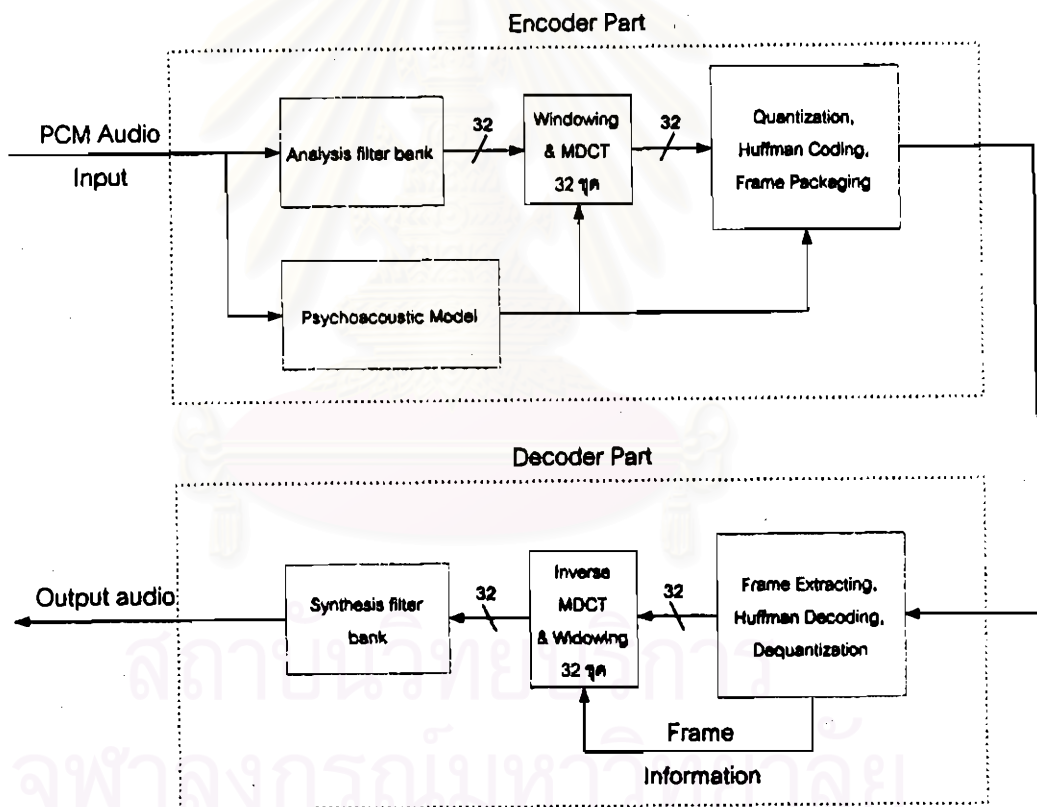


บทที่ 2

การเข้ารหัสและถอดรหัสแบบ MPEG-1 ลำดับชั้น 3

มาตรฐาน MPEG-1 หรือ ISO-11172 เป็นมาตรฐานที่ใช้ในการบีบอัดข้อมูลภาพเคลื่อนไหวและเสียง ภายในมาตรฐานจะแบ่งเป็นสองส่วนคือส่วนของภาพเคลื่อนไหว และส่วนของเสียง โดยที่ส่วนของเสียงจะแบ่งย่อยออกเป็นอีกสามลำดับชั้นคือ ลำดับชั้น 1, ลำดับชั้น 2 และ ลำดับชั้น 3 ความซับซ้อนในการเข้ารหัสจะเพิ่มขึ้นตามลำดับชั้น โดยลำดับชั้นที่ 3 จะมีความซับซ้อนมากที่สุด ซึ่งจะทำให้เราสามารถเข้ารหัสสัญญาณเสียงที่มีความถี่ในการสุ่มตัวอย่าง 44.1 kHz จำนวน 2 ช่องได้ในอัตราส่วนถึง 1:10 การเข้ารหัสและถอดรหัสลำดับชั้น 3 แสดงดังรูปที่ 2.1



รูปที่ 2.1 ขั้นตอนในการเข้ารหัสและถอดรหัสตามมาตรฐาน MPEG-1 ลำดับชั้น 3

2.1 การเข้ารหัส

2.1.1 Analysis Filter Bank

การทำ Analysis Filter Bank เป็นการแยกสัญญาณเสียงที่เก็บตัวอย่างเข้ามาออกเป็น ช่วงความถี่เท่าๆกันทั้งสิ้น 32 ช่วงความถี่ย่อย (Subband) ฟังก์ชันถ่ายโอน (Transfer Function) ในเชิงเวลาของตัวกรองความถี่สำหรับช่วงความถี่ย่อยแต่ละช่วงเป็นไปตามสมการ (1)

$$H_i[n] = h[n] \times \cos\left[\frac{(2i+1)(n-16)\pi}{64}\right] \quad (1)$$

โดยที่ $h[n]$ คือ $-C[n]$ เมื่อจำนวนเต็มที่ปิดเศษทิ้งจากการหาร $n/64$ เป็นจำนวนคี่ มิฉะนั้นจะเท่ากับ $C[n]$ โดยที่ $C[n]$ เป็นค่าคงที่ที่ได้กำหนดไว้แล้วในมาตรฐาน [1]

ในการวิเคราะห์เราจะผ่านค่าอินพุตทีละ 32 ค่า และจะได้ค่าเอาต์พุตดังสมการ (2)

$$s_i[i] = \sum_{n=0}^{511} x[i-n] \times H_i[n] \quad (2)$$

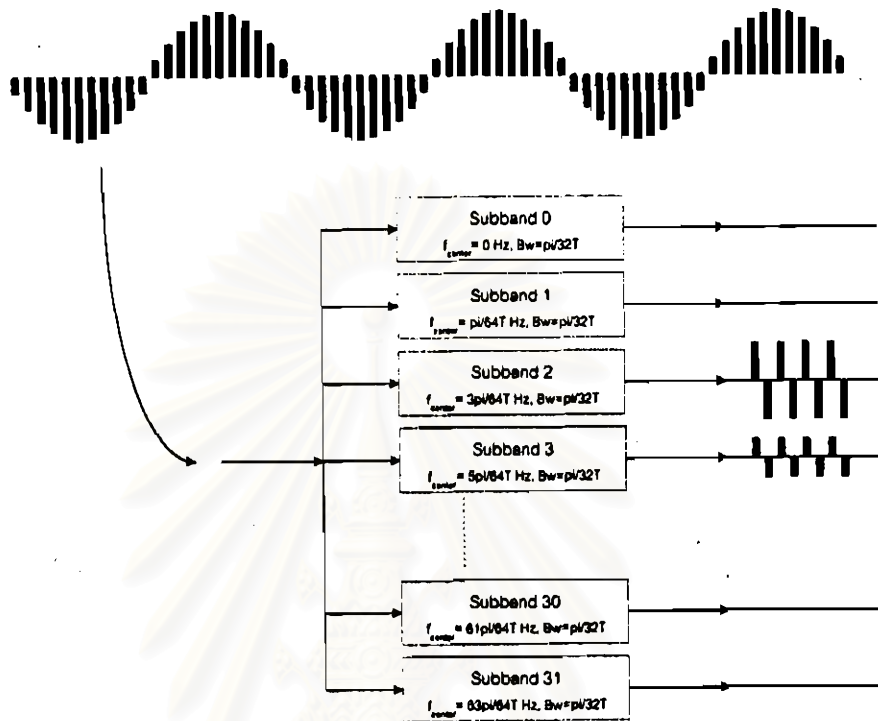
ตัวกรองความถี่ทั้ง 32 ตัวนี้จะมีความถี่ศูนย์กลาง (center frequency) อยู่ที่ จำนวนคี่เท่าของ $\frac{\pi}{64T}$ และมีความกว้างของแถบความถี่ (Bandwidth) $\frac{\pi}{32T}$ เมื่อ T คือคาบของการสุ่มตัวอย่าง

รูปที่ 2.2 แสดงตัวอย่างการผ่านสัญญาณรูปไซน์ความถี่ 1.5 kHz ที่ความถี่ในการสุ่มตัวอย่าง 32 kHz จำนวน 256 ตัวอย่างเข้าสู่ Analysis Filter Bank ซึ่งตัวอย่างทุกๆ 32 ตัวอย่างที่ผ่านเข้าไปจะให้เอาต์พุต subband ละ 1 ตัวอย่าง ดังนั้นสัญญาณ 256 ตัวอย่างจะให้เอาต์พุตทั้งสิ้น subband ละ 8 ตัวอย่าง แต่ในกรณีนี้สัญญาณอินพุตเป็นสัญญาณที่มีความถี่เดียว จึงมีเอาต์พุตที่ subband ที่ 2 และ 3 เท่านั้น ทั้งนี้เนื่องจากช่วงความถี่ย่อยทั้งสองมีผลตอบเชิงความถี่ที่คาบเกี่ยวกันในบางช่วง

2.1.2 Windowing & Modified Discrete Cosine Transform (MDCT)

การทำ MDCT คือการแปลงสัญญาณเชิงเวลาจำนวน $2N$ จุดไปเป็นสัญญาณทางความถี่ N จุด ดังสมการ (3) ก่อนทำ MDCT เราจะนำสัญญาณที่ได้จากช่วงความถี่ย่อยแต่ละช่วงมาผ่าน Window Function ตามรูปที่ 2.3 ซึ่งมีทั้งสิ้น 2 แบบ คือแบบ long block และ short block แบบ long block จะมีความกว้าง 36 ตัวอย่าง แบ่งย่อยออกเป็น 3 แบบคือ type 0, 1 และ 3 ตามสมการ (4), (5) และ (6) และรูปที่ 2.4, 2.5 และ 2.6 ตามลำดับ อีกแบบหนึ่งคือแบบ short block หรือ

type 2 มีความกว้าง 12 ตัวอย่าง ตามสมการ (7) และรูปที่ 2.7 สำหรับ short block นี้จะใช้ครั้ง
ละ 3 รอบ ดังรูปที่ 2.8 ซึ่งก็จะได้ความกว้างเป็น 36 ตัวอย่างเช่นเดียวกับแบบ long block



รูปที่ 2.2 ตัวอย่างการผ่านสัญญาณรูปไซน์ความถี่ 1.5 KHz ที่ความถี่ในการสุ่มตัวอย่าง 32 kHz
จำนวน 256 ตัวอย่างเข้าสู่ Analysis Filter Bank

$$x_i = \sum_{k=0}^{n-1} z_k \cos\left(\frac{\pi}{2n}\left(2k+1+\frac{n}{2}\right)(2i+1)\right), i=0 \text{ to } 35 \quad (3)$$

$$z_i = x_i \sin\left(\frac{\pi}{36}\left(i+\frac{1}{2}\right)\right), i=0 \text{ to } 35 \quad (4)$$

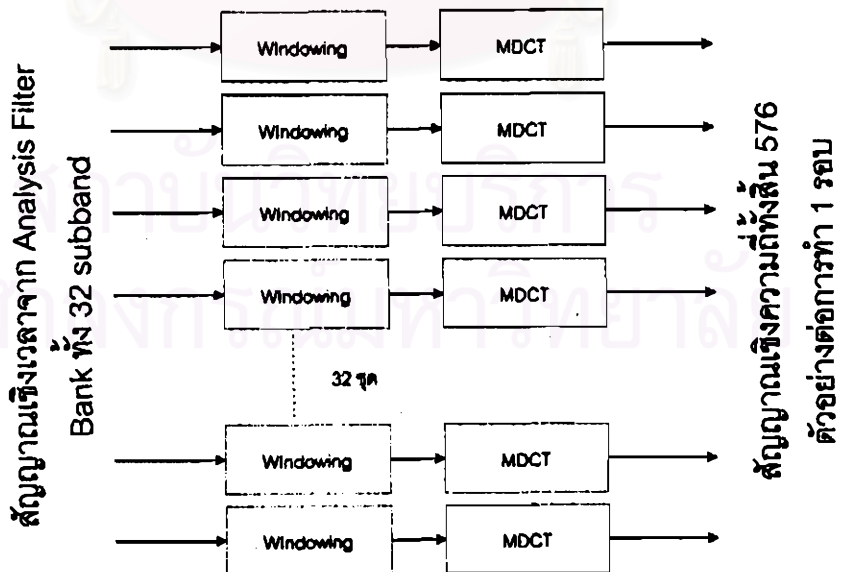
$$z_i = \begin{cases} x_i \sin\left(\frac{\pi}{36}\left(i+\frac{1}{2}\right)\right) & ,i=0 \text{ to } 17 \\ x_i & ,i=18 \text{ to } 23 \\ x_i \sin\left(\frac{\pi}{12}\left(i-18+\frac{1}{2}\right)\right) & ,i=24 \text{ to } 29 \\ 0 & ,i=30 \text{ to } 35 \end{cases} \quad (5)$$

$$z_i = \begin{cases} 0 & ,i=0 \text{ to } 5 \\ x_i \sin\left(\frac{\pi}{12}\left(i-6+\frac{1}{2}\right)\right) & ,i=6 \text{ to } 11 \\ x_i & ,i=12 \text{ to } 17 \\ x_i \sin\left(\frac{\pi}{36}\left(i+\frac{1}{2}\right)\right) & ,i=18 \text{ to } 35 \end{cases} \quad (6)$$

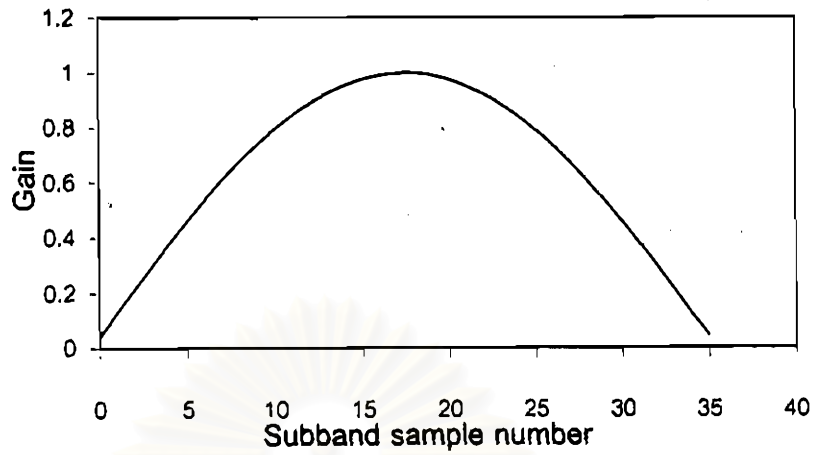
$$z_i^{(j)} = y_i^{(j)} \sin\left(\frac{\pi}{12}\left(i+\frac{1}{2}\right)\right) \quad ,i=0 \text{ to } 11, j=0 \text{ to } 2 \quad (7)$$

เมื่อ

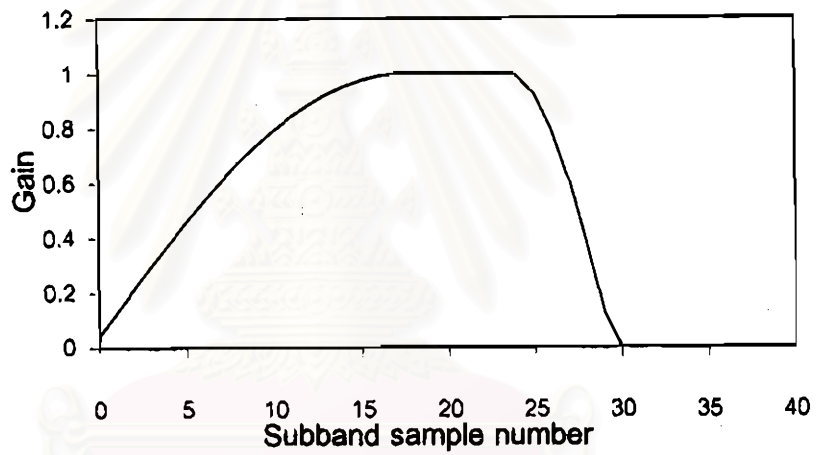
$$\begin{aligned} y_i^{(0)} &= x_{i+6} & ,i=0 \text{ to } 11 \\ y_i^{(1)} &= x_{i+12} & ,i=0 \text{ to } 11 \\ y_i^{(2)} &= x_{i+18} & ,i=0 \text{ to } 11 \end{aligned}$$



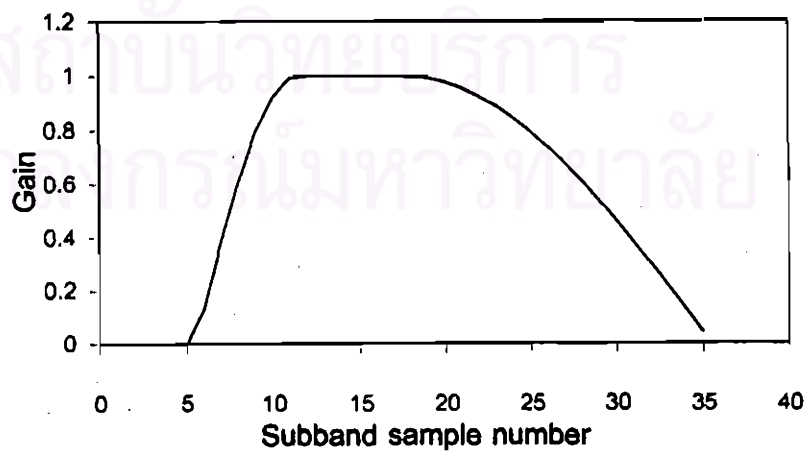
รูปที่ 2.3 โครงสร้างของกระบวนการ Windowing และ MDCT



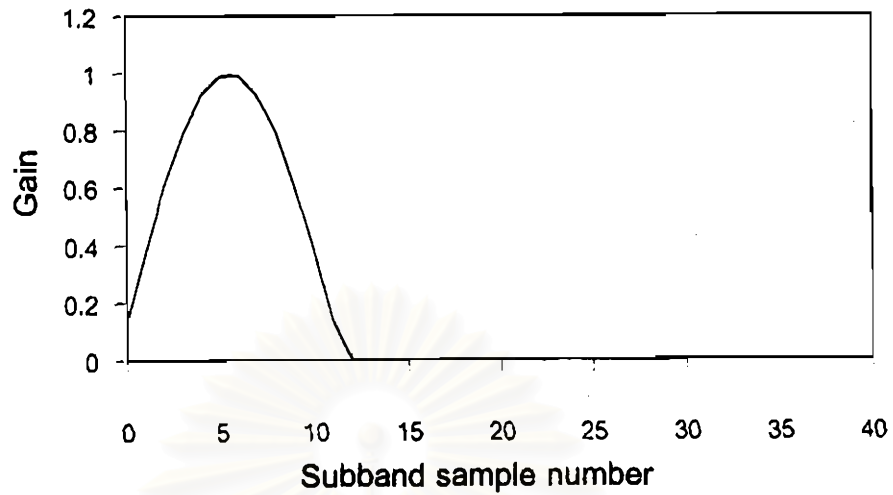
รูปที่ 2.4 อัตราขยายที่ตัวอย่างต่างๆ ของช่วงความถี่ย่อยของหน้าต่าง type 0



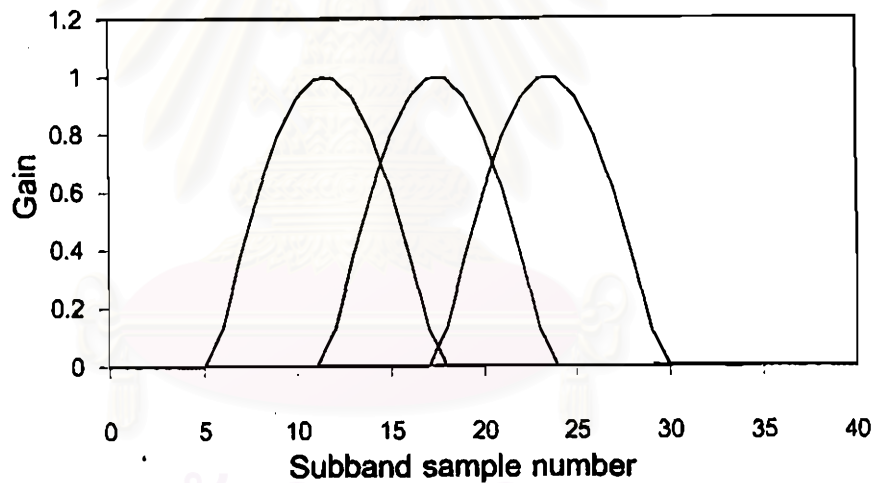
รูปที่ 2.5 อัตราขยายที่ตัวอย่างต่างๆ ของช่วงความถี่ย่อยของหน้าต่าง type 1



รูปที่ 2.6 อัตราขยายที่ตัวอย่างต่างๆ ของช่วงความถี่ย่อยของหน้าต่าง type 3

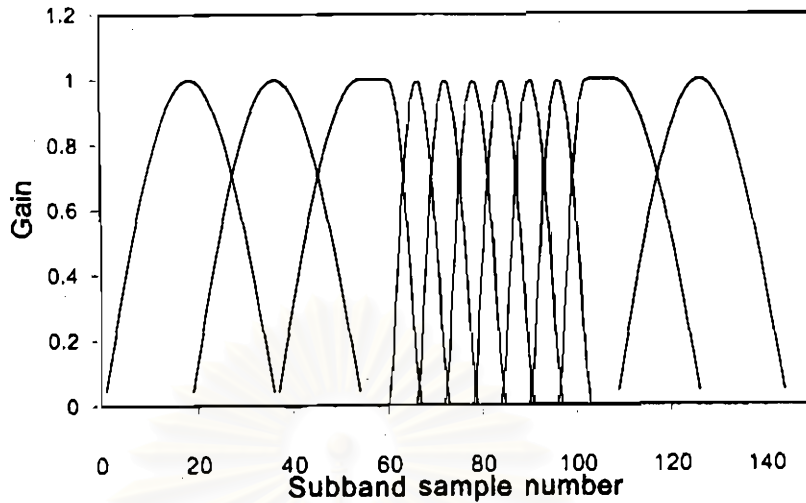


รูปที่ 2.7 อัตราขยายที่ตัวอย่างต่างๆ ของช่วงความถี่ย่อยของหน้าต่าง type 2



รูปที่ 2.8 การใช้งาน block type 2 จำนวน 3 ครั้งใน 1 รอบการทำงาน

ในการวิเคราะห์สัญญาณเชิงเวลาจาก Analysis Subband Filter เริ่มต้นรอบแรกที่ 36 ตัวอย่างและรอบต่อไปจะเพิ่มทีละ 18 ตัวอย่าง โดยหน้าต่างจะมีการเหลื่อมกัน 50% รูปที่ 2.9 แสดงตัวอย่างการใช้ window function แบบ 0, 0, 1, 2, 2, 3 และ 0 ตามลำดับ ซึ่งการวิเคราะห์แต่ละครั้งจะได้สัญญาณเชิงความถี่ 18 ค่าในช่วงความถี่ย่อยแต่ละช่วง



รูปที่ 2.9 ตัวอย่างการใช้ window function แบบที่ 0, 0, 1, 2, 2, 3 และ 0 ตามลำดับ

2.1.3 Psychoacoustic Model

ขั้นตอนนี้จะวิเคราะห์ส่วนประกอบของเสียงที่สามารถตัดทิ้งได้ โดยอาศัยธรรมชาติของการได้ยินของหูรวมถึงปรากฏการณ์ที่สำคัญเกี่ยวกับการได้ยินที่จะนำมาใช้ในการพิจารณา ดังต่อไปนี้

2.1.3.1 ผลตอบสนองทางความถี่ของหู

ประสาทที่ทำหน้าที่ในการฟังของมนุษย์จะมีลักษณะในการตอบสนองต่อเสียงที่ความถี่ต่างๆ คล้ายกับตัวกรองความถี่แบบผ่านแถบ (Band Pass Filter) จากการวิจัย [13] พบว่ามีความถี่ศูนย์กลางของตัวกรองความถี่ทั้งสิ้น 25 ความถี่ โดยที่ตัวกรองความถี่แต่ละตัวจะมีความกว้างของแถบความถี่ (bandwidth) แตกต่างกันไป ดังแสดงในตารางที่ 1

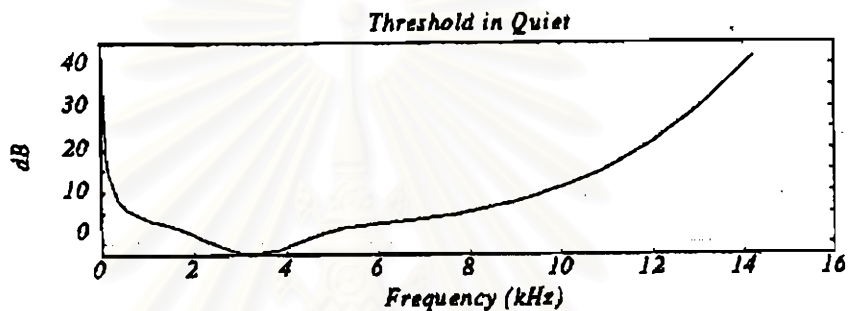
ในตารางที่ 1 ที่ความถี่ต่ำกว่า 500 Hz ค่าของความถี่ศูนย์กลางในแต่ละช่วงจะเพิ่มขึ้นอย่างคงที่ช่วงละประมาณ 100 Hz นอกจากนี้ค่า Q มีค่าคงที่ในช่วง 6-7 ที่ช่วงความถี่ 1000 Hz ถึง 4000 Hz

Critical-band rate (bark) or band number	Lower edge (Hz)	Center frequency (Hz)	Upper edge frequency (Hz)	Bandwidth (Hz)	Q-factor
1	0	50	100	100	0.50
2	100	150	200	100	1.50
3	200	250	300	100	2.50
4	300	350	400	100	3.50
5	400	450	510	110	4.50
6	510	570	630	120	4.75
7	630	700	770	140	5.00
8	770	840	920	150	5.60
9	920	1000	1080	160	6.25
10	1080	1170	1270	190	6.15
11	1270	1370	1480	210	6.52
12	1480	1600	1720	240	6.66
13	1720	1850	2000	280	6.60
14	2000	2150	2320	320	6.72
15	2320	2500	2700	380	6.58
16	2700	2900	3150	450	6.44
17	3450	3400	3700	550	6.18
18	3700	4000	4400	700	5.71
19	4400	4800	5300	900	5.33
20	5300	5800	6400	1100	5.27
21	6400	7000	7700	1300	5.38
22	7700	8500	9500	1800	4.72
23	9500	10500	12000	2500	4.20
24	12000	13500	15500	3500	3.86
25	15500	19500	-	-	-

ตารางที่ 1 การตอบสนองต่อความถี่ในช่วงต่างๆ ของหูในการได้ยิน

2.1.3.2 ธรรมชาติของหูสำหรับการเริ่มได้ยินเสียง

โดยปกติหูของมนุษย์จะมีการตอบสนองต่อการเริ่มได้ยินของเสียงแต่ละความถี่ได้ไม่เท่ากัน ระดับความดันของเสียง (Sound pressure level: SPL) ที่ต่ำที่สุดที่สามารถได้ยินได้คือ $0.0002 \mu\text{bar}$ (micro bar) เทียบเท่า 0 dB และสูงที่สุดคือ $2000 \mu\text{bar}$ เทียบเท่า 140 dB นอกจากนี้ความสามารถในการเริ่มได้ยินเสียงแต่ละความถี่ก็มีระดับความดันของเสียงต่างกันด้วย ดังแสดงในรูปที่ 2.10 เส้นกราฟจะแบ่งค่าของระดับความดันของเสียงที่หูได้ยินและไม่ได้ยิน ที่ขอบของเส้นเราเรียกว่า Threshold in quiet

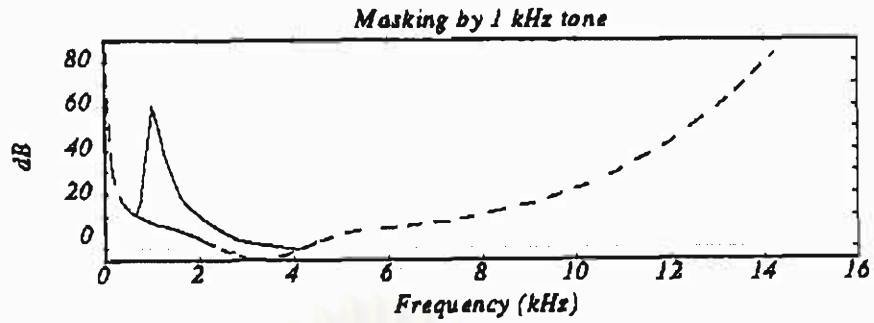


รูปที่ 2.10 เส้นแบ่งเขตเริ่มต้นการได้ยิน

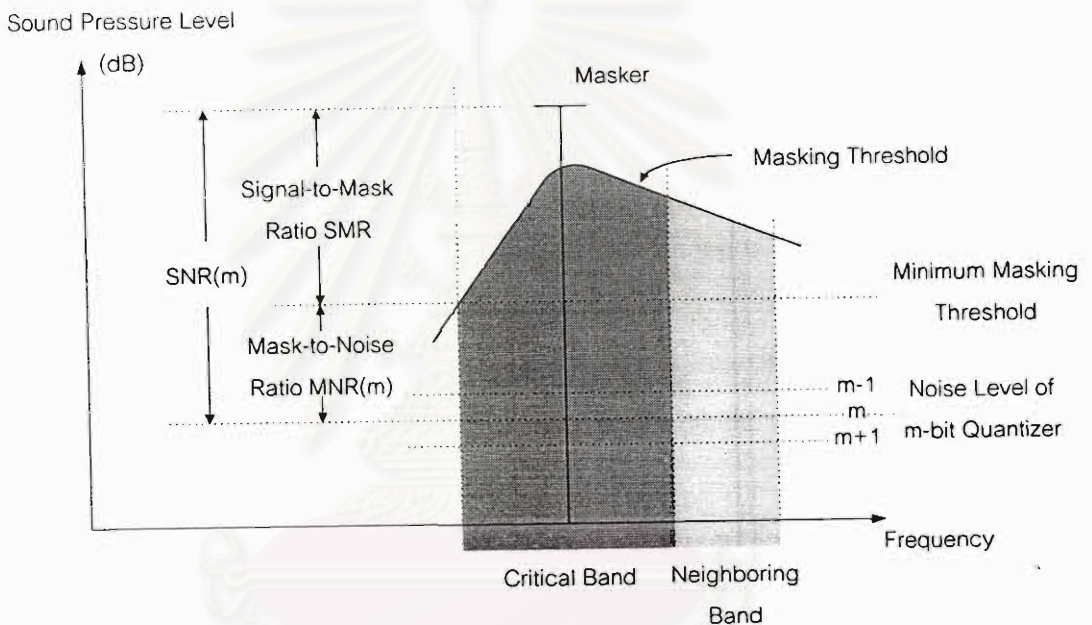
2.1.3.3 ปรากฏการณ์ Frequency Masking

เป็นปรากฏการณ์ที่หูไม่สามารถตรวจจับเสียงที่ความถี่หนึ่งได้เนื่องจากมีเสียงอีกความถี่หนึ่งที่ใกล้เคียงมาบังไว้ (mask) จนกว่าความถี่นั้นจะมีความดันของเสียงเพิ่มขึ้นจนถึงจุดๆ หนึ่ง ตัวอย่างเช่น หากเราเปิดเสียงที่ความถี่ 1 kHz ด้วยระดับคงที่ 60 dB แล้วเปิดเสียงความถี่ 1.1 kHz เพียงเล็กน้อย เราจะไม่สามารถตรวจจับเสียงที่ 1.1 kHz นั้นได้ จนกว่าจะเร่งระดับให้มากขึ้นจนถึงจุดๆ หนึ่งเท่านั้น เราเรียกช่วงที่ไม่สามารถได้ยินเสียงที่ 1.1 kHz ว่าถูกมาสก์โดยเสียงที่ความถี่ 1 kHz ดังแสดงในรูปที่ 2.11 เสียงที่อยู่ใต้เส้นกราฟจะไม่สามารถได้ยินได้ และในกรณีนี้เราเรียกเสียงที่ความถี่ว่าตัวมาสก์ (Masker)

ปรากฏการณ์นี้จะทำให้เราได้นิยามค่าพารามิเตอร์ที่สำคัญขึ้นมาอีกสองตัวคือ MNR (Mask-to-Noise Ratio) และ SMR (Signal-to-Mask Ratio) โดยที่ MNR คือระยะระหว่างระดับของสัญญาณรบกวนเนื่องจากการควอนไทซ์ (Noise Level of Quantizer) ถึงจุดต่ำสุดของขอบเขตการมาสก์ในช่วงความถี่นั้น และ SMR คือระยะตั้งแต่ตัวมาสก์ถึงจุดต่ำสุดของขอบเขตการมาสก์ในช่วงความถี่นั้น ดังแสดงตัวอย่างในรูปที่ 2.12



รูปที่ 2.11 ตัวอย่างของปรากฏการณ์ Frequency Masking



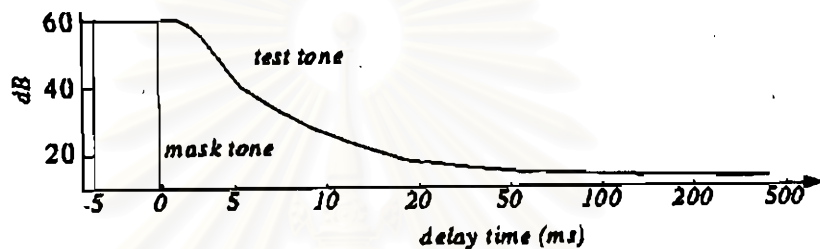
รูปที่ 2.12 นิยามของค่า MNR (Mask-to-Noise Ratio) และ SMR (Signal-to-Mask Ratio)

2.1.3.4 ปรากฏการณ์ Temporal Masking

คือปรากฏการณ์ที่เราไม่สามารถได้ยินเสียงความถี่หนึ่ง ในช่วงเวลาหนึ่งเนื่องจากเสียงอีกความถี่หนึ่งที่มีความดันของเสียงที่มากกว่าเพิ่งเงียบไป ตัวอย่างเช่น หากเราเปิดเสียงที่ความถี่ 1 kHz ไว้สักครู่ เมื่อทำการปิดเสียงแล้วเปิดเสียงที่ความถี่ 1.1 kHz ในทันทีทันใด เราจะไม่สามารถได้ยินเสียงที่ความถี่ 1.1 kHz ได้จนกว่าเวลาจะผ่านไปช่วงหนึ่ง โดยระยะเวลาที่ไม่สามารถได้ยินนั้นจะขึ้นอยู่กับความดันของเสียงนั้น ถ้าความดันยิ่งน้อยก็จะต้องใช้เวลานานกว่าที่จะสามารถได้ยินได้ ตัวอย่างของกราฟของปรากฏการณ์นี้แสดงดังรูปที่ 2.13

ผลลัพธ์ที่ได้จากการวิเคราะห์จะเป็นค่า SMR ของช่วงความถี่ย่อยแต่ละช่วงสำหรับลำดับชั้น 1 และ 2 และ SMR แต่ละช่วงของ Scale Factor สำหรับ ลำดับชั้น 3

ค่า SMR ที่ได้จะเป็นตัวบอกว่าควรจะใช้จำนวนบิตเท่าใดในการเข้ารหัสสเปกตรัมในช่วงความถี่ย่อยนั้นๆ จากรูปที่ 2.13 หากลดจำนวนบิตที่ใช้ในการเข้ารหัสลง ค่า MNR ก็จะลดลงด้วย แต่ตราบิตที่ค่า MNR ยังเป็นบวกหรือยังอยู่ในช่วงถูกมาสก์อยู่ การผิดพลาดของเสียงเนื่องจากการลดบิตในการเข้ารหัสก็จะไม่สามารถได้ยินได้



รูปที่ 2.13 ตัวอย่างของปรากฏการณ์ Temporal Masking

Psychoacoustic Model จะมีสองแบบคือแบบที่ 1 และแบบที่ 2 ให้เลือกใช้ โดยแบบที่ 1 จะมีความซับซ้อนน้อยกว่า สำหรับลำดับชั้น 1 และ 2 สามารถเลือกใช้ได้ทั้งสองแบบ แต่โดยปกติจะใช้กับแบบที่ 1 ส่วน ลำดับชั้น 3 จะใช้แบบที่ 2 เท่านั้น

Psychoacoustic Model แบบที่ 2 จะวิเคราะห์สัญญาณเสียงในช่วงความถี่ที่ละเอียดกว่าแบบที่ 1 ซึ่งจะทำได้ค่า SMR ที่ดีกว่า

2.1.4 Quantization, Huffman Coding และ Frame Packaging

ขั้นตอนการทำควอนไทซ์จะเป็นการนำค่า SMR ที่ได้จาก Psychoacoustic Model มาพิจารณาในการเข้ารหัสสัญญาณเชิงความถี่ทั้ง 576 ค่าให้เหมาะสมที่จะทำให้ค่า MNR เป็นบวกอยู่เสมอ

หลังจากนั้นจะส่งสัญญาณต่อไปเพื่อทำ Huffman Coding โดยจะแบ่งสัญญาณเชิงความถี่ทั้ง 57 ค่าเป็น 3 ส่วน คือในช่วงที่สัญญาณมีค่ามากซึ่งปกติก็คือค่าในช่วงความถี่ต่ำนั่นเอง ช่วงนี้จะถูกเข้ารหัสด้วยตารางที่มีให้เลือกทั้งสิ้น 32 ตาราง ช่วงถัดมาคือช่วงที่มีค่าเพียง 3 ค่าคือ 1, 0, และ -1 เท่านั้น ช่วงนี้จะมีตารางให้เลือก 2 ตาราง และช่วงสุดท้ายคือช่วงที่มีค่าเป็น 0 ตลอดจนครบ 576 ตัวอย่าง ในช่วงนี้จะไม่มีการส่งข้อมูลออกไป

ขั้นตอนสุดท้ายคือการทำให้ Frame Packaging ซึ่งจะเป็นการรวบรวมข้อมูลทั้งหมดให้เป็นเฟรมพร้อมกับเพิ่มเติมส่วนหัวของเฟรม ซึ่งจะใช้ในการบอกรายละเอียดที่จำเป็นในการถอดรหัส โดยในหนึ่งเฟรมจะต้องทำขั้นตอนต่างๆ ตั้งแต่ Analysis Filter Bank 2 รอบ ดังนั้นในหนึ่งเฟรมจะประกอบไปด้วยสัญญาณเชิงความถี่ทั้งสิ้น 1152 ค่า

2.2 การถอดรหัส

2.2.1 Frame Extracting, Huffman Decoding, Dequantization

ขั้นตอนนี้จะเป็นการแยกส่วนประกอบต่างๆ ของเฟรมออกมา ข้อมูลในส่วนหัว (header) ของเฟรมจากนั้นจะทำ Huffman decoding และ Dequantization หลังจากทำเสร็จหนึ่งรอบแล้วก็จะได้ค่าสัญญาณในเชิงความถี่จำนวน 576 ตัวอย่าง/ช่อง เพื่อส่งต่อไปยังกระบวนการสร้างสัญญาณเชิงความถี่กลับมาเป็นสัญญาณทางเวลา

2.2.2 IMDCT และ Windowing

ขั้นตอนนี้เป็นการแปลงสัญญาณเชิงความถี่ทั้งสิ้น 576 ตัวอย่าง กลับไปเป็นสัญญาณเชิงเวลาในช่วงความถี่ต่างๆ 32 ช่วงความถี่ การทำ IMDCT เป็นกระบวนการย้อนกลับของ MDCT โดยจะเป็นการแปลงสัญญาณทางความถี่จำนวน N จุดไปเป็นสัญญาณเชิงเวลา $2N$ จุด ดังสมการ (8)

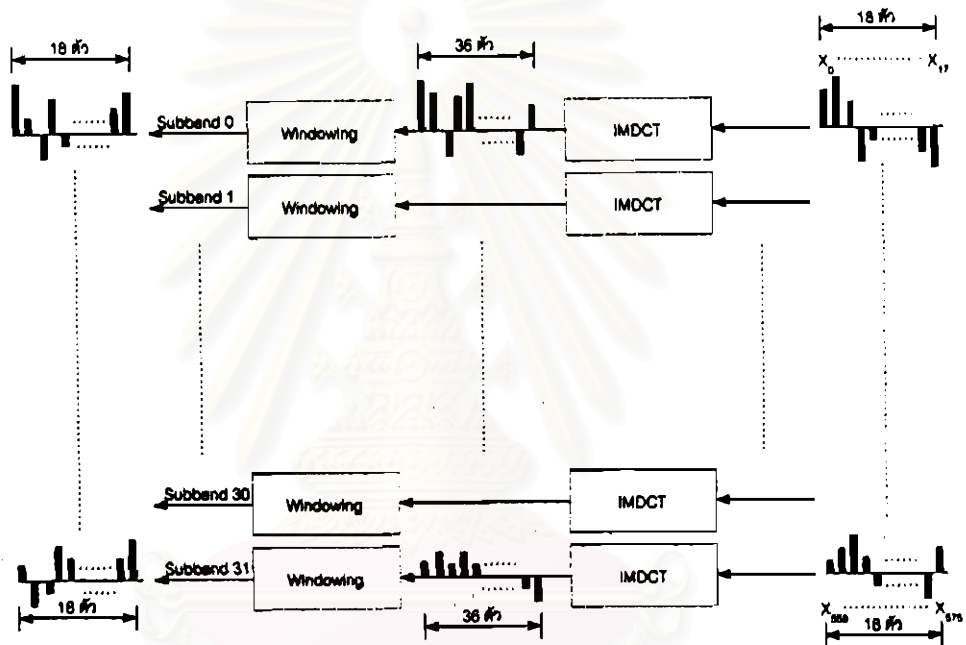
$$x_i = \sum_{k=0}^{n-1} X_k \cos\left(\frac{\pi}{2n}(2i+1+\frac{n}{2})(2k+1)\right) \quad , i = 0 \text{ to } n-1 \quad (8)$$

ในการทำการแปลงเราจะแบ่งสัญญาณเชิงความถี่ 576 ตัวอย่างเป็นชุดเรียงตามลำดับ จากความถี่ต่ำไปหาสูง ชุดละ 18 ตัวอย่างจำนวน 32 ชุด ดังรูปที่ 2.14 ในแต่ละชุดจะมีการคำนวณได้สองแบบขึ้นอยู่กับชนิดของ window block ว่าเป็นแบบ long หรือ short ในกรณีที่เป็น long จะคำนวณตามสมการ (8) ครั้งเดียว โดยให้ $n=36$ หากกรณีเป็น short จะแบ่งย่อยเป็น 3 ชุดๆ ละเท่ากัน และคำนวณตามสมการ (8) ทั้งหมด 3 ครั้ง โดยให้ $n=16$

สมการ (8) จะใช้การบวกทั้งสิ้น $N/2-N$ ครั้ง และการคูณ $N/2$ ครั้ง ในการคำนวณ 1 ชุด หรือ 18 ตัวอย่างของสัญญาณเชิงความถี่ ซึ่งเป็นการยากที่จะนำมาใช้เขียนโปรแกรมจริงที่มีเวลา

ในการคำนวณอย่างจำกัด เราจึงต้องปรับเปลี่ยนวิธีการคำนวณให้เร็วขึ้นซึ่งจะกล่าวในหัวข้อที่ 3 ต่อไป

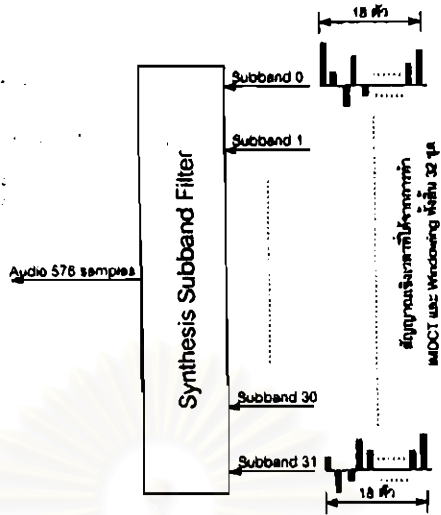
หลังจากทำ IMDCT เสร็จแล้ว ก็จะได้จำนวน x มากขึ้นเป็นชุดละ 36 ตัว เราจะสัญญาณนำมาผ่าน window function ตามสมการที่ (4), (5), (6) หรือ (7) ผลลัพธ์ที่ได้ 18 ตัวแรกจะต้องบวกด้วยผลลัพธ์ที่ได้ของ 18 ตัวหลังของการคำนวณรอบที่แล้ว ซึ่งผลของการบวกจะเป็นเอาต์พุตของกระบวนการ IMDCT และ windowing นี้ ส่วน 18 ตัวหลังของการทำ windowing จะถูกเก็บเอาไว้บวกกับผลลัพธ์ที่ได้ 18 ตัวแรกของการทำ windowing ในรอบการคำนวณถัดไป



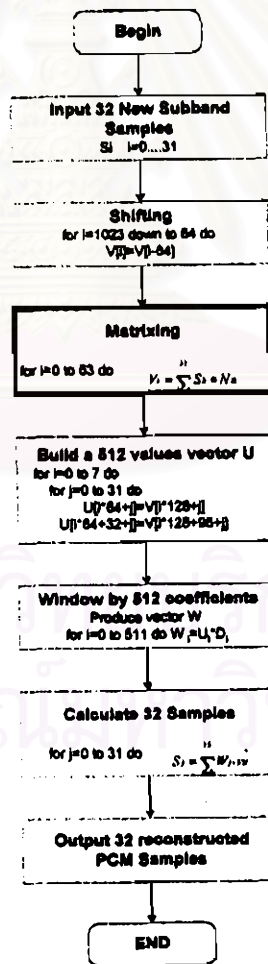
รูปที่ 2.14 การทำ IMDCT และ Windowing

2.3.2 การทำ Synthesis Filter Bank

ขั้นตอนนี้เป็น การสร้างสัญญาณเชิงเวลาใน 32 ช่วงความถี่ย่อย (subband) กลับเป็นสัญญาณเชิงเวลาตามปกติ ดังแสดงในรูปที่ 2.15 ซึ่งเป็นการทำย้อนกลับกระบวนการ Analysis Subband Filter ขั้นตอนการทํานั้น [1] ได้เสนอไว้ดังรูปที่ 2.16



รูปที่ 2.15 การทำ Synthesis Filter Bank



รูปที่ 2.16 ขั้นตอนการทำ Synthesis Filter Bank

จากรูปที่ 2.16 ขั้นตอนที่ใช้เวลาในการคำนวณมากที่สุดคือขั้นตอนที่ 3 ซึ่งเป็นขั้นตอนของ Matrixing

$$V_i = \sum_{k=0}^{31} S_k N_{ik} \quad , i = 0 \text{ to } 63 \quad (9)$$

เมื่อ

$$N_{ik} = \cos\left[(16+i)(2k+1)\frac{\pi}{64}\right]$$

S_k คือ สัมประสิทธิ์อินพุตของ Analysis Filter Bank

จะเห็นว่าในขั้นตอนนี้ต้องมีการบวกและคูณเป็นจำนวนมาก เราจำเป็นต้องดัดแปลงวิธีการคำนวณใหม่เพื่อให้สามารถทำงานได้รวดเร็วยิ่งขึ้น

2.3 การดัดแปลงวิธีการคำนวณ

2.3.1 การทำ IMDCT และ Windowing

จาก [1] ได้กำหนดวิธีการทำในขั้นตอนนี้ดังสมการ (10) โดยที่ $n=12$ และ 36 เมื่อข้อมูลเป็น short block และ long block ตามลำดับ

$$x_i = \sum_{k=0}^{\frac{n}{2}-1} X_k \cos\left(\frac{\pi}{2n}(2i+1+\frac{n}{2})(2k+1)\right) \quad , i = 0 \text{ to } n-1 \quad (10)$$

ในกรณีของ Short Block

$$x_i = \sum_{k=0}^5 X_k \cos\left(\frac{\pi}{24}(2i+1+6)(2k+1)\right) \quad , i = 0 \text{ to } 11$$

ให้

$$g_i = \sum_{k=0}^5 X_k \cos\left(\frac{\pi}{24}(2i+1)(2k+1)\right) \quad , i = 0 \text{ to } 5 \quad (11)$$

เราจะได้ว่า g_i คือการทำ IDCT (Inverse Discrete Cosine Transform) 12 จุด โดยที่ g_i คือลำดับที่เป็นเลขคู่ของลำดับอินพุต 12 ตัว ที่มีลำดับที่เป็นเลขคี่เป็นศูนย์ทั้งหมดตามอัลกอริทึมของ Lee [7] ดังนั้นเราสามารถสรุปขั้นตอนการทำ IMDCT (Inverse Modified Discrete Cosine Transform) แบบ fast algorithm ได้ดังนี้

1. สร้างลำดับขนาด 12 ตัว ที่มีค่าเป็นศูนย์ทั้งหมด
2. แทนค่า ลงในลำดับที่เป็นเลขคี่ที่สร้างขึ้น
3. ใช้อัลกอริทึมของ Lee แบ่ง ออกเป็นการทำ IDCT 3 จุด 2 ชุด
4. หา IDCT 3 จุดด้วยวิธีตามปกติ
5. รวม IDCT 3 จุด 2 ชุด เป็น 6 จุด
6. รวม IDCT 6 จุด 2 ชุด เป็น 12 จุด จะได้ ตาม (2)
7. จุดเรียง ด้วยวิธีการเปลี่ยนลำดับเพื่อให้ได้

ในกรณีของ Long Block

จะคล้ายกับวิธี Short Block เพียงแต่เริ่มต้นด้วยลำดับขนาด 24 จุดและต้องหา IDCT ขนาด 9 จุดด้วยวิธีปกติ ส่วนขั้นตอนอื่นก็ทำเช่นเดียวกับ Short Block

2.3.2 การทำ Synthesis Filter Bank

จากรูปที่ 2.16 ในขั้นตอนที่ 3

$$V_i = \sum_{k=0}^{31} S_k \cdot N_{ik} \quad , i = 0 \text{ to } 63 \quad (12)$$

เมื่อ

$$N_{ik} = \cos\left[(16+i)(2k+1)\frac{\pi}{64}\right]$$

S_k คือ สัญญาณอินพุตของ Polyphase filter bank

เมื่อแทนค่า N_{ik} ลงไปใน (3) เราจะได้ว่า

$$V(i) = X(i+16) \quad , i = 0 \text{ to } 63 \quad (13)$$

เมื่อ $X(i)$ คือผลของ DCT (Discrete Cosine Transform) [9] ของ S_k มีสมการเป็น

$$X(m) = \sum_{k=0}^{31} S_k \cos\left[\frac{(2k+1)\pi m}{64}\right], \quad m = 0 \text{ to } 31$$

ดังนั้นหากเราใช้การทำ FDCT (Fast Discrete Cosine Transform) ต่อ S_k เมื่อ $N=32$ ก็จะได้ $X(m)$ วิธีการทำ FDCT มีหลายแบบ ในงานวิจัยนี้จะเลือกอัลกอริทึมของ Lee [7] $X(m)$ ที่ได้มีเพียง 32 ค่า เมื่อเรานำไปแทนใน (13) จึงต้องมีการจัดเรียงใหม่ตาม (14) [8]

$$V(i) = \begin{cases} X(i+16) & , i = 0,1,\dots,15 \\ 0 & , i = 16 \\ -X(48-i) & , i = 17,18,\dots,48 \\ -X(i-48) & , i = 49,50,\dots,63 \end{cases} \quad (14)$$

นอกเหนือจากการเปลี่ยนวิธีการคำนวณแล้ว ในการเขียนโปรแกรมจริงต้องคำนึงถึงการดึงเอาประสิทธิภาพของตัวประมวลผลออกมาใช้ให้มากที่สุด ซึ่งจะกล่าวรายละเอียดในบทที่ 6

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย