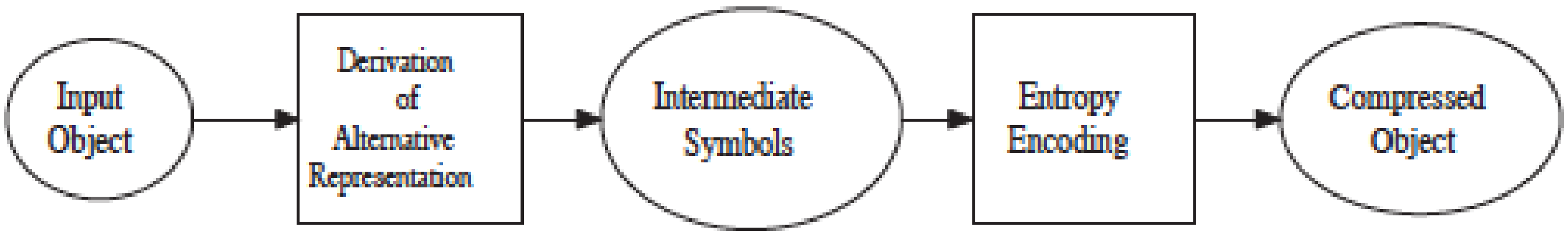
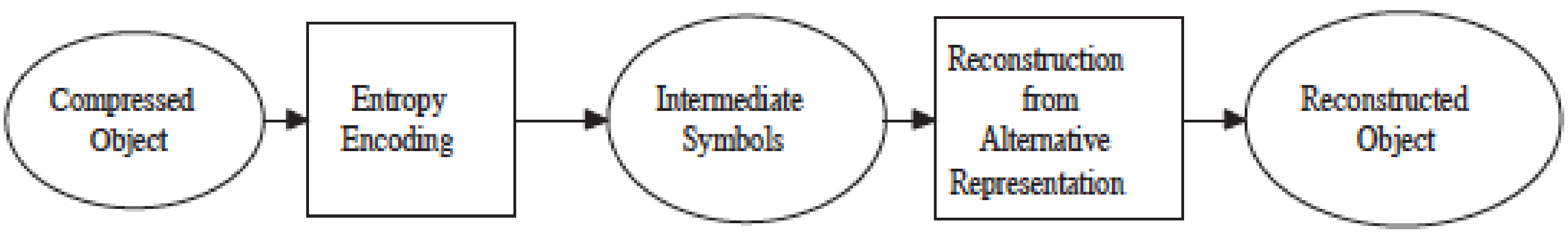
Processing in block DCT domain

Jayanta Mukhopadhyay
Dept. of Computer Science and Engg.

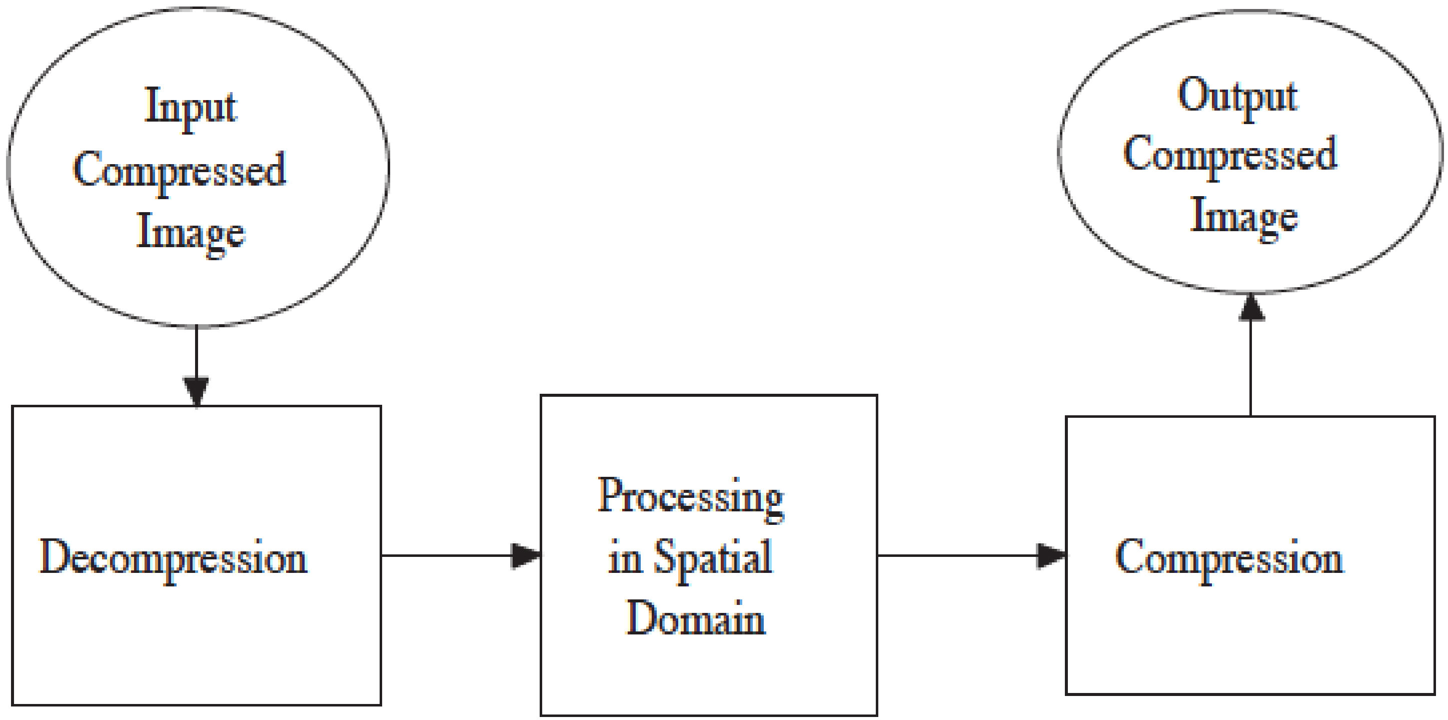
"Image and video processing in the compressed domain", CRC Press, 2011.

Image compression and decompression

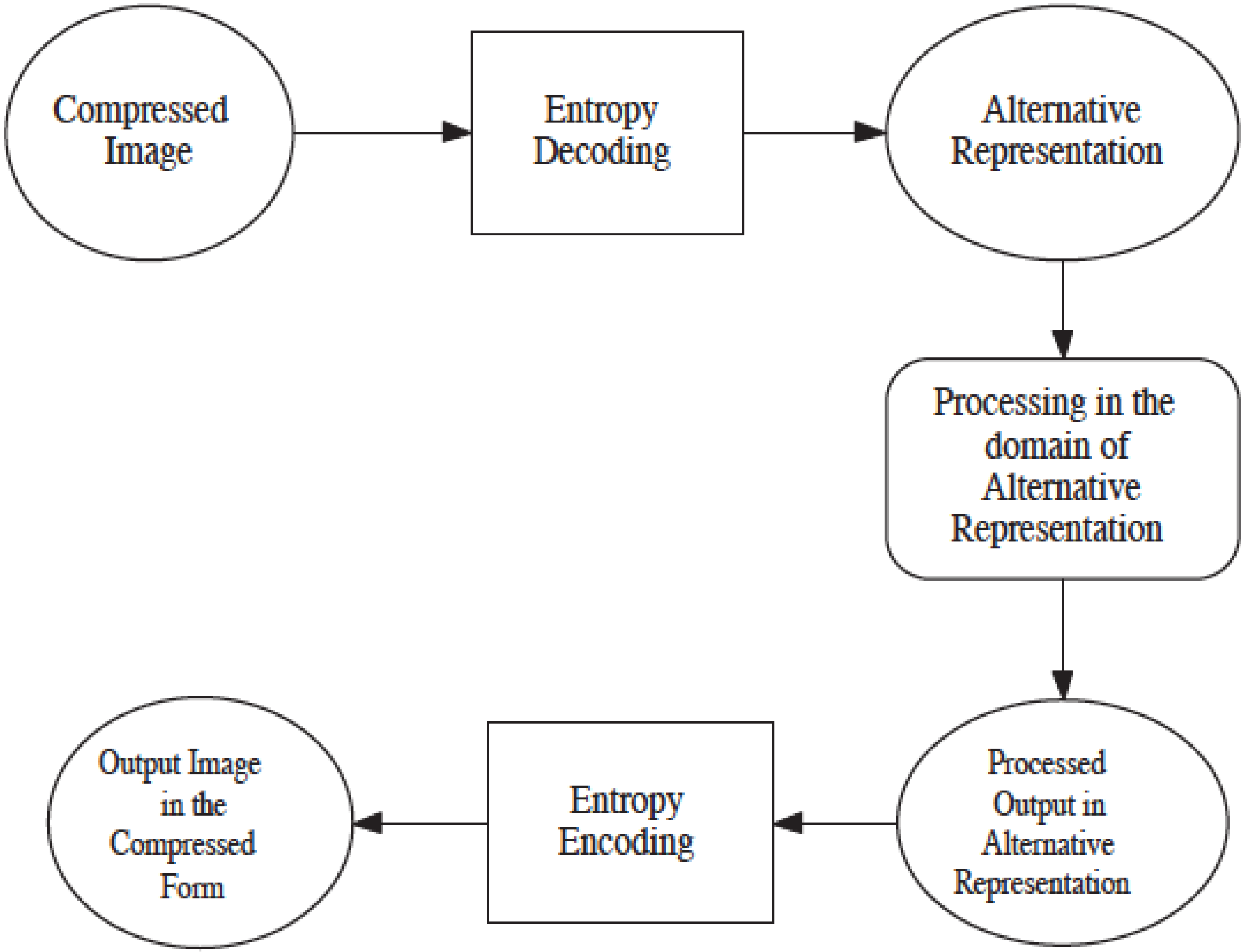




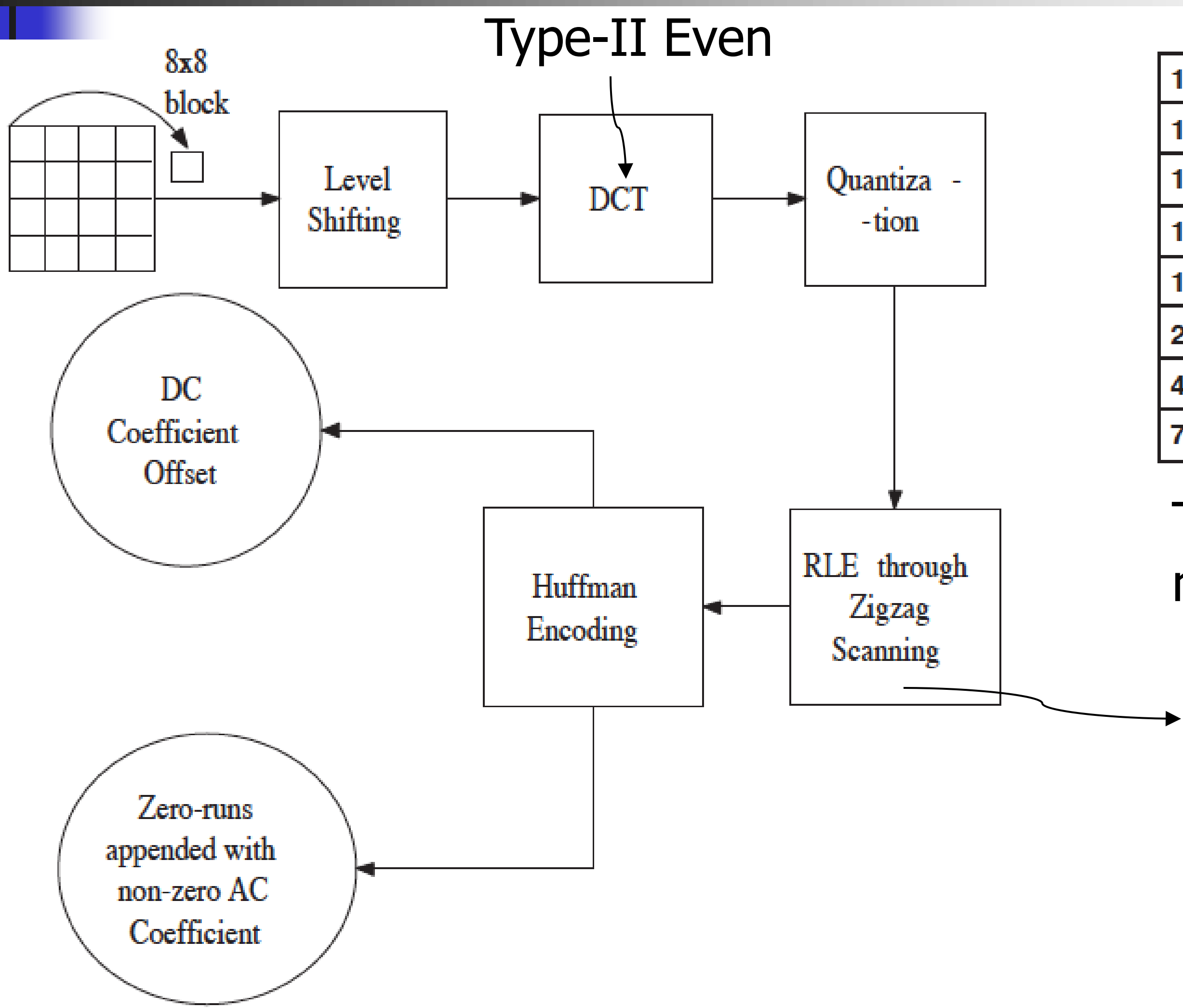
Processing with compressed image: Spatial domain approach



Processing with compressed image: Compressed domain approach

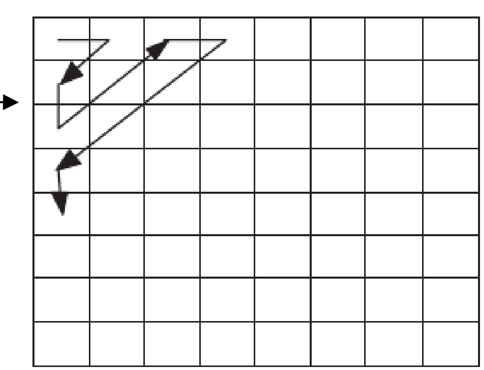


JPEG: Baseline scheme



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	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

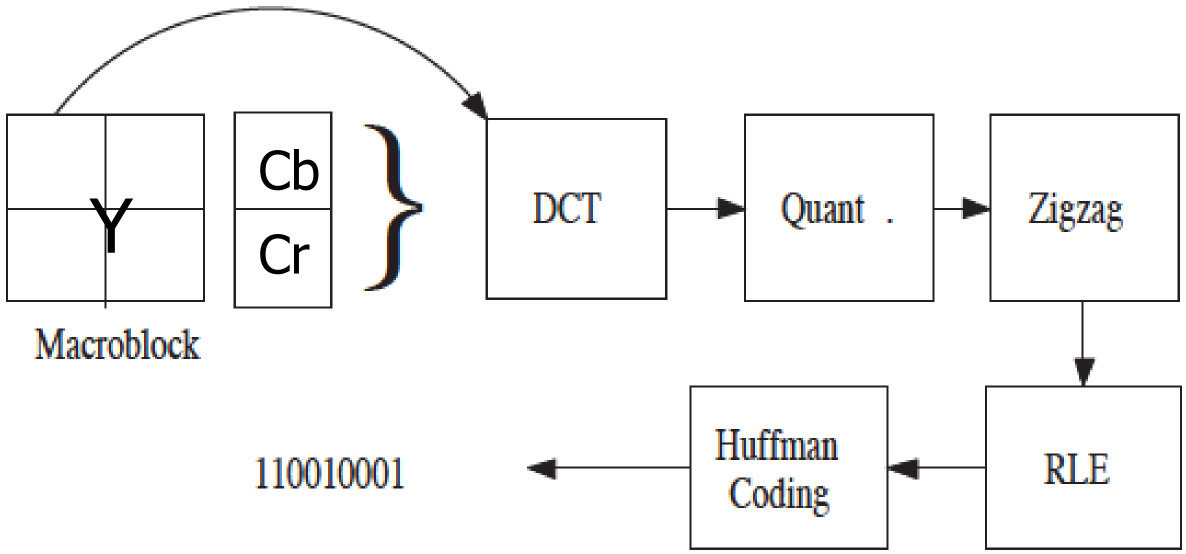
Typical quantization mask.



Color encoding in JPEG

- Y-Cb-Cr color space:

$$
\begin{aligned}
Y &= 0.502G + 0.098B + 0.256R, \\
Cb &= -0.290G + 0.438B - 0.148R + 128, \\
Cr &= -0.366G - 0.071B + 0.438R + 128.
\end{aligned}
$$



Motivations

- Computation with reduced storage.
- Avoid overhead of entropy decoding and encoding.
- Exploit spectral factorization for improving the quality of result and speed of computation.

Typical Applications

- Resizing.
- Filtering.
- Enhancement.

Image Resizing

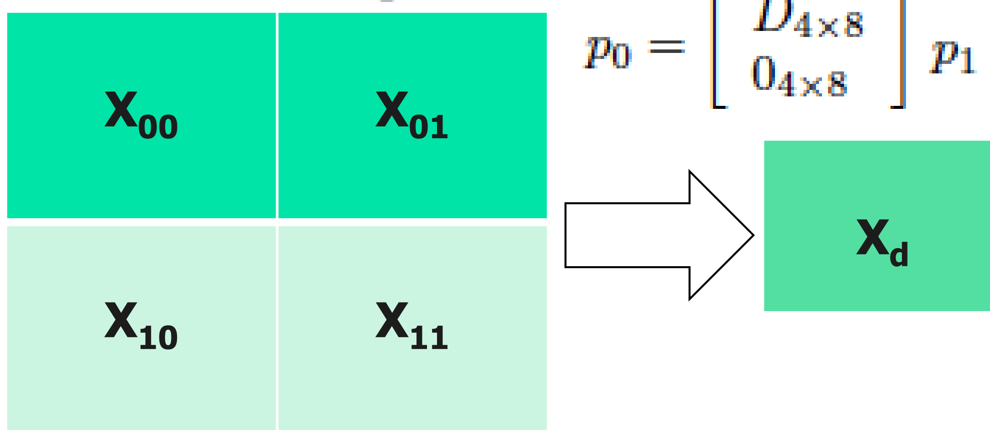
Image Halving

- Use of linear and distributive properties.

$$D_{4\times 8} = \begin{bmatrix} 0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 \end{bmatrix}$$

$$\mathbf{x_d} = \Sigma_{j=0}^{1}\Sigma_{i=0}^{1} p_i \mathbf{x_{ij}} p_j^T ;$$

$$p_0 = \begin{bmatrix} D_{4\times 8} \\ 0_{4\times 8} \end{bmatrix} \quad p_1 = \begin{bmatrix} 0_{4\times 8} \\ D_{4\times 8} \end{bmatrix}$$



$$DCT(\mathbf{x_d}) = \Sigma_{j=0}^{1}\Sigma_{i=0}^{1} DCT(p_i) DCT(\mathbf{x_{ij}}) DCT(p_j^T).$$

Not so sparse matrix multiplication!

DCT(p_0): Not so sparse. No gain!

$$\begin{bmatrix} 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.453 & 0.204 & -0.034 & 0.010 & 0 & -0.006 & 0.014 & -0.041 \\ 0 & 0.49 & 0 & 0 & 0 & 0 & 0 & -0.098 \\ -0.159 & 0.388 & 0.237 & -0.041 & 0 & 0.027 & -0.098 & -0.077 \\ 0 & 0 & 0.462 & 0 & 0 & 0 & -0.191 & 0 \\ 0.106 & -0.173 & 0.355 & 0.204 & 0 & -0.136 & -0.147 & 0.034 \\ 0 & 0 & 0 & 0.416 & 0 & -0.278 & 0 & 0 \\ -0.090 & 0.136 & -0.173 & 0.360 & 0 & -0.240 & 0.072 & -0.027 \end{bmatrix},$$

DCT(p_1)

$$\begin{bmatrix} 0.500 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.453 & 0.204 & 0.034 & 0.010 & 0 & -0.006 & -0.014 & -0.041 \\ 0 & -0.49 & 0 & 0 & 0 & 0 & 0 & 0.098 \\ 0.159 & 0.388 & -0.237 & -0.041 & 0 & 0.027 & 0.098 & -0.077 \\ 0 & 0 & 0.462 & 0 & 0 & 0 & -0.191 & 0 \\ -0.106 & -0.173 & -0.355 & 0.204 & 0 & -0.136 & 0.147 & 0.034 \\ 0 & 0 & 0 & -0.416 & 0 & 0.278 & 0 & 0 \\ 0.090 & 0.136 & 0.173 & 0.360 & 0 & -0.240 & -0.072 & -0.027 \end{bmatrix}$$

Typical result:

Original

Bi-linear

Linear and distributive method

2D DCT: Sub-band relation

$$\begin{aligned} x_{LL}(m,n) &= \tfrac{1}{4}\{x(2m,2n) + x(2m+1,2n) \\ &\quad + x(2m,2n+1) + x(2m+1,2n+1)\},\ 0 \le m,n \le \tfrac{N}{2} - 1. \end{aligned}$$

Low-pass truncated approximation:

$$X(k,l) = \begin{cases} 2\overline{X_{LL}}(k,l), & k,l = 0,1,......,\frac{N}{2}-1 \\ 0, & \text{otherwise.} \end{cases}$$

Block composition and decomposition

$$X^{(N)} = C_N \mathrm{x}$$

- To convert M adjacent N-point DCT blocks to a single MxN-point DCT block.

$$X^{(MN)} = A_{(M,N)}[X_0^{(N)} X_1^{(N)} \ldots X_{M-1}^{(N)}]^T;$$

$$A_{(M,N)} = C_{MN} \begin{bmatrix} C_N^{-1} & 0_N & 0_N & \cdots & 0_N & 0_N \\ 0_N & C_N^{-1} & 0_N & \cdots & 0_N & 0_N \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0_N & 0_N & 0_N & \cdots & C_N^{-1} & 0_N \\ 0_N & 0_N & 0_N & \cdots & 0_N & C_N^{-1} \end{bmatrix},$$

NxN zero matrix

2D DCT: Block composition and decomposition

$$X^{(LN\times MN)} = A_{(L,N)} \begin{bmatrix} X_{0,0}^{(N\times N)} & X_{0,1}^{(N\times N)} & \cdots & X_{0,M-1}^{(N\times N)} \\ X_{1,0}^{(N\times N)} & X_{1,1}^{(N\times N)} & \cdots & X_{1,M-1}^{(N\times N)} \\ \vdots & \vdots & \ddots & \vdots \\ X_{L-1,0}^{(N\times N)} & X_{L-1,1}^{(N\times N)} & \cdots & X_{L-1,M-1}^{(N\times N)} \end{bmatrix} A_{(M,N)}^{T}$$

$$\begin{bmatrix} X_{0,0}^{(N\times N)} & X_{0,1}^{(N\times N)} & \cdots & X_{0,M-1}^{(N\times N)} \\ X_{1,0}^{(N\times N)} & X_{1,1}^{(N\times N)} & \cdots & X_{1,M-1}^{(N\times N)} \\ \vdots & \vdots & \ddots & \vdots \\ X_{L-1,0}^{(N\times N)} & X_{L-1,1}^{(N\times N)} & \cdots & X_{L-1,M-1}^{(N\times N)} \end{bmatrix} = A_{(L,N)}^{-1} X^{(LN\times MN)} A_{(M,N)}^{-1^T}$$

Useful conversion for halving or doubling 8-point DCT blocks.

$$
\begin{aligned}
A_{(2,4)} &= C_8 . \begin{bmatrix} C_4^{-1} & 0_4 \\ 0_4 & C_4^{-1} \end{bmatrix} \\
&= \begin{bmatrix}
0.7071 & 0 & 0 & 0 & 0.7071 & 0 & 0 & 0 \\
0.6407 & 0.294 & -0.0528 & 0.0162 & -0.6407 & 0.294 & 0.0528 & 0.0162 \\
0 & 0.7071 & 0 & 0 & 0 & 0.7071 & 0 & 0 \\
-0.225 & 0.5594 & 0.3629 & -0.0690 & 0.225 & 0.5594 & -0.3629 & -0.069 \\
0 & 0 & 0.7071 & 0 & 0 & 0 & 0.7071 & 0 \\
0.1503 & -0.2492 & 0.5432 & 0.3468 & -0.1503 & -0.2492 & -0.5432 & 0.3468 \\
0 & 0 & 0 & 0.7071 & 0 & 0 & 0 & -0.7071 \\
-0.1274 & 0.1964 & -0.2654 & 0.6122 & 0.1274 & 0.1964 & 0.2654 & 0.6122
\end{bmatrix} .
\end{aligned}
$$

Composition

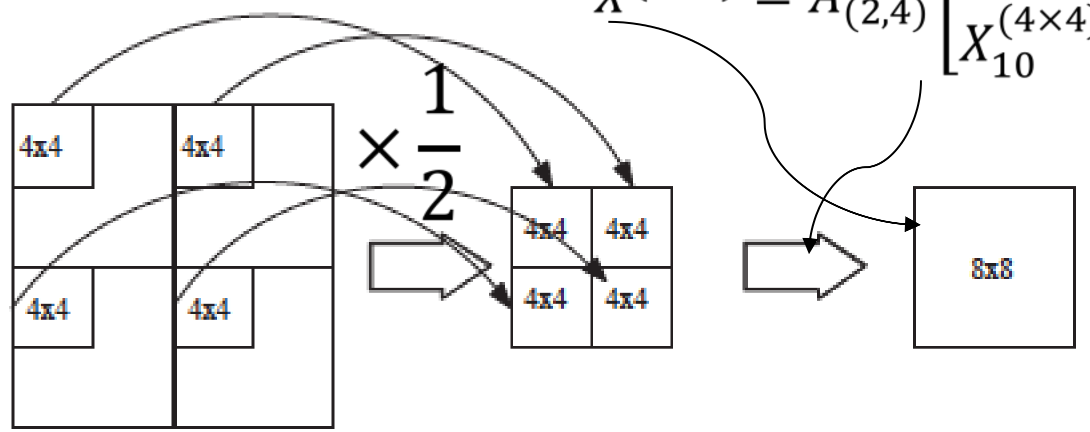
$$X^{(8\times8)} = A_{(2,4)} \begin{bmatrix} X_{00}^{(4\times4)} & X_{01}^{(4\times4)} \\ X_{10}^{(4\times4)} & X_{11}^{(4\times4)} \end{bmatrix} A_{(2,4)}^{T}$$

Decomposition

$$\begin{bmatrix} X_{00}^{(4\times4)} & X_{01}^{(4\times4)} \\ X_{10}^{(4\times4)} & X_{11}^{(4\times4)} \end{bmatrix} = A_{(2,4)}^{-1} X^{(8\times8)} A_{(2,4)}^{-T}$$

Image Halving: Approximation followed by Composition (IHAC)

$$X^{(8\times 8)} = A_{(2,4)} \begin{bmatrix} X_{00}^{(4\times 4)} & X_{01}^{(4\times 4)} \\ X_{10}^{(4\times 4)} & X_{11}^{(4\times 4)} \end{bmatrix} A_{(2,4)}^T$$

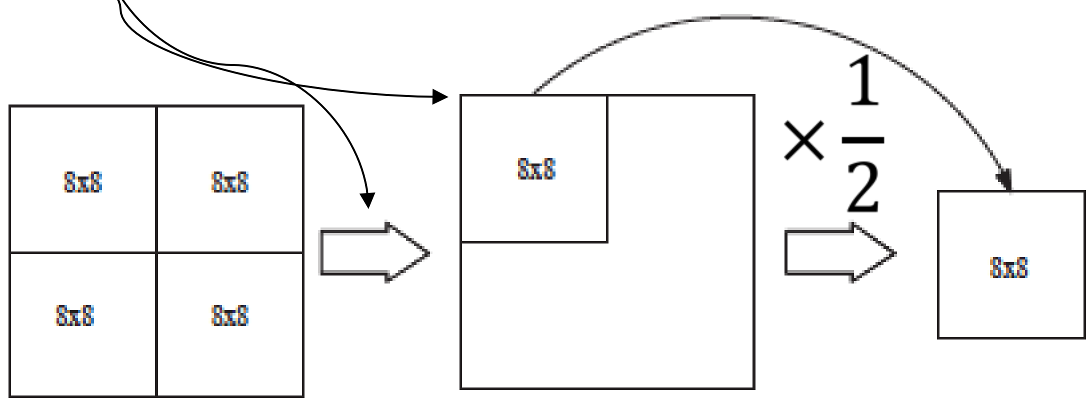


$$X_{ij}^d = \frac{1}{2}\left[X_{ij}(k,l)\right]_{0\le k,l\le 3}.$$

$$Y = A_{(2,4)} \begin{bmatrix} X_{00}^d & X_{01}^d \\ X_{10}^d & X_{11}^d \end{bmatrix} A_{(2,4)}^T$$

Image Halving: Composition followed by Approximation (IHAC)

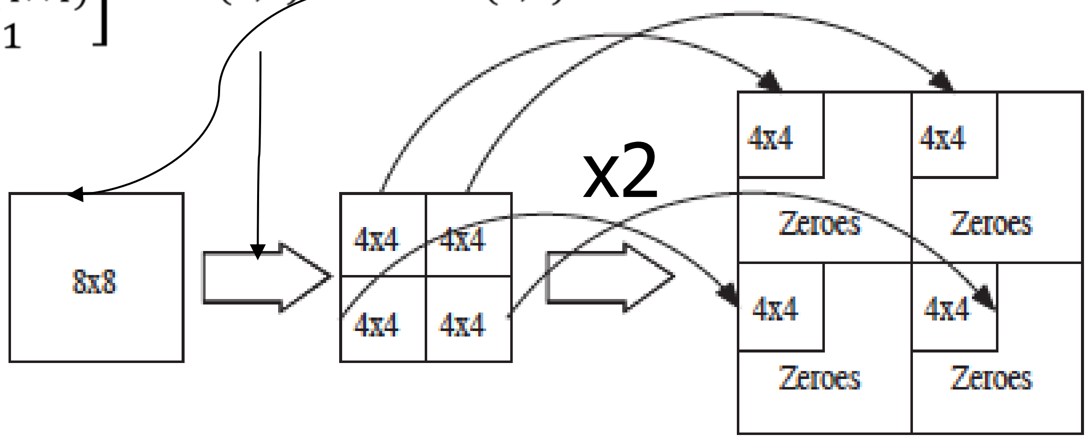
$$X^{(16\times16)} = A_{(2,4)} \begin{bmatrix} X_{00}^{(8\times8)} & X_{01}^{(8\times8)} \\ X_{10}^{(8\times8)} & X_{11}^{(8\times8)} \end{bmatrix} A_{(2,4)}^{T}$$



$$Z = A_{(2,8)} \begin{bmatrix} X_{00} & X_{01} \\ X_{10} & X_{11} \end{bmatrix} A_{(2,8)}^{T}. \qquad Y = \frac{1}{2}\left[Z(k,l)\right] 0 \le k,l \le 7.$$

Image Doubling: Decomposition followed by Approximation (IDDA)

$$\begin{bmatrix} X_{00}^{(4\times4)} & X_{01}^{(4\times4)} \\ X_{10}^{(4\times4)} & X_{11}^{(4\times4)} \end{bmatrix} = A_{(2,4)}^{-1} X^{(8\times8)} A_{(2,4)}^{-T}$$

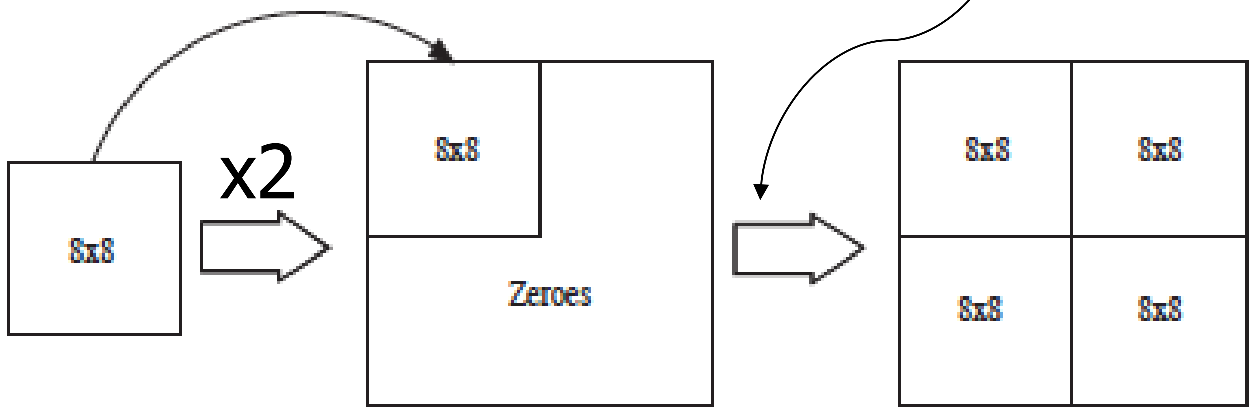


$$\begin{bmatrix} Y_{00}^{d} & Y_{01}^{d} \\ Y_{10}^{d} & Y_{11}^{d} \end{bmatrix} = A_{(2,4)}^{T} X A_{(2,4)}$$

$$Y_{00} = 2 \begin{bmatrix} Y_{00}^{d} & 0_4 \\ 0_4 & 0_4 \end{bmatrix}.$$

Image Doubling: Approximation followed by Decomposition (IDAD)

$$\begin{bmatrix} X_{00}^{(8\times8)} & X_{01}^{(8\times8)} \\ X_{10}^{(8\times8)} & X_{11}^{(8\times8)} \end{bmatrix} = A_{(2,8)}^{-1} X^{(16\times16)} A_{(2,4)}^{-T}$$




TACHYMETER

IDDA


TACHYMETER
60
400
300
250
200
70
1/10 SEC
Chronograph
100 METERS
15
45
MIN
30
10
20
50
40
30

IDAD

Resizing with integral factors

To convert NxN block to LNxMN block.

LN x MN block

NxN DCT block

$$X(k,l) = \begin{cases} \sqrt{LM}X_{LL}(k,l), & k,l = 0,1,\ldots,N-1, \\ 0, & otherwise. \end{cases}$$

LxM D/S (LMDS) 1. Merge LxM adjacent DCT blocks.

$$Z = A_{(L,N)} \begin{bmatrix} X_{0,0} & X_{0,1} & \cdots & X_{0,(M-1)} \\ X_{1,0} & X_{1,1} & \cdots & X_{1,(M-1)} \\ . & . & \cdots & . \\ X_{(L-1),0} & X_{(L-1),1} & \cdots & X_{(L-1),(M-1)} \end{bmatrix} A^T_{(M,N)}$$

2. Sub-band approximation to a NxN DCT block.

$$Y = \sqrt{\frac{1}{LM}}\,[Z(k,l)]_{0 \le k,l \le N-1}$$

LxM U/S (LMUS)

1. Convert NxN to LNxMN block

$$\hat{B} = \begin{bmatrix} \sqrt{LM}B & 0_{(N,(M-1)N)} \\ 0_{((L-1)N,N)} & 0_{((L-1)N,(M-1)N)} \end{bmatrix}$$

Efficiently compute exploiting large blocks of zeroes.

2. Decompose into LxM NxN blocks.

An example: 3x2 D/S and U/S

Arbitrary Resizing (P/Q x R/S)

- U/S-D/S Resizing Algorithm (UDRA)
  - U/S by PxR
  - D/S by QxS
- D/S-U/S Resizing Algorithm (DURA)
  - D/S by QxS
  - U/S PxR

HDTV (1080x920) to NTSC (480x640)

UDRA

DURA

Hybrid Resizing (HRA)

More general sub-band relation

X: DCT block of QNxSN
Y: DCT block of PNxRN

$$
\begin{aligned}
\mathbf{Y} &= \sqrt{\tfrac{PR}{QS}}\,[\mathbf{X}]_{0\le i<PN,0\le j<RN}, & P\le Q,\ R\le S,\\
&= \sqrt{\tfrac{PR}{QS}}\begin{bmatrix}[\mathbf{X}]_{0\le i<QN,0\le j<RN}\\ 0_{(P-Q)N\times RN}\end{bmatrix}, & P>Q,\ R\le S,\\
&= \sqrt{\tfrac{PR}{QS}}\begin{bmatrix}[\mathbf{X}]_{0\le i<PN,0\le j<SN} & 0_{PN\times(R-S)N}\end{bmatrix}, & P\le Q,\ R>S,\\
&= \sqrt{\tfrac{PR}{QS}}\begin{bmatrix}[\mathbf{X}]_{0\le i<QN,0\le j<SN} & 0_{QN\times(R-S)N}\\ 0_{(P-Q)N\times SN} & 0_{(P-Q)N\times(R-S)N}\end{bmatrix}, & P>Q,\ R>S.
\end{aligned}
$$

Truncated DCT block of X or padded with zeroes, if required.

$$\mathbf{Y} = \sqrt{\frac{PR}{QS}}\hat{\mathbf{X}}$$

1. Merge QxS input DCT blocks to form X.
2. Take PxR blocks to form Y with the sub-band approximation.
3. Decompose Y into PxR NxN blocks.

Hybrid Resizing: A few examples

$\frac{2}{3} \times \frac{4}{5}$

$\frac{2}{3} \times \frac{3}{2}$

$\frac{3}{2} \times \frac{5}{4}$

$\frac{3}{2} \times \frac{2}{3}.$

DCT Domain Filtering

Filtering a finite sequence

Input: $x(n), n=0,1, \ldots, N\text{-}1$

Filter: $h(n)$: $n=0,1,2, .., L\text{-}1$

Pad with zeroes at undefined points.

Linear convolution

$$y(n) = \sum_{m=0}^{N-1} x(m)h(n-m), \qquad n = 0,1,..(N+L-1)$$

$$\begin{bmatrix} y(0) \\ y(1) \\ . \\ . \\ y(L-1) \\ . \\ y(N-1) \\ . \\ y(N+L-1) \end{bmatrix} = \begin{bmatrix} h(0) & 0 & \cdots & 0 & \cdots & 0 \\ h(1) & h(0) & \cdots & 0 & \cdots & 0 \\ . & . & \cdots & . & \cdots & . \\ . & . & \cdots & . & \cdots & . \\ h(L-1) & h(L-2) & \cdots & h(0) & \cdots & 0 \\ . & . & \cdots & . & \cdots & . \\ 0 & 0 & \cdots & . & \cdots & h(0) \\ . & . & \cdots & . & \cdots & . \\ 0 & 0 & \cdots & 0 & \cdots & h(L-1) \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ . \\ . \\ x(L-1) \\ . \\ x(N-1) \end{bmatrix}.$$

$$DCT(\mathbf{y}) = DCT(H_l^{(N+L-1)\times N})DCT(\mathbf{x})$$

Filtering with periodic extension of input sequence and response

$x(n+N)=x(n)$

Input: $x(n)$, $n=0,1, \ldots, N\text{-}1$

Periodically extended with period N.

Filter: $h(n)$: $n=0,1,2, \ldots, N\text{-}1$

Circular convolution

$$y(n) = x(n) \circledast h(n) = \sum_{m=0}^{n} x(m)h(n-m) + \sum_{m=n+1}^{N-1} x(m)h(n-m+N).$$

$H_c^{(N\mathrm{x}N)}$

$$\begin{bmatrix} y(0) \\ y(1) \\ . \\ . \\ y(N-1) \end{bmatrix} = \begin{bmatrix} h(0) & h(N-1) & \cdots & h(1) \\ h(1) & h(0) & \cdots & h(2) \\ . & . & \cdots & . \\ . & . & \cdots & . \\ h(N-1) & h(N-2) & \cdots & h(0) \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ . \\ . \\ x(N-1) \end{bmatrix}.$$

Can be made equiv. to. linear conv. by zero padding of $x(n)$ and $h(n)$.

Filtering with antiperiodic extension

$x(n+N)=-x(n)$

Input: $x(n)$, $n=0,1, \ldots, N\text{-}1$

Filter: $h(n)$: $n=0,1,2, .., N\text{-}1$

Antiperiodic extension extended with antiperiod N.

Skew Circular convolution

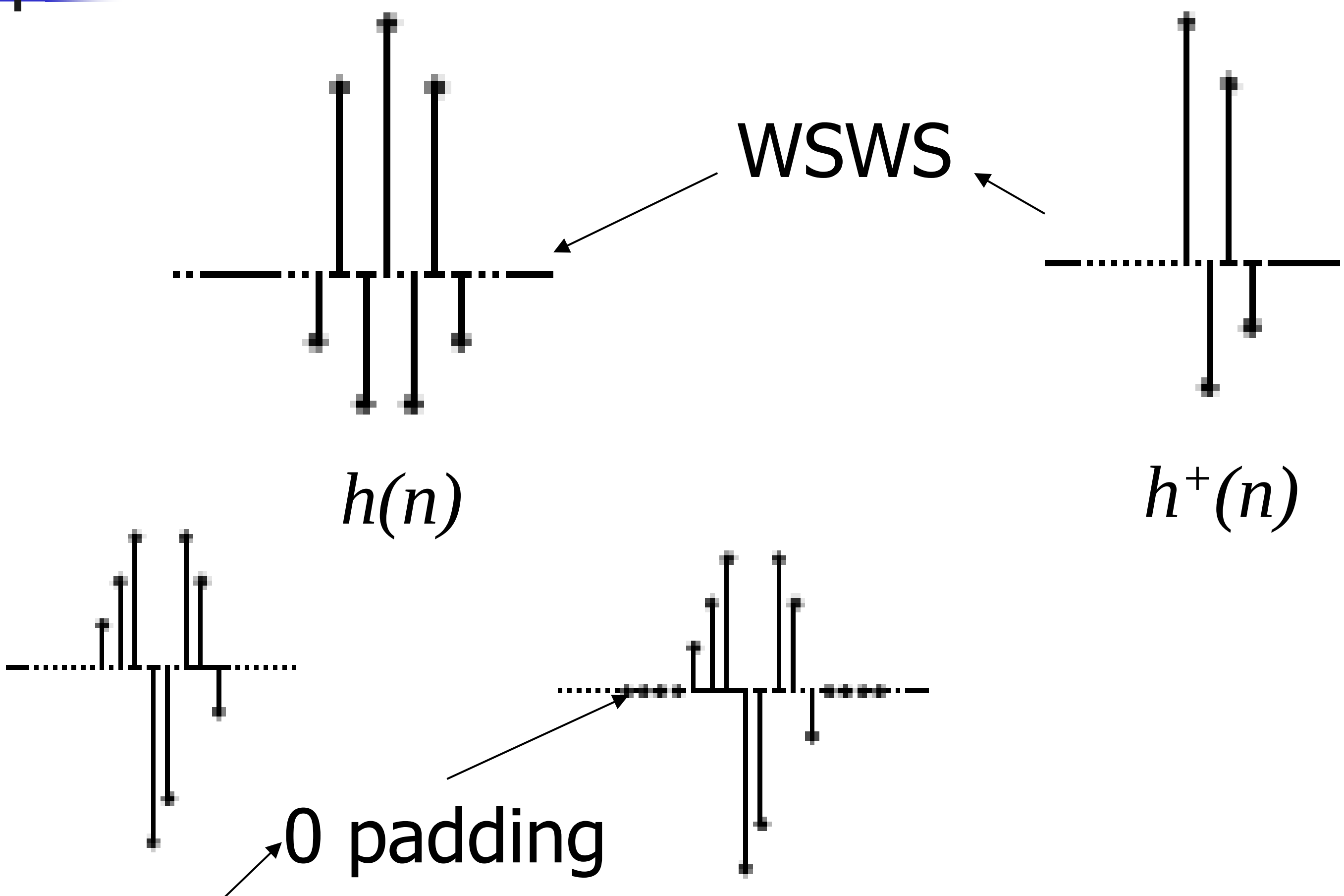
$$y(n) = x(n)Ⓢh(n) = \sum_{m=0}^{n} x(m)h(n-m) - \sum_{m=n+1}^{N-1} x(m)h(n-m+N)$$

$H_s^{(N\text{x}N)}$

$$H_s^{N\times N} = \begin{bmatrix} h(0) & -h(N-1) & \cdots & -h(1) \\ h(1) & h(0) & \cdots & -h(2) \\ . & . & \cdots & . \\ . & . & \cdots & . \\ h(N-1) & h(N-2) & \cdots & h(0) \end{bmatrix}$$

Can be made equiv. to. linear conv. by zero padding of $x(n)$ and $h(n)$.

Symmetric convolution and CMP for type II DCT



$$C_{2e}(y(n)) = \sqrt{2N}\, C_{1e}(h^+(n))\, C_{2e}(x(n))$$

CONVOLUTION-MULTIPLICATION PROPERTY

$$x(n) \longrightarrow \boxed{h(n)} \longrightarrow y(n)=h(n)*x(n)$$

$\Updownarrow$ **•Symmetric convolution operation**

$$X_{II}(k) \longrightarrow \boxed{H_I(k)} \longrightarrow Y_{II}(k)=H_I(k).X_{II}(k)$$

$$Y_{II}^{(N)} = \begin{bmatrix} H_{I,0} & 0 & 0 & 0 \\ 0 & H_{I,1} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & H_{I,N-1} \end{bmatrix} X_{II}^{(N)}$$

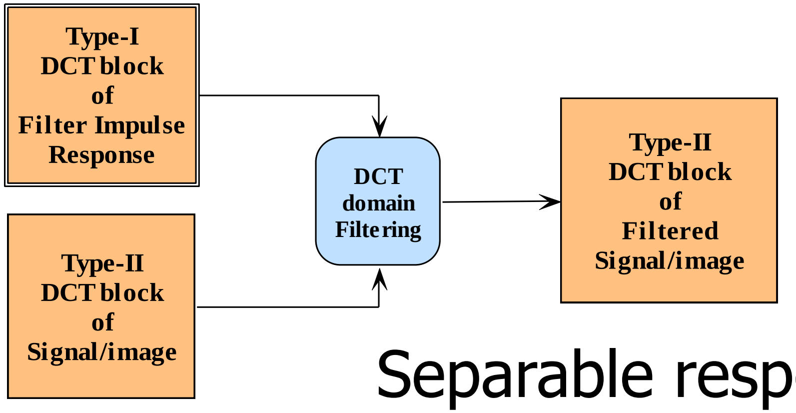
$diag(H_I)$

Diagonal matrix formed by $H_I(k)$

Filtering in 2-D block DCT space

Given, **type-I DCT** of the impulse response of a filter and an input in **type-II DCT space**, filtered output can be transformed in the same space (i.e. type-II DCT space).

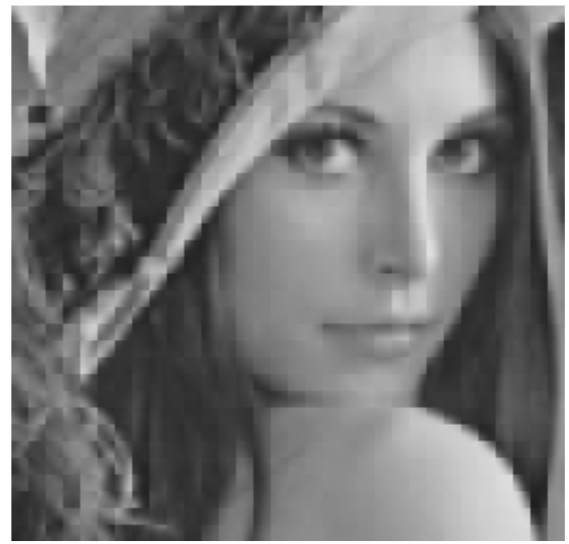
$$
\begin{aligned}
Y_{II}^{(N\times N)} &= (diag(H_I)\left(diag(H_I)X_{II}^{(N\times N)}\right)^T)^T \\
&= diag(H_I)X_{II}^{(N\times N)}diag(H_I)^T \\
&= diag(H_I)X_{II}^{(N\times N)}diag(H_I)
\end{aligned}
$$



Separable response: $h(x,y)=h(x)h(y)$

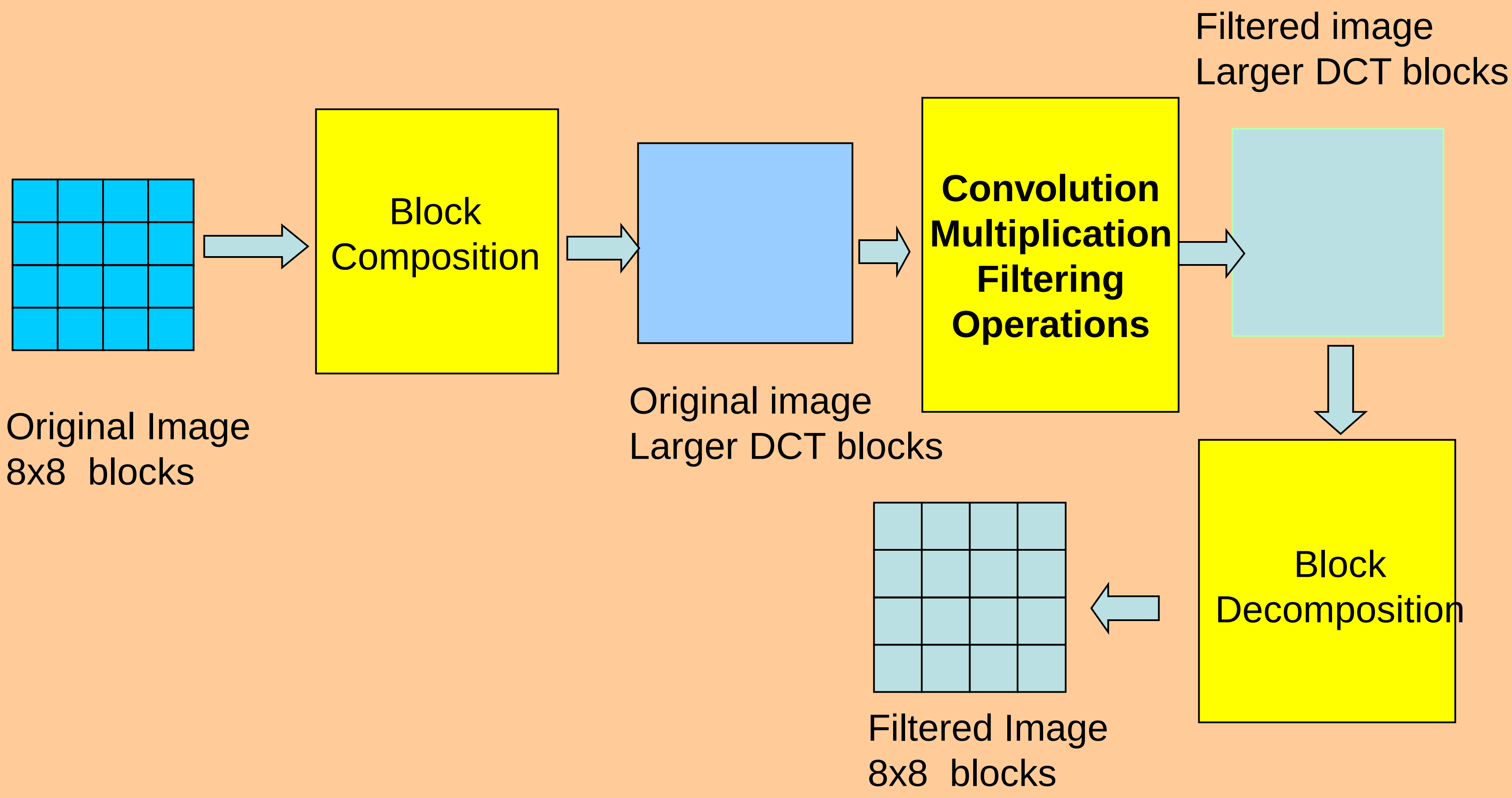
BOUNDARY EFFECT IN BLOCK DCT DOMAIN

Linear Convolution



Block Symmetric Convolution

Computation with larger blocks: The BFCD filtering Algorithm



Composite operation

- Block composition + Multiplication with filter coefficients + Block decomposition can be expressed by a composite matrix for the linear operation.

In 1-D: For filtering 3 NxN adjacent DCT blocks:

$$U^{(3N\text{x}3N)} = A^T_{(3,N)}\mathbb{D}(\{\sqrt{6N}C^I_{3N}\mathbf{h}^+\}_0^{3N-1})A_{(3,N)}$$

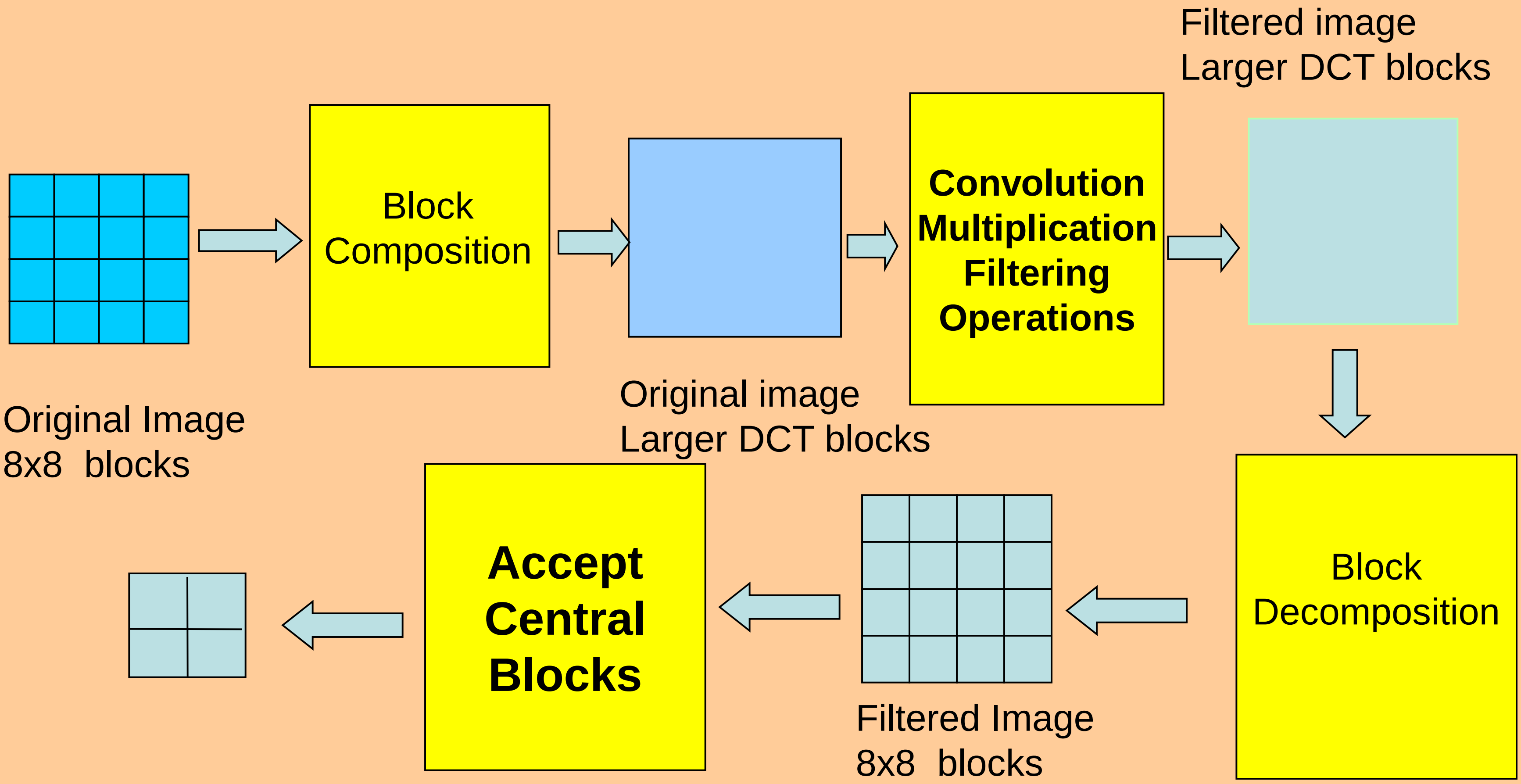
Decomp. Multiplication Composition

$$\begin{bmatrix} Y_1^{(N)} \\ Y_2^{(N)} \\ Y_3^{(N)} \end{bmatrix} = U \begin{bmatrix} X_1^{(N)} \\ X_2^{(N)} \\ X_3^{(N)} \end{bmatrix}$$

In 2-D

$$\begin{bmatrix} Y_{11}^{(N\times N)} & Y_{12}^{(N\times N)} & Y_{13}^{(N\times N)} \\ Y_{21}^{(N\times N)} & Y_{22}^{(N\times N)} & Y_{23}^{(N\times N)} \\ Y_{31}^{(N\times N)} & Y_{32}^{(N\times N)} & Y_{33}^{(N\times N)} \end{bmatrix} = U \begin{bmatrix} X_{11}^{(N\times N)} & X_{12}^{(N\times N)} & X_{13}^{(N\times N)} \\ X_{21}^{(N\times N)} & X_{22}^{(N\times N)} & X_{23}^{(N\times N)} \\ X_{31}^{(N\times N)} & X_{32}^{(N\times N)} & X_{33}^{(N\times N)} \end{bmatrix} U^T$$

Exact Computation: The Overlapping and Save (OBFCD) filtering Algorithm

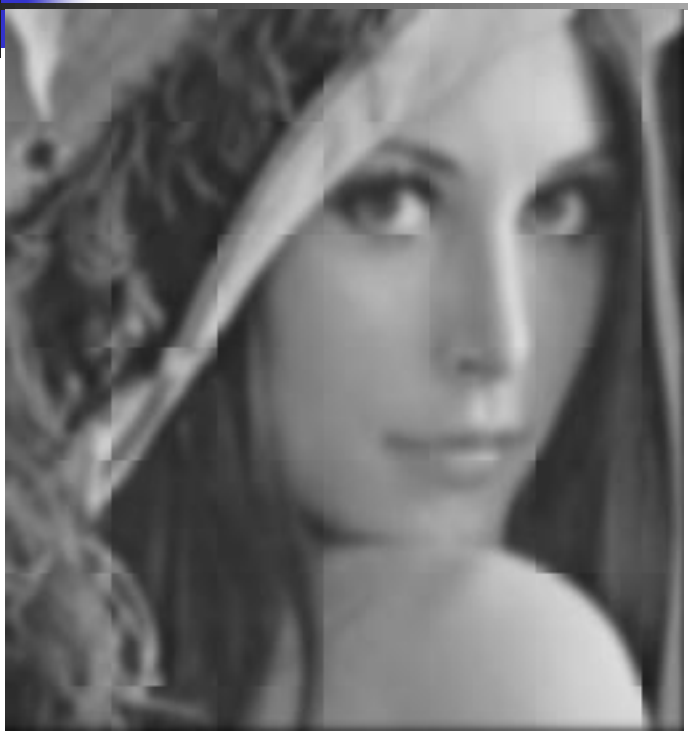


Overlap and Save Strategy

$$\begin{bmatrix} Y_{11}^{(N\times N)} & Y_{12}^{(N\times N)} & Y_{13}^{(N\times N)} \\ Y_{21}^{(N\times N)} & \color{red}{Y_{22}^{(N\times N)}} & Y_{23}^{(N\times N)} \\ Y_{31}^{(N\times N)} & Y_{32}^{(N\times N)} & Y_{33}^{(N\times N)} \end{bmatrix} = U \begin{bmatrix} X_{11}^{(N\times N)} & X_{12}^{(N\times N)} & X_{13}^{(N\times N)} \\ X_{21}^{(N\times N)} & X_{22}^{(N\times N)} & X_{23}^{(N\times N)} \\ X_{31}^{(N\times N)} & X_{32}^{(N\times N)} & X_{33}^{(N\times N)} \end{bmatrix} U^T$$

Save only the central block.

Filtered Images



BFCD Algorithm (5x5)

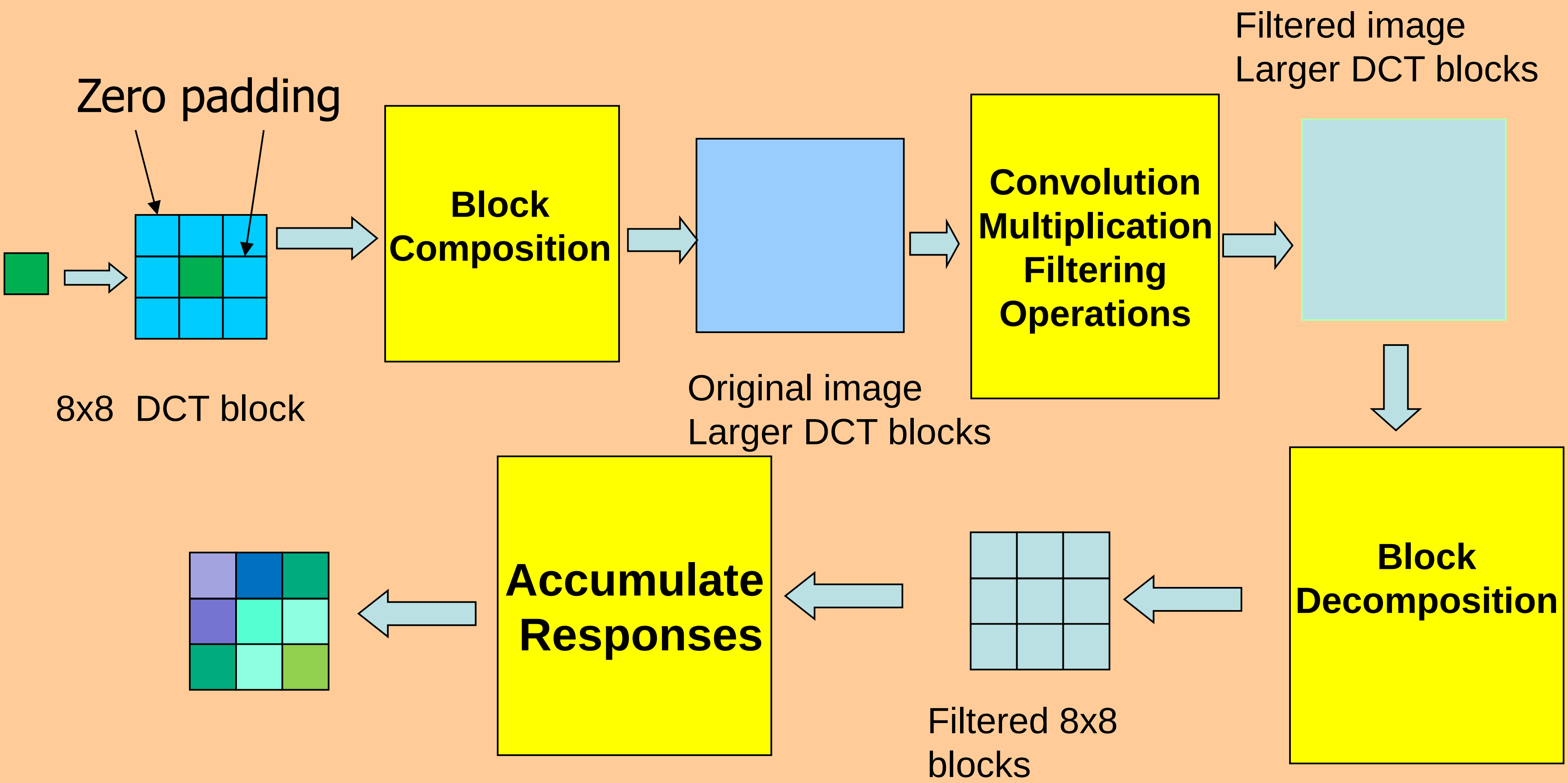
OBFCD Algorithm

Overlap and add strategy

$$\begin{bmatrix} Y_{11}^{(N\times N)} & Y_{12}^{(N\times N)} & Y_{13}^{(N\times N)} \\ Y_{21}^{(N\times N)} & Y_{22}^{(N\times N)} & Y_{23}^{(N\times N)} \\ Y_{31}^{(N\times N)} & Y_{32}^{(N\times N)} & Y_{33}^{(N\times N)} \end{bmatrix} = U \begin{bmatrix} X_{11}^{(N\times N)} & X_{12}^{(N\times N)} & X_{13}^{(N\times N)} \\ X_{21}^{(N\times N)} & {\color{red}X_{22}^{(N\times N)}} & X_{23}^{(N\times N)} \\ X_{31}^{(N\times N)} & X_{32}^{(N\times N)} & X_{33}^{(N\times N)} \end{bmatrix} U^T$$

- For computing contribution of the central block X_{22} to neighboring blocks all other blocks in the input set to zeros.
- Add accumulated contribution at every block.

Exact Computation: The Overlapping and Add filtering Algorithm



Removal of Blocking Artifacts

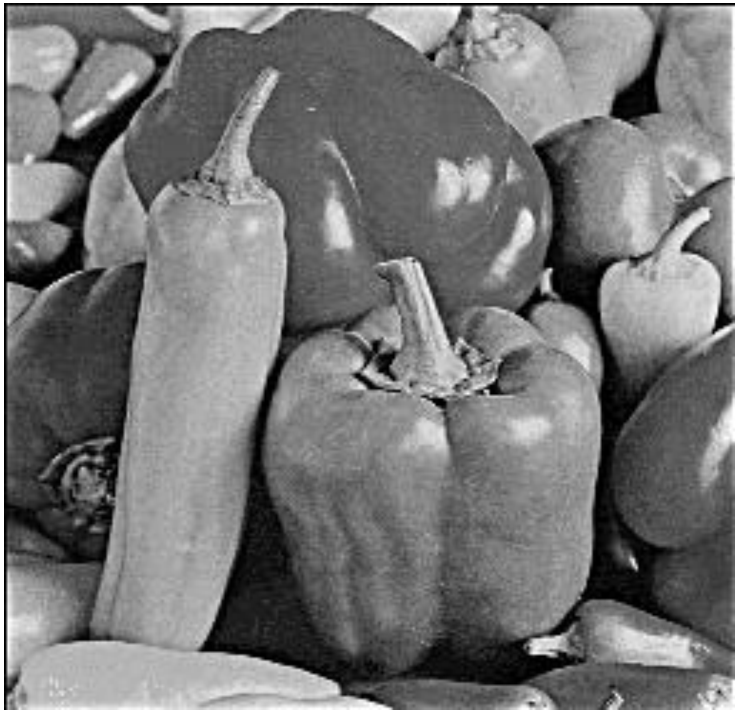
Blocking artifacts
Compressed with quality factor 10

Artifacts removed
By filtering in DCT domain

Image Sharpening in DCT domain

$X_s = X + k.\ (X - X_{lpf})$

Lena

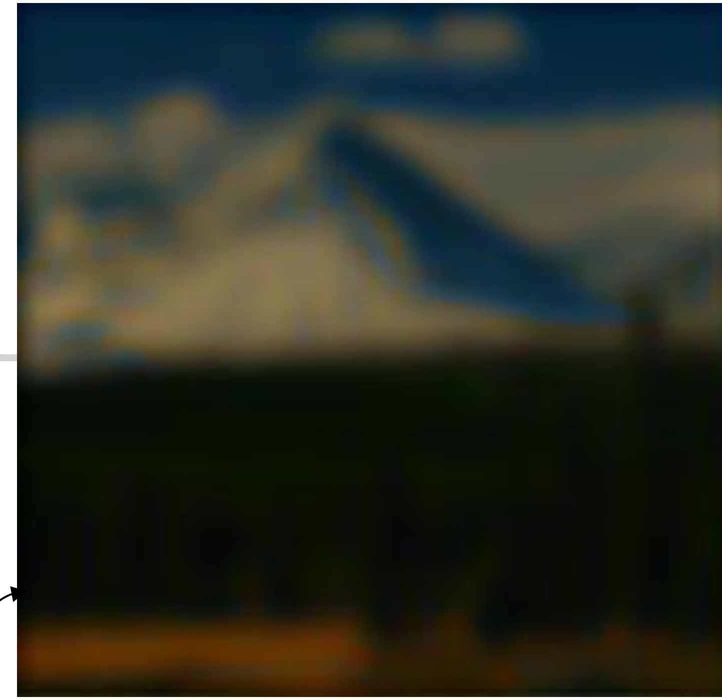


Peppers

Color Sharpening

Original

Filtered



Sharpened

Color Enhancement

$$0 < \alpha < 1$$

$$\tilde{X}(i,j) = X(i,j)\left|\frac{X(i,j)}{Y_{max}}\right|^{\alpha-1}$$

Maximum Luminance

Alpha Rooting

- Smaller values get magnified and larger values get reduced.

Original

AR

Multi-contrast Enhancement (MCE)

- X: 8x8 DCT block: Only AC Coefficients modified.
- Spectral energy: $E_k = \frac{1}{N_k} \Sigma_{i+j=k} |X(i,j)|,$
- Contrast measure: $c_k = \frac{E_k}{\Sigma_{t<k} E_t}.$
- The coefficients of k-th band are scaled so that the resulting contrast becomes $w.c_k$, where w is a constant.
- By scaling coefficients in k-th band with $w.H_k$.

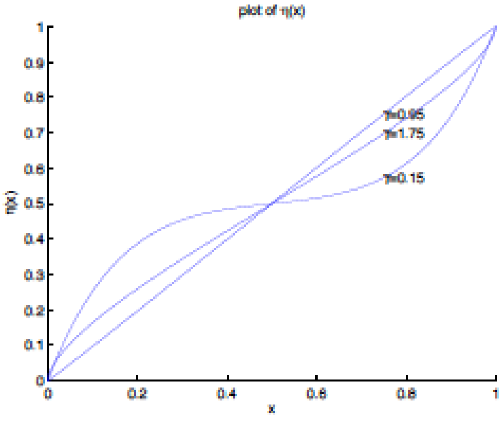
$$H_k = \frac{\Sigma_{t<k} \widetilde{E}_t}{\Sigma_{t<k} E_t},$$

Spectral energy in the enhanced image

MCE with Dynamic Range Compression

- Also modifies DC coefficient, by increasing its dynamic range.
- Expression for spectral energy slightly modified.

$$E_k = \sqrt{\frac{1}{N_k} \Sigma_{i+j=k} |X(i,j)|^2}.$$



$$\eta(x) = \frac{(x^{\frac{1}{\gamma}} + (1 - (1-x)^{\frac{1}{\gamma}}))}{2}, \quad 0 \le x \le 1.$$

Contrast : Definition

Let μ and σ denote the mean and standard deviation of an image. Contrast ζ of an image is defined here as: .

$$\zeta = \frac{\mu}{\sigma}$$

Weber Law: $$\zeta = \frac{\Delta L}{L}$$

where ΔL is the difference in luminance between a stimulus and its surround, and L is the luminance of the surround

Theorem on Contrast Preservation in the DCT Domain

κ_d: the scale factor for the DC coefficient

κ_a : the scale factor for the AC coefficients

$$Y_e(i,j) = \begin{cases} \kappa_d Y(i,j), & i=j=0 \\ \kappa_a Y(i,j), & otherwise \end{cases}$$

The contrast of the processed image : κ_a / κ_d times of the contrast of the original image.

$\kappa_d = \kappa_a = \kappa$ preserves the contrast.

Preservation of Colors in the DCT Domain

U , V : Blocks of DCT coefficients of C_b and C_r

κ :Scale factor for the luminance component Y

$$U_e(i,j) = \begin{cases} N(\kappa\ (\frac{U(i,j)}{N} - 128)) + 128, & i = j = 0 \\ \kappa U(i,j), & otherwise \end{cases}$$

$$V_e(i,j) = \begin{cases} N(\kappa\ (\frac{V(i,j)}{N} - 128)) + 128, & i = j = 0 \\ \kappa V(i,j), & otherwise \end{cases}$$

Color Enhancement by Scaling Coefficients (CES)

- Find the scale factor by mapping the DC coefficient with a monotonically increasing function.

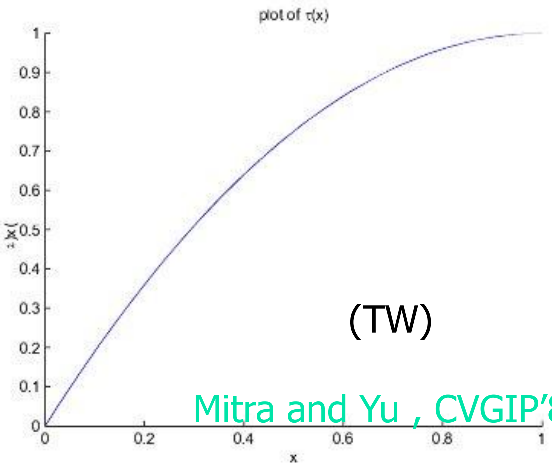
$$1 \leq \kappa \leq \frac{B_{max}}{\mu + \lambda.\sigma}$$

Max. intensity

Scale factor

- Apply scaling to all other coefficients in all the components.
- For blocks having greater details judged by s.d., apply block decomposition and re-composition strategy.

Mapping functions for adjusting the local background illumination

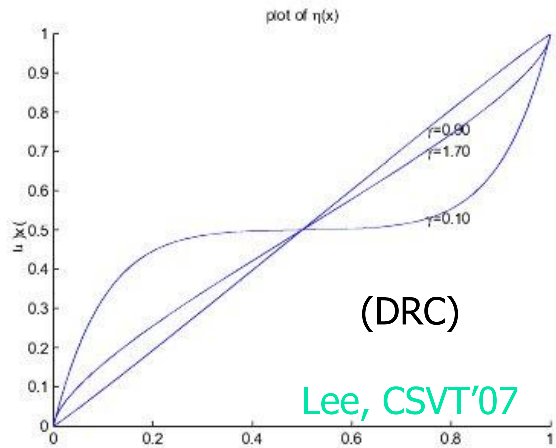


(TW)

Mitra and Yu , CVGIP'87

$$\tau(x) = x(2-x)$$

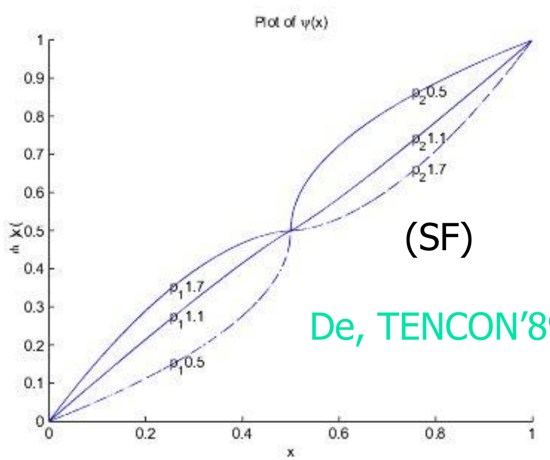
$$\kappa = \frac{f(Y(0,0))}{Y(0,0)}$$



(DRC)

Lee, CSVT'07

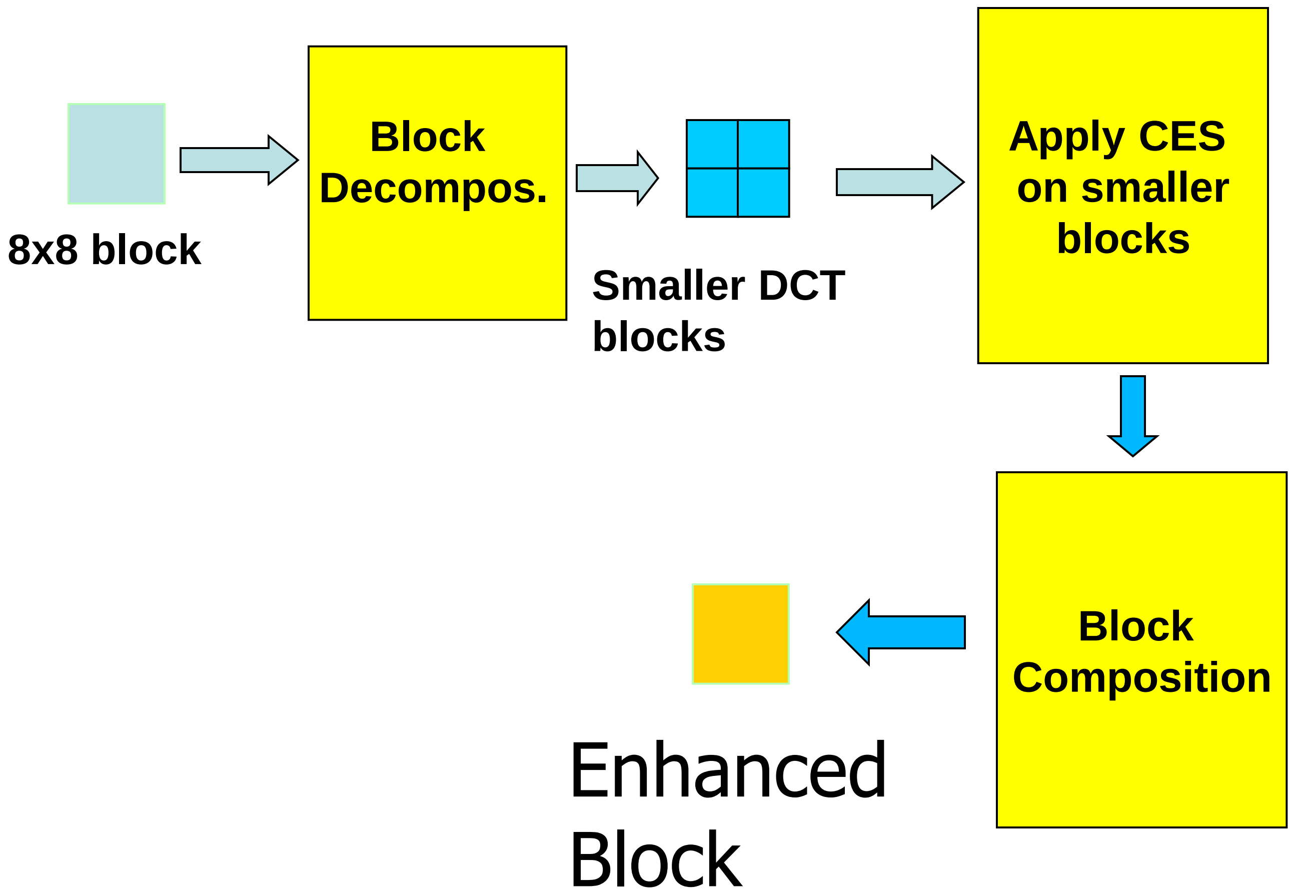
$$\eta(x) = \frac{(x^{\frac{1}{\gamma}} + (1-(1-x)^{\frac{1}{\gamma}}))}{2}, \quad 0 \leq x \leq 1.$$



(SF)

De, TENCON'89

Enhancement of Blocks with more details



MCE

MCEDRC

CES

Some Results

original

AR

MCE

MCEDRC

TW-CES-BLK

MSR

Iterative Enhancement

Iteration no.=1

Run CES iteratively.

original

Iteration no.=3

Iteration no.=2

Iteration no.=4

Color constancy

- Computation of color of illuminant
  - Avg. of colors (Gray world)
  - Maximum of each channel (White world)
  - Use of DC coefficient of DCT blocks in compressed domain.
- Diagonal Color correction: $(R_s, G_s, B_s) \rightarrow (R_d, G_d, B_d)$

$$
\begin{array}{llllll}
k_r = \frac{R_d}{R_s}, & k_g = \frac{G_d}{G_s}, & k_b = \frac{B_d}{B_s}, \\
& f = \frac{R+G+B}{k_r R + k_g G + k_b B}, & \\
R_u = f k_r R, & G_u = f k_g G, & B_u = f k_b B.
\end{array}
$$

- Chromatic Shift in Y-Cb-Cr:

$$
\begin{aligned}
Y_u &= Y, \\
C_{bu} &= C_b + C_{bd} - C_{bs}, \\
C_{ru} &= C_r + C_{rd} - C_{rs}
\end{aligned}
$$

Color correction: Example

Original

Diagonal Correction

Chromatic Shift

Color constancy coupled enhancement

- Perform color correction
- Perform color enhancement

Enhancement without color correction

Thank you!