Color image processing in the compressed domain

Jayanta Mukhopadhyay
Dept. of Computer Science and Engg.
Indian Institute of Technology, Kharagpur
Processing with compressed image: Compressed domain approach

Color encoding in JPEG

- Y-Cb-Cr color space:

\[
Y = 0.502G + 0.098B + 0.256R, \\
Cb = -0.290G + 0.438B - 0.148R + 128, \\
Cr = -0.366G - 0.071B + 0.438R + 128. 
\]
Motivations

- Computation with reduced storage.
- Avoid overhead of forward and reverse transform.
- Exploit spectral factorization for improving the quality of result and speed of computation.
- Color Image Processing in DCT domain.
  - Filtering
  - Enhancement
  - Color constancy
  - Resizing
2D DCT

- **Type-I Even:**
  
  \[ C_N^I = \left[ \sqrt{\frac{2}{N}} \beta(k) \cos\left(\frac{\pi k n}{N}\right) \right]_{0 \leq (k,n) \leq N} \]

- **Type-II Even:**
  
  \[ C_N^{II} = \left[ \sqrt{\frac{2}{N}} \alpha(k) \cos\left(\frac{\pi k (2n + 1)}{2N}\right) \right]_{0 \leq (k,n) \leq N-1} \]

- **Type-I DCT of** \( h(m,n) \):
  
  \[ H = C_N^I h C_N^{IT} \]

- **Type-II DCT of** \( x(m,n) \):
  
  \[ X = C_N^{II} x C_N^{IT} \]

\[
\alpha(k) = \begin{cases} 
\sqrt{\frac{1}{2}} & k = 0 \\
1 & \text{otherwise}
\end{cases} \quad \beta(k) = \begin{cases} 
\frac{1}{2} & k = 0, N \\
1 & \text{Otherwise}
\end{cases}
\]
Useful properties of DCT blocks
2D DCT: Sub-band relation

\[ x_{LL}(m,n) = \frac{1}{4} \{ x(2m, 2n) + x(2m + 1, 2n) + x(2m, 2n + 1) + x(2m + 1, 2n + 1) \}, \quad 0 \leq m, n \leq \frac{N}{2} - 1. \]

Sub-band approximation:

2D DCT of \( x_{LL}(m,n) \)

\[
X(k, l) = \begin{cases} 
2\cos\left(\frac{\pi k}{2N}\right)\cos\left(\frac{\pi l}{2N}\right)X_{LL}(k, l), & k, l = 0, 1, \ldots, \frac{N}{2} - 1 \\
0, & \text{otherwise.}
\end{cases}
\]

Low-pass truncated approximation:

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

Image downsampling

Sub-band approximation

Image upsampling

Sub-band approximation

\[
\begin{array}{cc}
4x4 & 4x4 \\
4x4 & 4x4 \\
\end{array}
\]

\[
\begin{array}{cccc}
4x4 & 0 & 4x4 & 0 \\
0 & 0 & 0 & 0 \\
4x4 & 0 & 4x4 & 0 \\
0 & 0 & 0 & 0 \\
\end{array}
\]

\[
\begin{array}{cc}
8x8 & 8x8 \\
8x8 & 8x8 \\
\end{array}
\]
2D DCT: Block composition and decomposition

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

Block composition and decomposition

\[
A_{(2,4)} = C_8 \cdot \begin{bmatrix}
C_4^{-1} & 0_4 \\
0_4 & C_4^{-1}
\end{bmatrix}
\]

\[
= \begin{bmatrix}
0.7071 & 0 & 0 & 0.7071 & 0 & 0 & 0 & 0 \\
0.6407 & 0.294 & -0.0528 & 0.0162 & -0.6407 & 0.294 & 0.0528 & 0.0162 \\
0 & 0.7071 & 0 & 0 & 0 & 0.7071 & 0 & 0 \\
-0.225 & 0.5594 & 0.3629 & -0.0690 & 0.225 & 0.5594 & -0.3629 & -0.0690 \\
0 & 0 & 0.7071 & 0 & 0 & 0 & 0.7071 & 0 \\
0.1503 & -0.2492 & 0.5432 & 0.3468 & -0.1503 & -0.2492 & -0.5432 & 0.3468 \\
0 & 0 & 0 & 0.7071 & 0 & 0 & 0 & 0.7071 \\
-0.1274 & 0.1964 & -0.2654 & 0.6122 & 0.1274 & 0.1964 & 0.2654 & 0.6122
\end{bmatrix}
\]

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

\[
\begin{bmatrix}
X_{00}^{(4 \times 4)} \\
X_{01}^{(4 \times 4)} \\
X_{10}^{(4 \times 4)} \\
X_{11}^{(4 \times 4)}
\end{bmatrix} = A_{(2,4)}^{-1} X^{(8 \times 8)} A_{(2,4)}^{-T}
\]

4x4 \quad 4x4

4x4 \quad 4x4

8x8
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

\[ x(n) \rightarrow h(n) \rightarrow y(n) = h(n) \ast x(n) \]

\[ X_{II}(k) \rightarrow H_I(k) \rightarrow Y_{II}(k) = H_I(k) \cdot X_{II}(k) \]

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

Diagonal matrix formed by \( H_I(k) \)

Symmetric convolution operation
2D DCT: CMPs

Circular convolution with respective symmetric extensions.

\[ C_{2e}\{x(m, n) \ast h(m, n)\} = C_{2e}\{x(m, n)\}C_{1e}\{h(m, n)\} \]
\[ C_{1e}\{x(m, n) \ast h(m, n)\} = C_{2e}\{x(m, n)\}C_{2e}\{h(m, n)\} \]

DCT Domain Filtering
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).

\[
Y_{II}^{(N \times N)} = \left( \text{diag}(H_I) \left( \text{diag}(H_I) X_{II}^{(N \times N)} \right)^T \right)^T
\]

\[
= \text{diag}(H_I) X_{II}^{(N \times N)} \text{diag}(H_I)^T
\]

\[
= \text{diag}(H_I) X_{II}^{(N \times N)} \text{diag}(H_I)
\]

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

Original Image
8x8 blocks

Block Composition

Original image
Larger DCT blocks

Convolution Multiplication Filtering Operations

Filtered Image
8x8 blocks

Filtered image
Larger DCT blocks

Block Decomposition

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:

\[
U^{(3N \times 3N)} = A^T_{(3,N)} \mathcal{D}\left(\{\sqrt{6N}C_{3N}^I h^+\}_{0}^{3N-1}\right) A_{(3,N)} \cdot
\]

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

Decomp. Multiplication Composition
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

Original Image
8x8 blocks

Filtering Operations

Convolution
Multiplication

Accept Central Blocks

Filtered Image
8x8 blocks

Block Composition

Original image
Larger DCT blocks

Block Decomposition

Larger DCT blocks

Overlap and Save Strategy

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

Save only the central block.
Filtered Images

BFCD Algorithm (5x5)  
OBFCD Algorithm

Overlap and add strategy

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

- 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

J. Mukhopadhyay. Isolating neighbor’s contribution towards image filtering in the block DCT space. In IEEE Int. Conf. on Image Processing (ICIP), Hong Kong, Sept. 25-30 2010. IEEE.
Removal of Blocking Artifacts

Blocking artifacts
Compressed with quality factor 10

Artifacts removed
By filtering in DCT domain

Image Sharpening in DCT domain

\[ X_s = X + k \cdot (X - X_{lpf}) \]

Lena

Peppers

Color preservation in the compressed domain

- R-G-B to Y-Cb-Cr : Affine
- R-G-B to Y-Cb’-Cr’, where Cb’=Cb-128 & Cr’=Cr-128: Linear
- DC of a Y-block (Y_{dc}) scaled by a factor k.
- Corresponding Cb’_{dc} and Cr’_{dc} should also be scaled by a factor k for color preservation.
- Transform Cb’_{dc} and Cr’_{dc} to Cb_{dc} and Cr_{dc} then by simple addition.
  - Cb_{dc} = Cb’_{dc} + 128N
  - Cr_{dc} = Cr’_{dc} + 128N
Color Sharpening

Original

Filtered

Sharpened
Color Saturation and Desaturation
The Commission Internationale de l’Eclairage (estd. 1931) defined 3 hypothetical additive primaries: X, Y, Z

\[
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix} = \begin{bmatrix}
0.6067 & 0.1736 & 0.2001 \\
0.2988 & 0.5868 & 0.1143 \\
0.0000 & 0.0661 & 1.1149
\end{bmatrix} \begin{bmatrix}
R \\
G \\
B
\end{bmatrix}
\]

Normalized \(x-y\) space:
\[
x = \frac{X}{X+Y+Z}
\]
\[
y = \frac{Y}{X+Y+Z}
\]
Saturation and De-saturation Operation

- Move radially to the gamut edge → Maximum Saturation given a hue.
- Move inward using center of gravity law of color mixing.

Luca Lucchese, SK Mitra, J Mukherjee, A new algorithm based on saturation and desaturation in the xy chromaticity diagram for enhancement and re-rendition of color images, ICIP 2001.
Desaturation using Center of Gravity Law

The Center of Gravity Law provides the resulting color \( C_2 = (x_2, y_2, Y_2) \) of the mixture of the two colors \( W = (x_w, y_w, |Y_w|) \) and \( S = (x_s, y_s, Y_1) \) where

\[
x_2 = \frac{x_w \frac{|Y_w|}{y_w} + x_s \frac{Y_1}{y_s}}{|Y_w| \frac{1}{y_w} + Y_1 \frac{1}{y_s}}, \quad y_2 = \frac{|Y_w| + Y_1}{|Y_w| \frac{1}{y_w} + Y_1 \frac{1}{y_s}}, \quad \text{and} \quad Y_2 = Y_w + Y_1.
\]  

\[
Y_W = \kappa Y_{avg}
\]

Apparent masses for chromatic mixture: \( Y_w/y_w \) and \( Y_1/y_s \).
Desaturation using Center of Gravity Law

\[ W = (x_w, y_w, Y_w) \]

\[ D = (x_d, y_d, Y_d) \]

\[ S = (x_s, y_s, Y_s) \]

\[ x_d = x_w \frac{|Y_w|}{y_w} + x_s \frac{|Y_s|}{y_s} \]

\[ y_d = \frac{|Y_w| + Y_s}{y_w} + \frac{|Y_s|}{y_s} \]

\[ Y_d = |Y_w| + Y_s \]

\[ Y_w = kY_{avg} \]

**nCb-nCr color space**

\[
\begin{align*}
nCb &= \frac{Cb - 128}{aY + b(Cb - 128) + c(Cr - 128)} \\
nCr &= \frac{Cr - 128}{aY + b(Cb - 128) + c(Cr - 128)}
\end{align*}
\]

\[
\begin{bmatrix} nCb \\ nCr \end{bmatrix} = \begin{bmatrix} -0.6823 & -0.7724 \\ 1.532 & -0.6047 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0.4849 \\ -0.3091 \end{bmatrix}
\]

\[
\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -0.3789 & 0.484 \\ -0.96 & -0.4275 \end{bmatrix} \begin{bmatrix} nCb \\ nCr \end{bmatrix} + \begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \end{bmatrix}
\]

\[a = 3.51, \quad b = 1.99, \quad c = 0.14\]

Cb-Cr from nCb-nCr given Y

\[
\begin{bmatrix}
Cb_s \\
Cr_s
\end{bmatrix} = a. Y \begin{bmatrix}
1 - b.nCb & -c.nCb \\
-b.nCr & 1 - c.nCr
\end{bmatrix}^{-1} \begin{bmatrix}
nCb \\
nCr
\end{bmatrix} + \begin{bmatrix}
128 \\
128
\end{bmatrix}.
\]
Saturation and De-saturation Operation in nCbCr space

Chromatic Shift in Cb-Cr space

- The ratio of (Cb-128) and (Cr-128) preserves the hue.

- Shift the (Cb,Cr) point along the line joining from (128,128) to it.

- Equivalent to saturation and desaturation operation.

Color Enhancement
Contrast : Definition

Weber Law: \( \zeta = \frac{\Delta L}{L} \)

where \( \Delta L \) is the difference in luminance between a stimulus and its surround, and \( L \) is the luminance of the surround.

\( \mu \): mean of a block \( \leftrightarrow \) DC coefficient of the block

\( \sigma \): standard deviation of a block \( \leftrightarrow \) Sum of AC coefficients

Contrast \( \zeta \) of an image is defined here as:

\( \zeta = \frac{\sigma}{\mu} \)
Theorem on Contrast Preservation in the DCT Domain

\( \kappa_d \): the scale factor for the DC coefficient
\( \kappa_a \): the scale factor for the AC coefficients

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

The contrast of the processed image: \( \kappa_a / \kappa_d \) 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 $C_r$

$\kappa$: Scale factor for the luminance component $Y$

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

$$V_e(i, j) = \begin{cases} 
N(\kappa \frac{V(i, j)}{N} - 128)) + 128, & i = j = 0 \\
\kappa V(i, j), & \text{otherwise}
\end{cases}$$
Color Enhancement by Scaling Coefficients (CES)

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

\[ 1 \leq \kappa \leq \frac{B_{\text{max}}}{\mu + \lambda \sigma} \]

- 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

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

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

\( \eta(x) = \frac{(x^{\frac{1}{7}} + (1 - (1 - x)^{\frac{1}{7}}))}{2}, \quad 0 \leq x \leq 1. \)

Mitra and Yu, CVGIP’87

Lee, CSVT’07

De, TENCON’89
Enhancement of Blocks with more details

8x8 block → Block Decompos. → Smaller DCT blocks → Apply CES on smaller blocks → Enhanced Block → Block Composition
Some Results

original

AR

MCE

MCEDRC

TW-CES-BLK

MSR
Iterative Enhancement

Run CES iteratively.

Iteration no.=1

Iteration no.=2

Iteration no.=3

Iteration no.=4

original
Color Constancy
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: \((R_s, G_s, B_s) \rightarrow (R_d, G_d, B_d)\)
  \[\begin{align*}
  k_r &= \frac{R_d}{R_s}, \\
  k_g &= \frac{G_d}{G_s}, \\
  f &= \frac{R + G + B}{k_r R + k_g G + k_b B}, \\
  R_u &= f k_r R, \\
  G_u &= f k_g G, \\
  B_u &= f k_b B.
  \end{align*}\]
- Chromatic Shift in Y-Cb-Cr:
  \[\begin{align*}
  Y_u &= Y, \\
  C_{bu} &= C_b + C_{bd} - C_{bs}, \\
  C_{ru} &= C_r + C_{rd} - C_{rs}
  \end{align*}\]
Color correction: Example

Original     Diagonal Correction     Chromatic Shift

Color constancy coupled enhancement

- Perform color correction
- Perform color enhancement

Enhancement without color correction
Color Image Resizing
Resizing with integral factors

To convert \( N \times N \) block to \( L \times M \times N \) block.

**LxM D/S (LMDS)**

1. Merge \( L \times M \) adjacent DCT blocks.

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

2. Sub-band approximation to a \( N \times N \) DCT block.

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

LMDS

LxM U/S (LMUS)

1. Convert NxN to LNxMN block

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

2. Decompose into LxM NxN blocks.

Efficiently compute exploiting large blocks of zeroes.
An example:
3x2 D/S and U/S
Arbitrary Resizing (P/R x Q/S)

- U/S-D/S Resizing Algorithm (UDRA)
  - U/S by PxQ
  - D/S by RxS
- D/S-U/S Resizing Algorithm (DURA)
  - D/S by RxS
  - U/S PxQ
HDTV (1080x920) to NTSC (480x640)
Issues involved in color resizing

- Baseline JPEG Compression: Usually the chromatic components Cb and Cr are at lower resolution than the Y component.

- Cascaded stages of down-sampling and up-sampling (the DURA algorithm) faces a problem of dimensionality mismatch.

- Appearance of color artifacts in boundary.
Lighthouse (original)
3/4 x 4/3 Resizing

DURA

UDRA
Useful properties of DCT exploited in developing algorithms.

- Linear and distributive property.
- Sub-band relationship
- Spatial relationship
- Convolution-Multiplication properties.

Processing of DC coefficients adapts spatial domain algorithm in the DCT domain.

Processing of AC coefficients also required to handle details, etc.
Conclusion

- Color space in compressed domain Y-Cb-Cr
  - Affine transformation from R-G-B space.
  - Usually Cb and Cr blocks are down-sampled.
- Processing in chromatic space required to preserve color vector, if luminance component (Y) is modified.
Thank you!