

マルチメディア工学

マルチメディアデータの解析

データ圧縮：離散コサイン変換とJPEG

佐藤 嘉伸

大阪大学 大学院医学系研究科
放射線統合医学講座
yoshi@image.med.osaka-u.ac.jp
<http://www.image.med.osaka-u.ac.jp/member/yoshi/>
講義ホームページ：日本語ページ → 授業の資料 → マルチメディア工学

マルチメディア工学：講義計画

- イントロダクション
- コンピュータグラフィックス (Computer Graphics: **CG**)
- マルチメディアデータの解析

マルチメディア工学：講義計画

- イントロダクション
- コンピュータグラフィックス (Computer Graphics: **CG**)
- **マルチメディアデータの解析**
 - 基礎数理
 - 代表的解析手法
 - データ圧縮：離散コサイン変換・JPEG
 - データ表現：形状の主成分分析
 - (データ認識：隠れマルコフモデル)
 - (音声の独立成分分析)

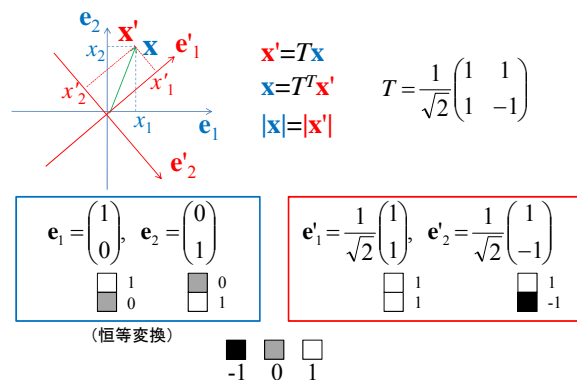
マルチメディアデータの解析

- 基礎数理
 - 最小二乗法
 - 直交変換、直交関数展開
- 代表的解析手法
 - データ圧縮：離散コサイン変換・JPEG
 - データ表現：形状の主成分分析
 - (データ認識：隠れマルコフモデル)
 - 音声・言語を含む時系列データに対して有効
 - (音声の独立成分分析)

データ圧縮：JPEG

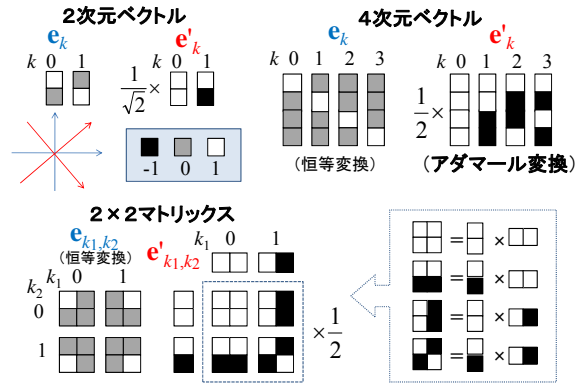
- JPEG圧縮の原理
 - 離散コサイン変換(DCT)
 - アダマール変換
 - 8×8離散コサイン変換
 - 離散フーリエ変換との比較
 - JPEG方式
 - 概要
 - 画像分割→離散コサイン変換→量子化→エントロピー符号化
 - 量子化
 - エントロピー符号化(ハフマン符号化)

直交変換：正規直交基底

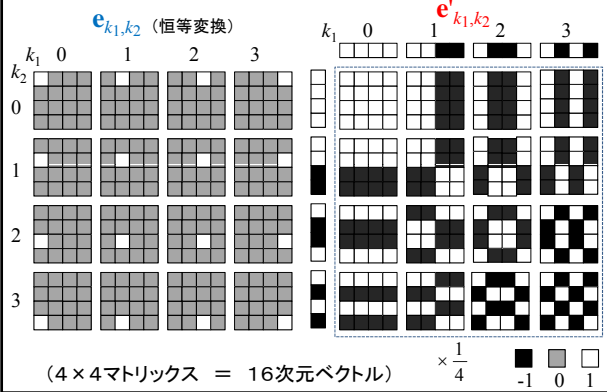


離散コサイン変換

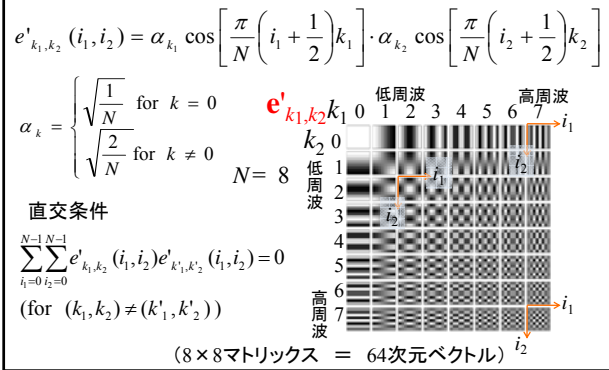
直交変換: 正規直交基底の例



アダマール変換の直交基底



離散コサイン変換の直交基底 (DCT: Discrete Cosine Transform)



離散コサイン変換と 離散フーリエ変換の比較

離散コサイン変換が用いられる理由

離散コサイン変換の目的

- 画像を、空間領域(直交基底 e_{k_1, k_2} の係数)から周波数領域(直交基底 e'_{k_1, k_2} の係数)に変換して、情報源のエントロピーを小さくする。(直交基底 e'_{k_1, k_2} が三角関数系のと き、変換後の空間は周波数領域と呼ばれる。)
- 空間領域の画像では、どの画素値も同じビット数で表現される。これは、どの画素においても、とりうる画素値の確率分布が同じと仮定しなければならないからである。→ エントロピーが大きい。
- 周波数領域の画像では、一部の画素(低周波成分)を除いて、とりうる画素値の確率分布幅を大幅に小さくできる。→ エントロピーを小さくできる。



空間領域



周波数領域

周波数領域では、真中あたり(低周波成分)を除いて、0に近い値のみをとる確率が極めて高くなる。→ エントロピーが小さくなる。

離散コサイン変換の目的

- 情報源符号化の際、平均符号長は、情報源のエントロピーより小さくすることはできない。すなわち、エントロピーを小さくすれば、圧縮効果は高まる。よって、エントロピーの低減はキーポイントである。
- 離散コサイン変換も、離散フーリエ変換も、画像を空間領域から周波数領域に変換するという点では同じである。
- しかし、離散コサイン変換は、離散フーリエ変換より、エントロピーを小さくする、すなわち、低周波成分への集中度を高めることができる。それは、なぜか？
- その理由を理解するには、それぞれの連続空間における数学的意味の検討が必要である。

フーリエ変換

- フーリエ変換

$$F(u) = \int_{-\infty}^{\infty} f(x)e^{-iux} dx = \int_{-\infty}^{\infty} f(x)\{\cos(ux) - i\sin(ux)\} dx$$

- 関数 $f(x)$ と直交関数系である三角関数系の複素数表記 $e^{-iux} = \cos(ux) - i\sin(ux)$ との内積をとる。
- 直交変換後の関数は、周波数 u の関数 $F(u)$ となる。
- $F(u)$ の実数(Re)成分が \cos 成分、虚数(Im)成分が \sin 成分がとなる。
- x が実数の場合、 $F(u) = F^*(-u)$ である。(*は複素共役)



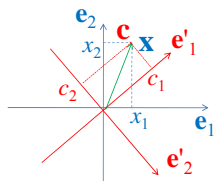
フーリエ変換・展開の整理

- (連続関数の)フーリエ変換
 - 入力(空間) $f(x)$: **連続・無限** (定義域 $-\infty$ から ∞ (無限区間) の連続関数)
 - 出力(周波数) $F(u)$: **連続・無限** (定義域 $-\infty$ から ∞ (無限区間) の連続関数)
- フーリエ級数展開 (周期的連続関数のフーリエ変換)
 - 入力(空間) $f(x)$: **連続・周期** (周期的連続関数)
 - 出力(周波数) c_n : **離散・無限** (離散値の無限系列)
- 離散フーリエ変換 (有限長の標本値系列のフーリエ変換)
 - 入力(空間) x_n : **離散・有限** (離散値の有限系列)
 - 出力(周波数) c_n : **離散・有限** (離散値の有限系列)

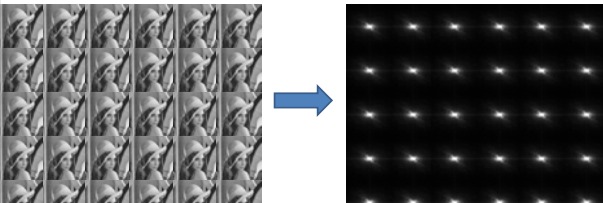
フーリエ変換・展開の整理

- (連続関数の)フーリエ変換
 - 入力(空間) $f(x)$: **連続・無限** (定義域 $-\infty$ から ∞ (無限区間) の連続関数)
 - 出力(周波数) $F(u)$: **連続・無限** (定義域 $-\infty$ から ∞ (無限区間) の連続関数)
- フーリエ級数展開 (周期的連続関数のフーリエ変換)
 - 入力(空間) $f(x)$: **連続・周期** (周期的連続関数)
 - 出力(周波数) c_n : **離散・無限** (離散値の無限系列)
- 離散フーリエ変換 (有限長の標本値系列のフーリエ変換)
 - 入力(空間) x_n : **離散・有限(周期)** (離散値の有限系列)
 - 出力(周波数) c_n : **離散・有限(周期)** (離散値の有限系列)

離散フーリエ変換の連続空間における数学的意味

- デジタル信号のフーリエ変換(離散フーリエ変換)
 - 入出力: **離散・有限** (離散値の有限系列)
- 空間領域 $\mathbf{x} = x_{i_1 i_2}$  \rightarrow  周波数領域 $\mathbf{c} = c_{k_1 k_2}$
- $\mathbf{e}'_{k_1 k_2}$ は三角関数系
- 
- $\mathbf{c} = T\mathbf{x}$
 $\mathbf{x} = T^T\mathbf{c}$
 $|\mathbf{x}| = |\mathbf{c}|$

離散フーリエ変換の連続空間における数学的意味

- 離散フーリエ変換の連続空間における意味
 - 入出力: 有限の定義域区間を周期とする(無限区間の)周期関数を一定間隔で標本化したもの \rightarrow **離散・周期**
- (空間領域) $f(x)$ **離散** \rightarrow **周期** $F(u)$ (周波数領域)
周期 \rightarrow **離散**
- 

コサイン変換

- コサイン変換

$$F_c(u) = \int_0^{\infty} f(x) \cos(ux) dx$$

- 区間 $[0, \infty)$ で定義される関数 $f(x)$ を原点に関して鏡像変換して得られる(偶)関数 $f'(x)$ ($f'(x) = f(-x)$) をフーリエ変換した結果(の1/2)と等価である。

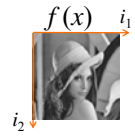
$$F_c(u) = \int_0^{\infty} f(x) \cos(ux) dx = \frac{1}{2} \int_{-\infty}^{\infty} f'(x) e^{-iux} dx$$

- $F_c(u)$ は実数(Re)成分のみとなる。
- $F_c(u) = F(-u)$ である。

離散フーリエ変換(DFT)と離散コサイン変換(DCT) 連続空間における数学的意味

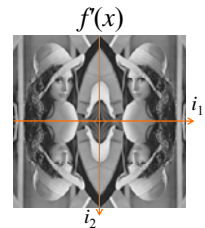
- 離散フーリエ変換

- 右の画像を上下左右に並べていき周期関数にしたものをフーリエ変換したものに等価。



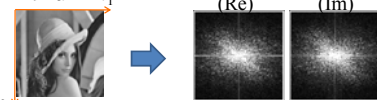
- 離散コサイン変換

- 上下左右の鏡像変換画像を並べた右の画像を、さらに、上下左右に並べていき周期関数にしたものをフーリエ変換したものに等価。(偶関数になっていることに注意！)

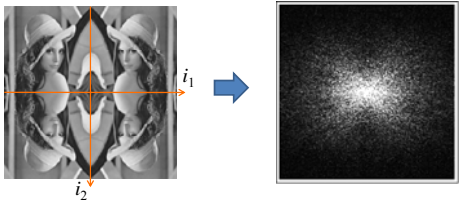


離散フーリエ変換(DFT)と離散コサイン変換(DCT) 連続空間における数学的意味

- 離散フーリエ変換

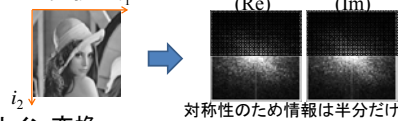


- 離散コサイン変換



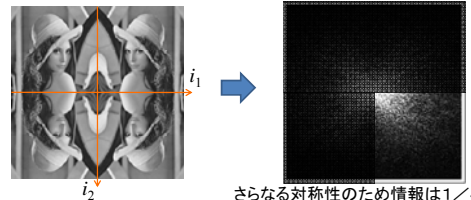
離散フーリエ変換(DFT)と離散コサイン変換(DCT) 連続空間における数学的意味

- 離散フーリエ変換



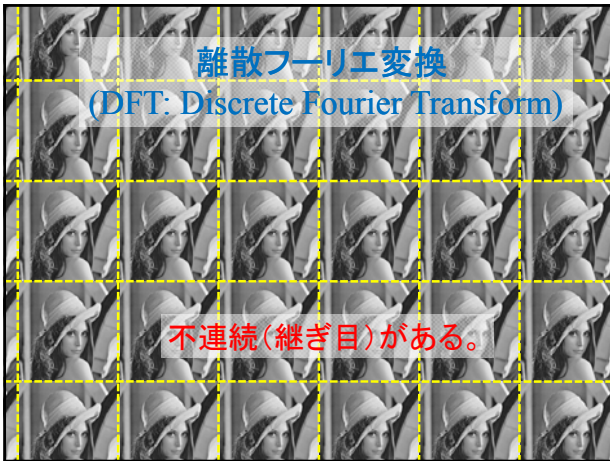
対称性のため情報は半分だけ

- 離散コサイン変換



さらなる対称性のため情報は1/4だけ





離散フーリエ変換(DFT)・離散コサイン変換(DCT)

- 連続系で解釈すれば、
 - 離散フーリエ変換は、有限の離散画像を単純に並べて周期化したものを、連続フーリエ変換したものである。
 - 画像の境界で不連続が発生する(本来の画像には無い高周波数成分が発生してしまう)。
 - これにより、低周波数成分への集中が弱まりエントロピーが大きくなる。
 - 離散コサイン変換は、有限の離散画像を(縦・横それぞれ)鏡像変換しながら並べて周期化したものである。
 - 鏡像変換することにより画像境界が連続的につながる(離散フーリエ変換のような高成分が発生しない)。

DFT

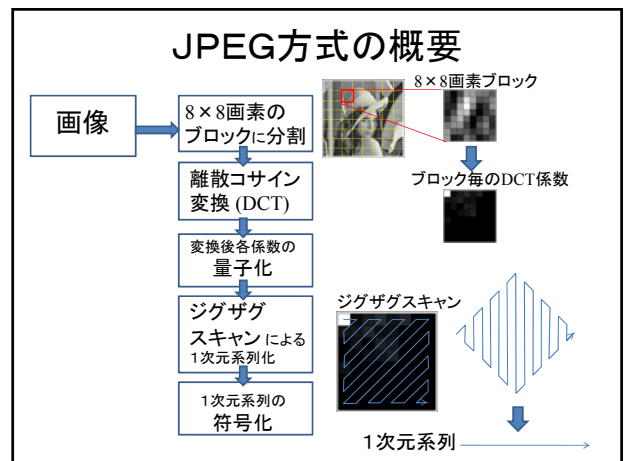
DCT

JPEG方式

本資料ではモノクロ画像(輝度成分)のみを対象とする。

JPEG方式の概要

- 画像を8×8画素の**ブロック分割**を行う。
- 8×8画素毎に以下の圧縮符号化を行う。
 - 離散コサイン変換(DCT)によりエントロピーを低減する。
 - DCTの各係数値を**量子化**する。
 - 量子化した各係数値を以下の手順で符号化する。
 - 8×8の2次元的配列を、**ジグザグスキャン**により、低周波成分から高周波成分に並ぶよう1次元系列化する。
 - 1次元系列となった係数値に対して、**差分符号化(直流成分)**、**ランレングス符号化**を行う。
 - さらに**ハフマン符号化**を行う。



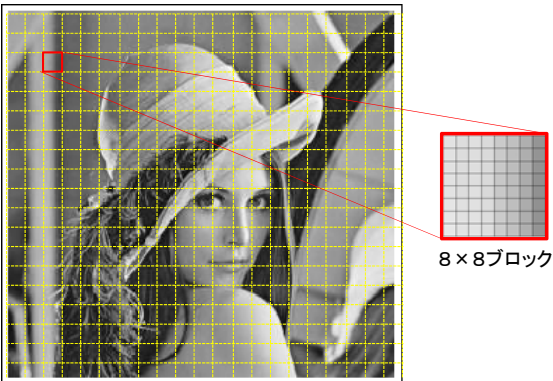
JPEG方式の概要

- 画像を8×8画素の**ブロック分割**を行う。
- 8×8画素毎に以下の圧縮符号化を行う。
 - **離散コサイン変換**によりエントロピーを低減する。
 - DCTの各係数値を**量子化**する。
 - 量子化した各係数値を以下の手順で符号化する。
 - 8×8の2次元的配列を、**ジグザグスキャン**により、低周波成分から高周波成分に並ぶよう1次元系列化する。
 - 1次元系列となった係数値に対して、**差分符号化(直流成分)**、**ランレングス符号化**を行う。
 - さらに**ハフマン符号化**を行う。

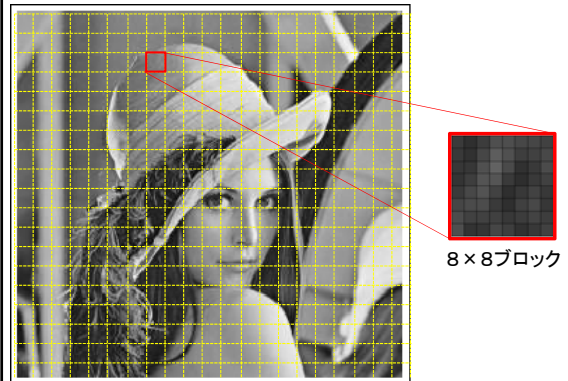
8×8ブロック分割



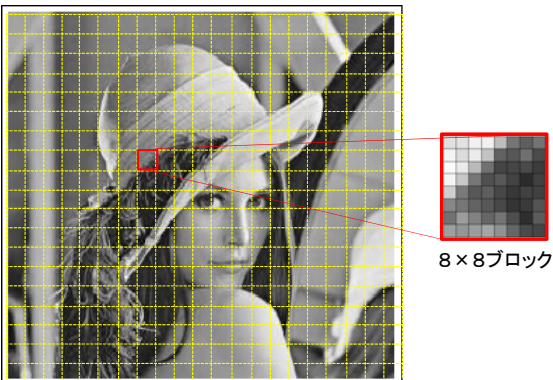
8×8ブロック分割



8×8ブロック分割

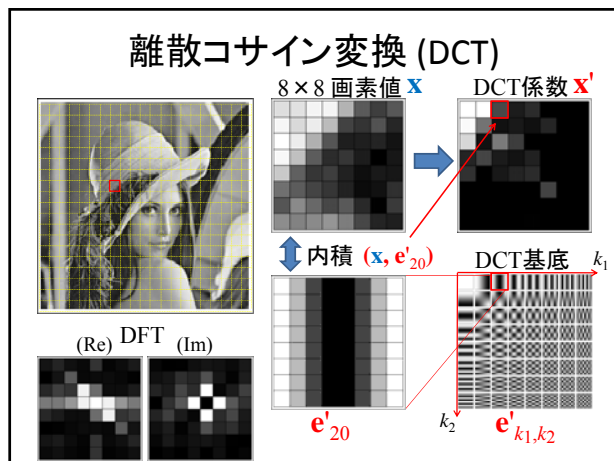
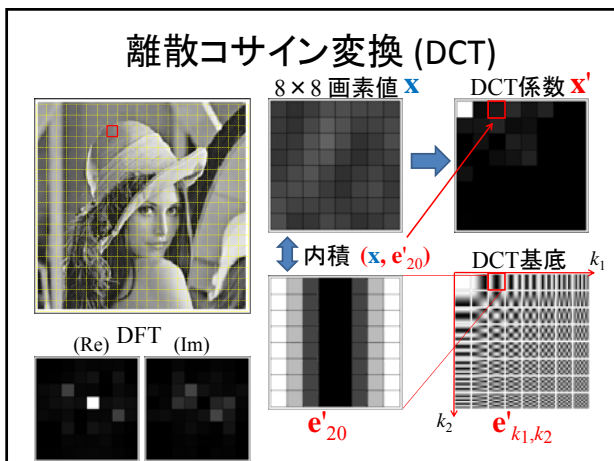
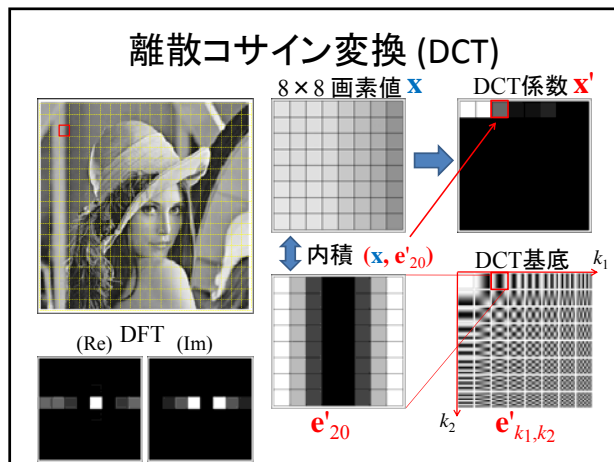
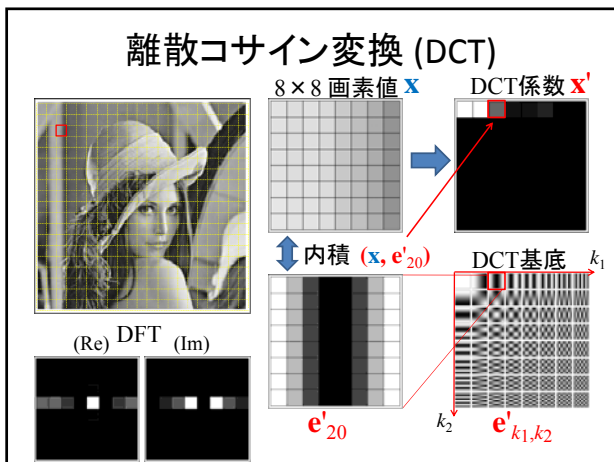
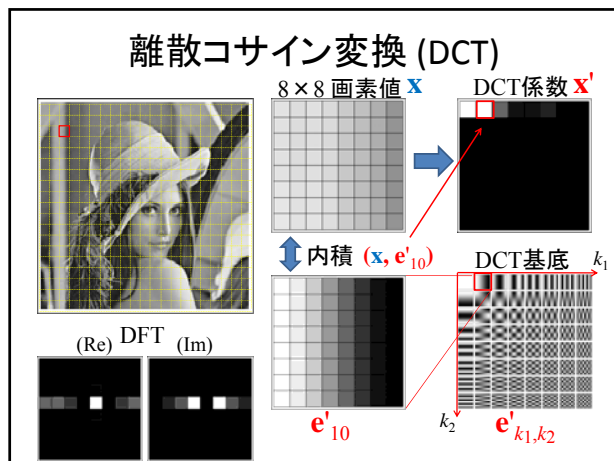
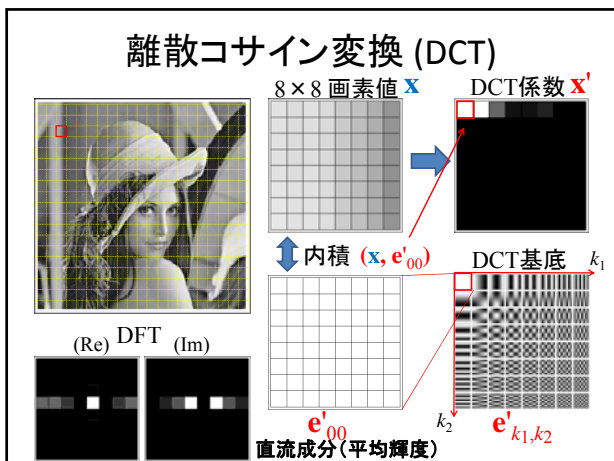


8×8ブロック分割



JPEG方式の概要

- 画像を8×8画素の**ブロック分割**を行う。
- 8×8画素毎に以下の圧縮符号化を行う。
 - **離散コサイン変換**によりエントロピーを低減する。
 - DCTの各係数値を**量子化**する。
 - 量子化した各係数の値を以下の手順で符号化する。
 - 8×8の2次元的配列を、**ジグザグスキャン**により、低周波成分から高周波成分に並ぶよう1次元系列化する。
 - 1次元系列となった係数値に対して、**差分符号化(直流成分)**、**ランレングス符号化**を行う。
 - さらに**ハフマン符号化**を行う。

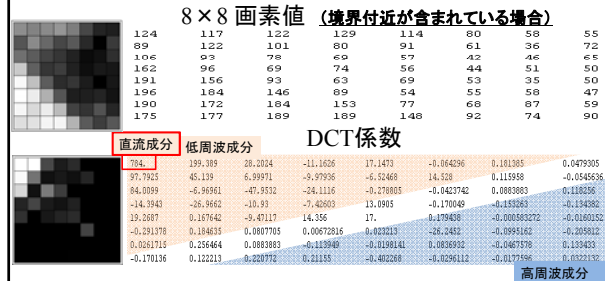


JPEG方式の概要

- 画像を8×8画素の**ブロック分割**を行う。
- 8×8画素毎に以下の**圧縮符号化**を行う。
 - **離散コサイン変換**によりエントロピーを低減する。
 - **DCTの各係数値を量子化**する。
 - 量子化した各係数値を以下の手順で符号化する。
 - 8×8の2次元配列を、**ジグザグスキャン**により、低周波成分から高周波成分に並ぶよう1次元系列化する。
 - 1次元系列となった係数値に対して、**差分符号化(直流成分)**、**ランレングス符号化**を行う。
 - さらに**ハフマン符号化**を行う。

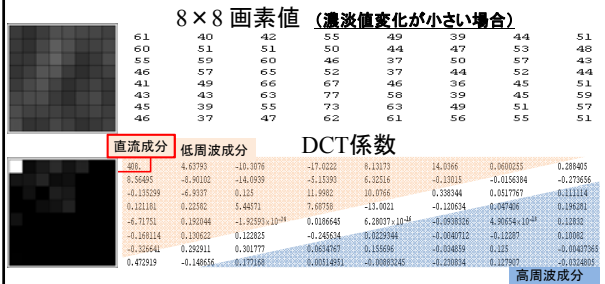
DCT係数の特性

- 浮動小数点の数値データである。
- 通常の画像では、高周波成分(青領域)は、直流(赤枠)・低周波成分(ピンク領域)にくらべはるかに小さな数値(無視できる程度の数値/ほぼゼロ)となる。



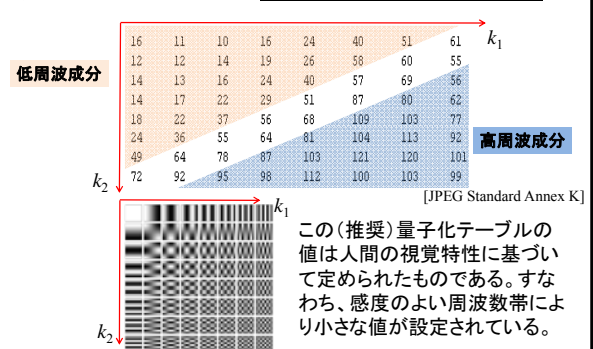
DCT係数の特性

- 不動小数点の数値データである。
- 通常の画像では、高周波成分(青領域)は、直流(赤枠)・低周波成分(ピンク領域)にくらべはるかに小さな数値(無視できる程度の数値/ほぼゼロ)となる。



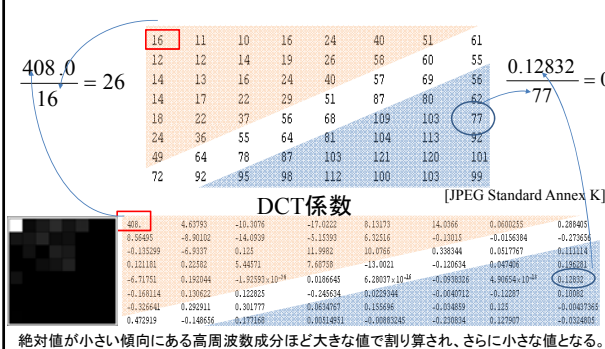
量子化テーブル

- DCTの各係数を、予め与えられた量子化テーブルの数値で割り算し、小数点以下を丸める。(高周波数成分の多くは、ゼロになる。)



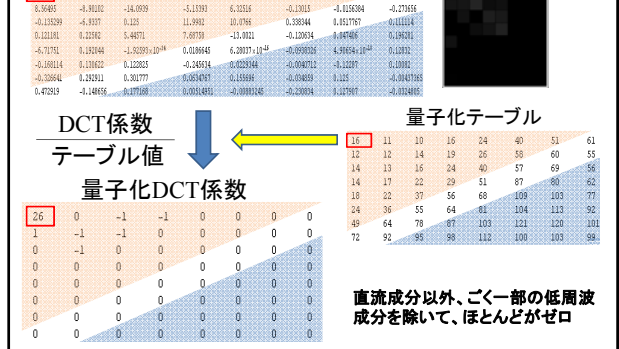
量子化テーブル

- DCTの各係数を、予め与えられた量子化テーブルの数値で割り算し、小数点以下を丸める。(高周波数成分の多くは、ゼロになる。)



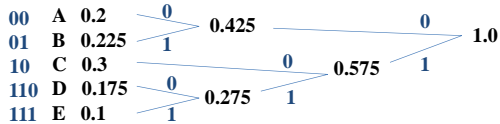
量子化

- DCTの各係数を、予め与えられた量子化テーブルの数値で割り算し、小数点以下を丸める。(高周波数成分の多くは、ゼロになる。)



ハフマン符号の復習

- 前提条件
 - 各シンボルの出現確率が既知である。
- アルゴリズム
 - 以下のペア化の処理を繰り返す。
 - 最も出現確率の低い2つのシンボルをペアにする。
 - これら2つのシンボルを新しい1つのシンボルとみなす。
 - 以上の処理により2分木が構成される。
 - 2分木の各分岐に0/1を割り当てる。
 - 根から各分岐の0/1を連結することにより符号を得る。



直流(DC)成分差分のサイズ分類とそのハフマン符号

- 直流成分の差分符号化における仮定
 - 隣接ブロックの直流成分(平均輝度値)は近い値である。
 - 差分値0の出現頻度が最も高く、差分値の絶対値が大きくなるにしたがい、出現頻度は低下する。
- 差分値をsize分類符号テーブルに従って符号化

DC Coef Difference	Size	Typical Huffman codes for Size	Additional Bits (in binary)
0	0	00	-
-1,1	1	010	0,1
-3,-2,2,3	2	011	00,01,10,11
-7,...,-4,4,...,7	3	100	000,...,011,100,...,111
-15,...,-8,8,...,15	4	101	0000,...,0111,1000,...,1111
:	:	:	:
-1023,...,-512,512,...,1023	10	1111 1110	00 0000 0000,...,11 1111 1111
-2047,...,1024,1024,...,2047	11	1 1111 1110	000 0000 0000,...,111 1111 1111

Kingsbury

<http://cnx.org/content/m11096/latest/>

直流(DC)成分差分のサイズ分類とそのハフマン符号

- 具体例
 - 差分値 0: Size 0 (Code 00), Additional Bits - → Huffman Code: **00**
 - 差分値 -1: Size 1 (010), Additional Bits 0 → Huffman Code: **010 0**
 - 差分値 4: Size 3 (100), Additional Bits 100 → Huffman Code **100 100**
 - 差分値 -15: Size 4 (101), Additional Bits 0000 → Huffman Code **101 0000**

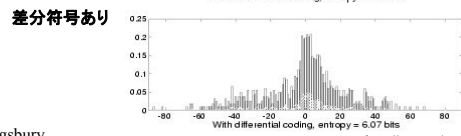
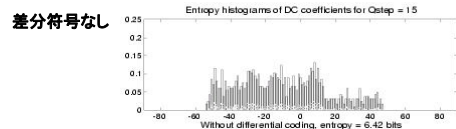
DC Coef Difference	Size	Typical Huffman codes for Size	Additional Bits (in binary)
0	0	00	-
-1,1	1	010	0,1
-3,-2,2,3	2	011	00,01,10,11
-7,...,-4,4,...,7	3	100	000,...,011,100,...,111
-15,...,-8,8,...,15	4	101	0000,...,0111,1000,...,1111
:	:	:	:
-1023,...,-512,512,...,1023	10	1111 1110	00 0000 0000,...,11 1111 1111
-2047,...,1024,1024,...,2047	11	1 1111 1110	000 0000 0000,...,111 1111 1111

Kingsbury

<http://cnx.org/content/m11096/latest/>

直流(DC)成分差分のハフマン符号

- 差分値出現頻度分布の実例
 - エントロピー
 - 差分符号なし: 6.42 bits
 - 差分符号あり: 6.07 bits



Kingsbury

<http://cnx.org/content/m11096/latest/>

交流(AC)成分のハフマン符号

- 交流成分は、ランレングス符号 (Run, Coef) の系列
 - (Run, Coef) : 0 が Run 回続いた後、係数値 Coef が出現
 - 例: (0,-23) (0,-9) (0,-1) (2,1) (0,2) (0,2) (2,1) (0,1) (0,1) (11,-1) (5,-1) (8,1) EOB
- (Run, Coef) のハフマン符号
 - (Run, Coef の Size) のハフマン符号と付加ビットで表現する。Coef の Size 分類については、DC 成分と同じ分類に基づく。

(Run,Size)	Code Byte (hex)	Code Word (binary)	(Run,Size)	Code Byte (hex)	Code Word (binary)
(0,1)	01	00	(0,6)	06	1111000
(0,2)	02	01	(1,3)	13	1111001
(0,3)	03	100	(5,1)	51	1111010
(EOB)	00	1010	(6,1)	61	1111011
(0,4)	04	1011	(0,7)	07	11111000
(1,1)	11	1100	(2,2)	22	11111001
(0,5)	05	11010	(7,1)	71	11111010
(1,2)	12	11011	(1,4)	14	11111011
(2,1)	21	11100	:	:	:
(3,1)	31	111010	(ZRL)	F0	1111111001
(4,1)	41	111011	:	:	:

Kingsbury

<http://cnx.org/content/m11096/latest/>

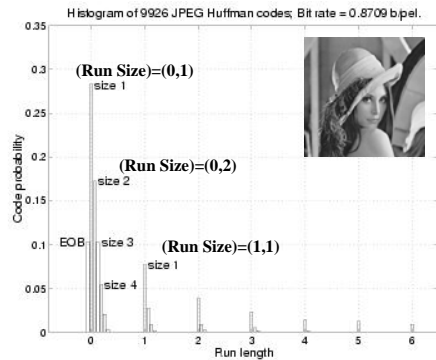
交流(AC)成分のハフマン符号

- 具体例
 - (Run, Coef) = (0,-7) : (Run Size)=(0,3) + 付加ビット000 → **100 000**
 - (Run, Coef) = (1,3): (Run, Size) = (1,2) + 付加ビット11 → **11011 11**

AC Coef	Size	Additional Bits (in binary)
0	0	-
-1,1	1	0,1
-3,-2,2,3	2	00,01,10,11
-7,...,-4,4,...,7	3	000,...,011,100,...,111

(Run,Size)	Code Word (binary)
(0,1)	00
(0,2)	01
(0,3)	100
(EOB)	1010
(0,4)	1011
(1,1)	1100
(0,5)	11010
(1,2)	11011
(2,1)	11100

交流成分係数の出現頻度



Kingsbury

<http://cnx.org/content/m11096/latest/>

JPEG方式まとめ

- 8×8画素ブロック分割 8×8画素 → DCT係数
- ブロック単位の
離散コサイン変換 (DCT) によるエントロピー削減
 - 人間視覚の周波数感度特性を考慮したDCT係数の量子化
 - ジグザグスキャンによる量子化DCT係数の1次元化 (直流、低周波～高周波数成分)
 - (直流成分の)差分符号化、(交流成分の)ランレンクス符号化

- ハフマン符号化

ジグザグスキャン

(0,-23) (0,-9) (0,-1) (2,1) (0,2) (0,2) (2,1)
(0,1) (0,1) (11,-1) (5,-1) (8,1) (8,1) EOB

ハフマン符号
テーブル

ハフマン符号

画像直交変換・JPEG関連の有用なウェブページ

- 信州大学・井澤先生のホームページ
 - <http://laputa.cs.shinshu-u.ac.jp/~yizawa/InfSys1/basic/index.htm>
 - <http://laputa.cs.shinshu-u.ac.jp/~yizawa/InfSys1/advanced/index.htm>
 - http://laputa.cs.shinshu-u.ac.jp/~yizawa/InfSys1/ref_contents/index.htm
- 広島大学・浅野先生のホームページ
 - <http://kuva.mis.hiroshima-u.ac.jp/~asano/Kougi/01a/Tokuron/>
- スタンフォード大Prof. Bernd Girod のImage Communication I 講義資料
 - <http://www.stanford.edu/class/ee398a/handouts.htm>
- JPEG Entropy Coding and 2D-DCT by Nick Kingsbury
 - <http://cnx.org/content/m11096/latest/>
 - <http://cnx.org/content/m11094/latest/>
- Wikipedia
 - JPEG, 離散コサイン変換