

---

# 画像データの圧縮

# 画像ファイルのデータ量

---

- 無圧縮画像ファイルのデータ量は  
(画素数) × (1画素ごとのビット数)で決まる
  - 例)iphone16 の場合  
4800万画素 × 3byte (24bit) ≈ 144 MB
- ファイルサイズが大きく扱いづらいので、  
人間の目で見ても違いがわからない程度に  
データを圧縮

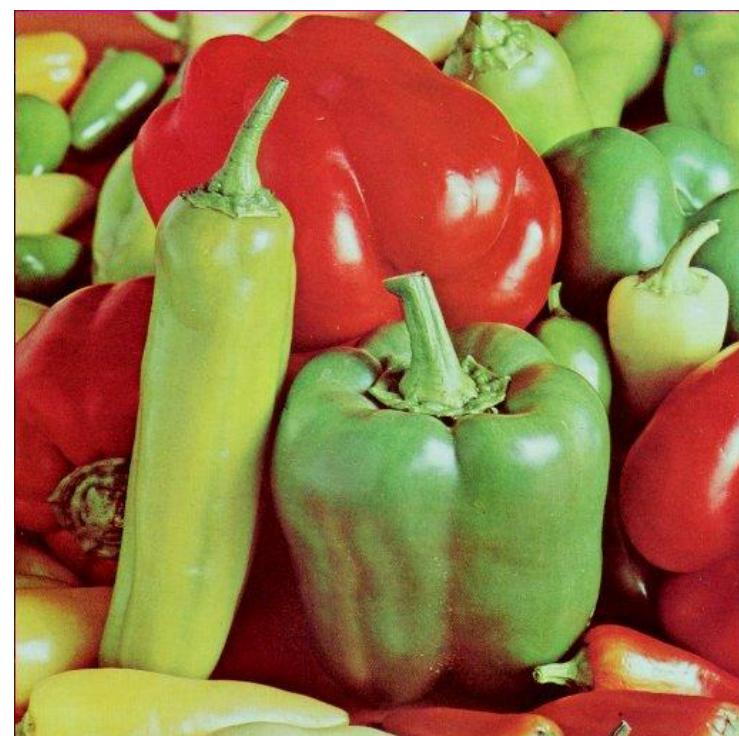
# 画像データの圧縮

---



圧縮前(BMP)

768 KB



圧縮後(JPG)

58 KB

# 画像の符号化方式

- 非可逆・可逆符号化でも分類できる

分類	特徴	符号化方式
エントロピー符号化	出現確率を利用する	ハフマン符号化
予測符号化	画素ごとにシンボル変換する	DPCM符号化
変換符号化	他の空間に変換する	離散コサイン変換
2値画像の符号化	画素値が0,1である特殊性を利用する	ランレンジス符号化

# 画像の情報量

---

- 情報量 = 出現確率の逆数を対数にとることで算出
- 画像のエントロピーは以下の式で求める
  - 画像が取り得る階調数に対して、各画素の出現確率と情報量を掛け合わせて、和を取ることで算出できる

$$H_f = \sum_{f=0}^{L-1} p(f) \log_2 \frac{1}{p(f)}$$

$L$ : 画像の階調数,  $p(f)$ : 画素値  $f$  の出現確率

# 可変長符号

---

- 等長符号

- 無圧縮の画像では、画素値を2進数列に対応
- 符号の長さが全ての画素値で等しい

- 可変長符号

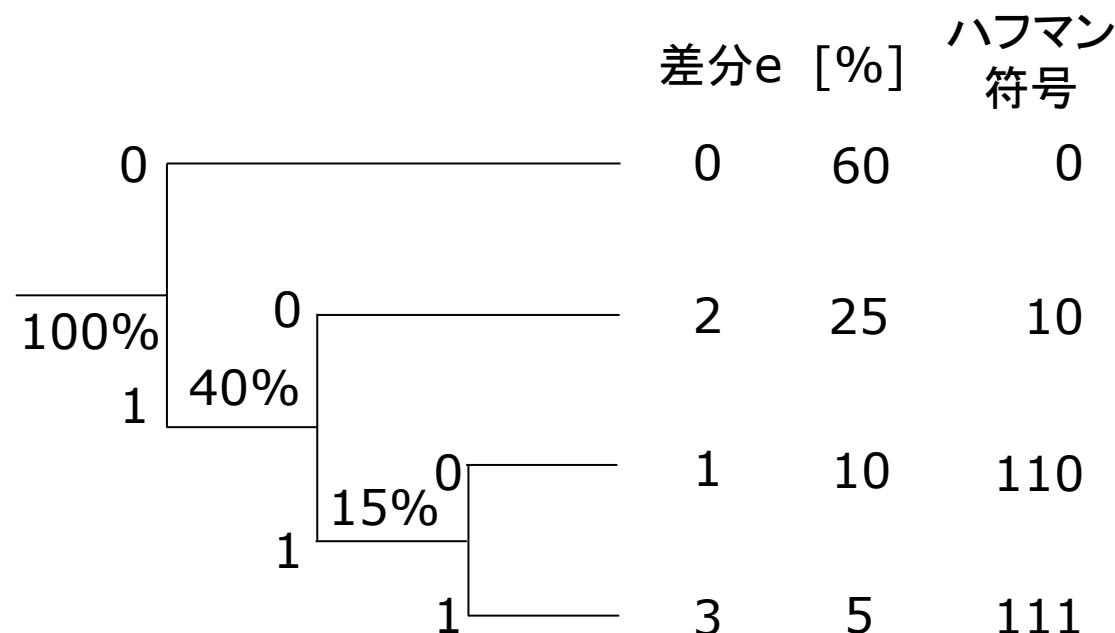
- 出現確率の大きい画素値に短い符号長、  
小さい画素値に長い符号長を割り当てることで  
平均符号長 $\lambda$ を小さくできる

$$\lambda = \sum_{f=0}^{L-1} p(f)l(f)$$

$l(f)$ : 画素値 $f$ の符号長

# エントロピー符号化: ハフマン符号化

画素値	0	1	2	3
出現確率[%]	60	10	25	5



# エントロピー符号化: ハフマン符号化

画素値	0	1	2	3
出現確率[%]	60	10	25	5
符号	0	110	10	111

平均符号長

$$1 \times 0.60 + 3 \times 0.10 + 2 \times 0.25 + 3 \times 0.05 = 1.55$$

復号化

符号データ 01101000111

0 110 10 0 0 111

復号データ 0 1 2 0 0 3

# 演習：ハフマン符号化

---

- 8つの画素値とその出現確率が表のように表されるとき、ハフマン符号化を行う
- 平均符号長を求め、等長符号の場合と比較する

8つの画素値と出現確率一覧

画素値	0	1	2	3	4	5	6	7
出現確率[%]	28	14	23	10	12	4	6	3

# 予測符号化

- 隣接する画素同士の濃度値は近い値を持つことを利用して符号化

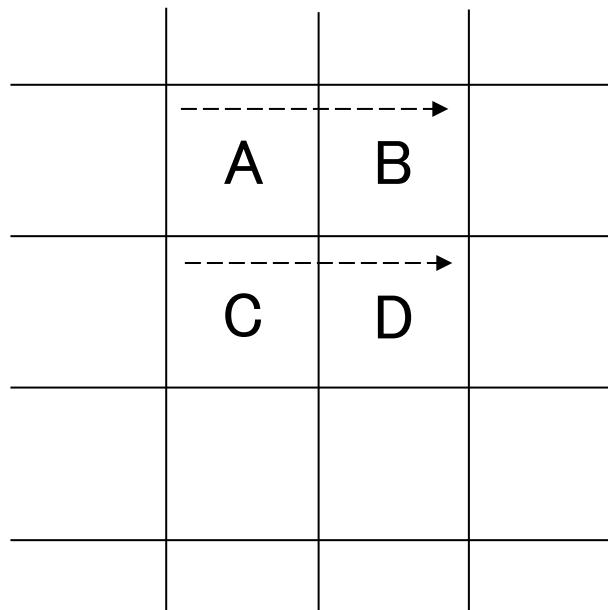
画素 $D$ の予測値 $\hat{D}$ は

$$\hat{D} = \frac{A + B + C}{3}$$

実際の値との差分

$$e = \hat{D} - D$$

を符号化データとして記録



# 予測符号化

- 差分のヒストグラムは鋭いピークを持つ
- 出現確率の偏りを利用してデータ圧縮

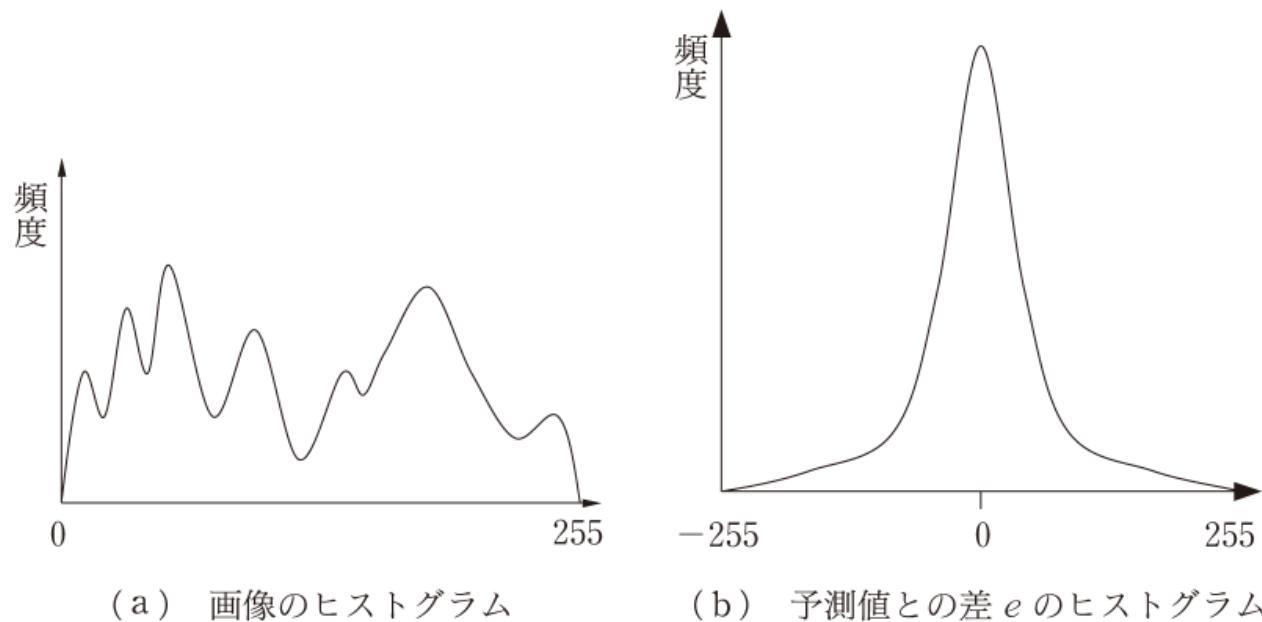


図 4.23 出現確率の偏りを利用した圧縮

# 予測符号化:DPCM符号化

---

- DPCM(Differential pulse code modulation)
- 注目画素 $f(i, j)$ と隣接画素 $f(i - 1, j)$ との  
差分 $d(i, j)$ を伝送する
$$d(i, j) = f(i, j) - f(i - 1, j)$$
- 受信側では、すでに得られた $f(i - 1, j)$ に  
差分 $d(i, j)$ を加算する
- 差分画像に対して、  
エントロピー符号化を行うことで圧縮率を高める

# 予測符号化:DPCM符号化

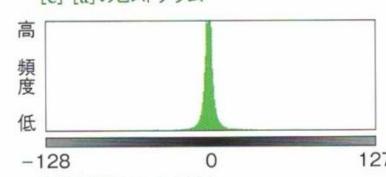
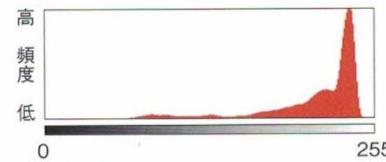
- 差分画像はヒストグラムに偏りが生じる



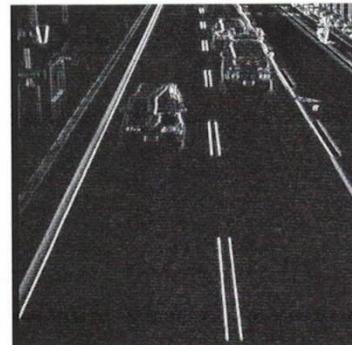
[a] 原画像



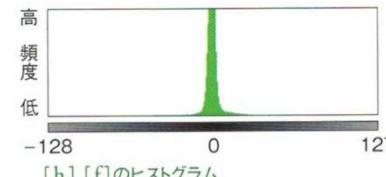
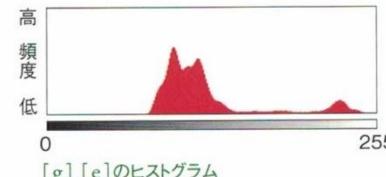
[b] 差分画像



[e] 原画像



[f] 差分画像



■図16.5——DPCM符号化

# 直交変換を用いた画像圧縮

- 隣接2画素の濃度値( $f_1, f_2$ )を取り出す
- この分布は $f_1 = f_2$ に集中するため、45度回転した座標軸に直交変換して考える

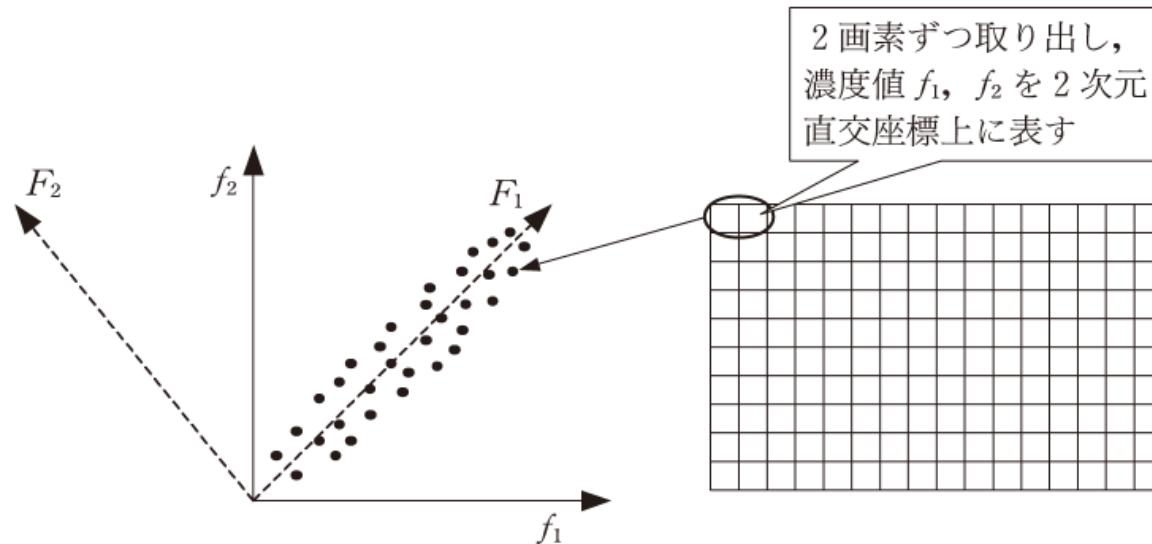
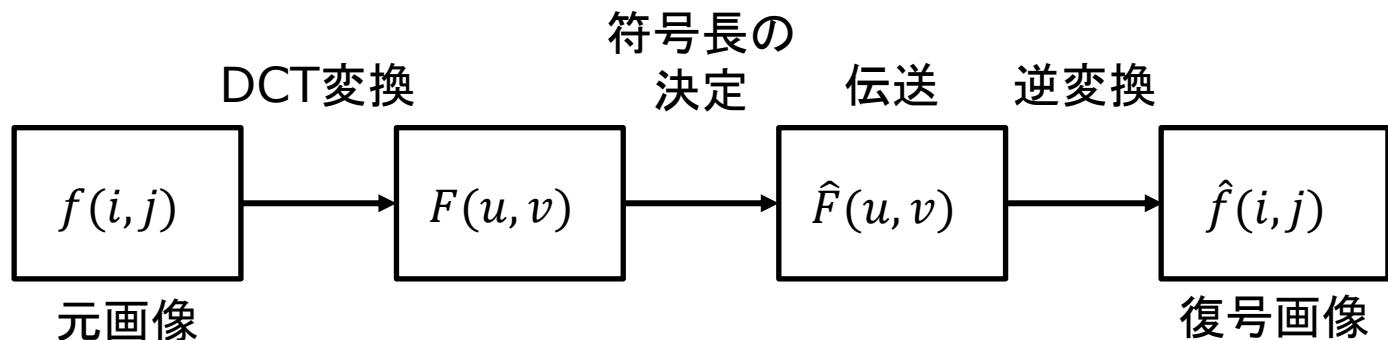


図 4.26 直交変換を用いた画像圧縮の基本概念

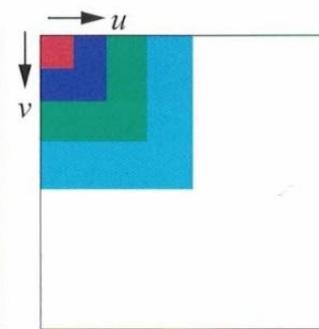
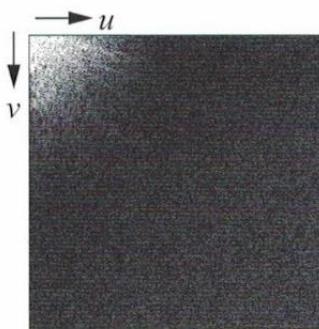
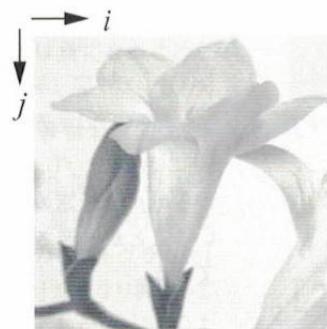
# 離散コサイン変換(DCT)符号化

- JPEGやMPEGなどの画像圧縮方式において用いられる
- DCT(Discrete cosine transform)を用いることで**比較的劣化が少なく高い圧縮率を得ている**



# 離散コサイン変換(DCT)符号化

- 符号化に用いる周波数成分と復号画像の関係



[a] 原画像

[b] DCT画像

[c] 符号化に用いる周波数成分



[d] 1/4の係数 ([c] のシアン色まで)による復号画像



[e] 1/16の係数 ([c] の緑色まで)による復号画像



[f] 1/64の係数 ([c] の青色まで)による復号画像

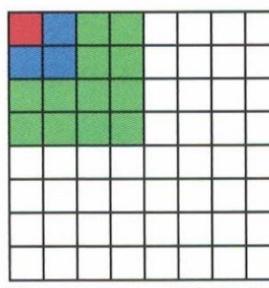
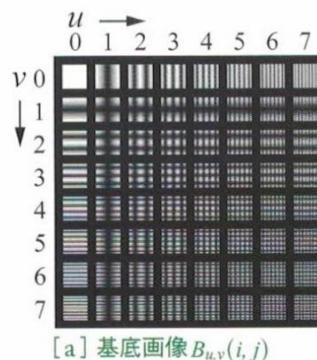


[g] 1/256の係数 ([c] の赤色)による復号画像

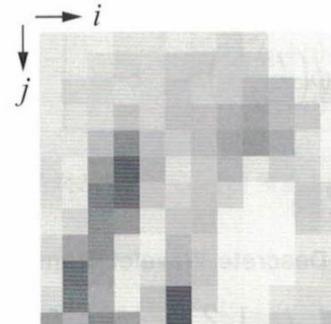
■図16.8—変換符号化と復号画像

# 離散コサイン変換(DCT)符号化

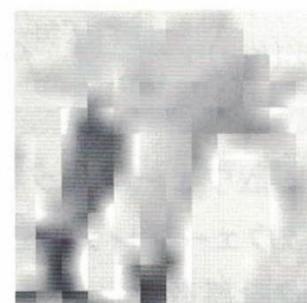
- 実際には、8×8画素のブロックに分割し、個々に変換
- 基底画像 $B_{u,v}(i,j)$ に重み $F(u,v)$ をつけて加算し復元



$$f(i,j) = \sum_u \sum_v F(u,v) B_{u,v}(i,j)$$



[c] 左上1枚の基底画像を用いた復号画像



[d] 左上4枚の基底画像を用いた復号画像



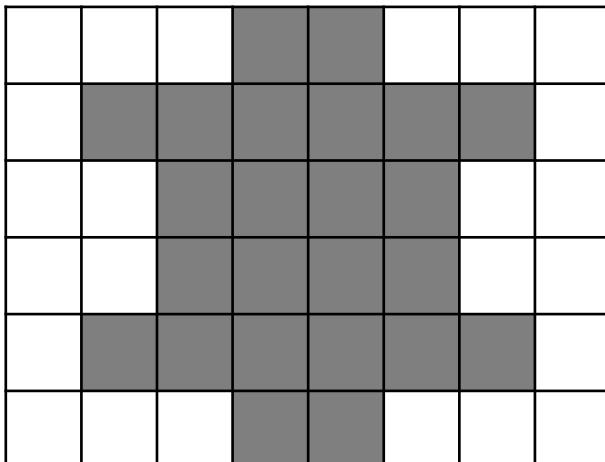
[e] 左上16枚の基底画像を用いた復号画像

■図 16.10——基底画像と復号画像

# 可逆符号化: ランレンジス符号化

- ラン: 同じ値のデータの並び
- ランレンジス: ランが連續する長さ

例: 8×6の2値画像



左上から白と黒のランレンジスを記述する  
白3黒2白4黒6白3黒4白4黒4白3黒6白4黒2白3

最初が白, 横幅が8画素であるとわかっていれば  
以下のように記述できる  
3245344436423

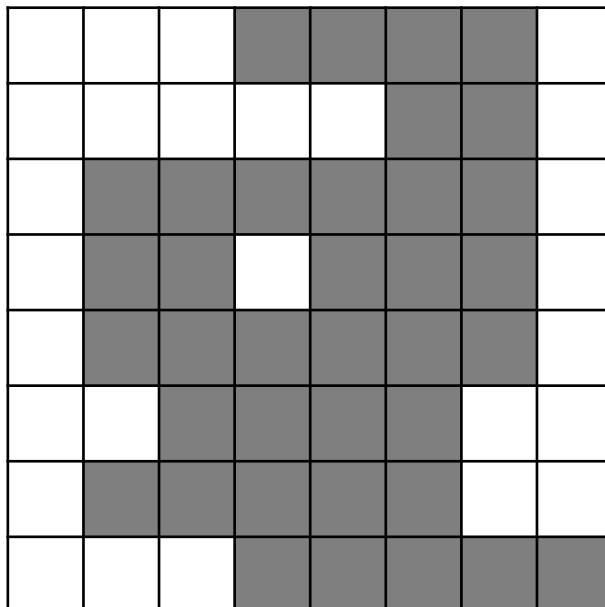
これら13個のシンボルに3bitずつ割り当てれば  
 $13 \times 3 = 39$ bit

# 演習: ランレンジス符号化

---

- 以下の $8 \times 8$ 画素の2値画像についてランレンジス符号化を行う

$8 \times 8$ の2値画像



# 標準化された符号化方式

---

- JPEGは、静止画像の国際標準符号化方式として採用され、広く使用されている
- 可逆符号化と非可逆符号化方式の2種類
  
- 可逆符号化の場合
  - DPCM符号化とエントロピー符号化により圧縮
- 非可逆符号化の場合
  - 色差情報の削減(間引き)
  - DCT変換による高周波成分の量子化
  - エントロピー符号化により圧縮

# JPEGの非可逆符号化方式

1. カラー画像をRGBからYIQ信号に変換  
I,Qの色差の情報量を1/4に間引く

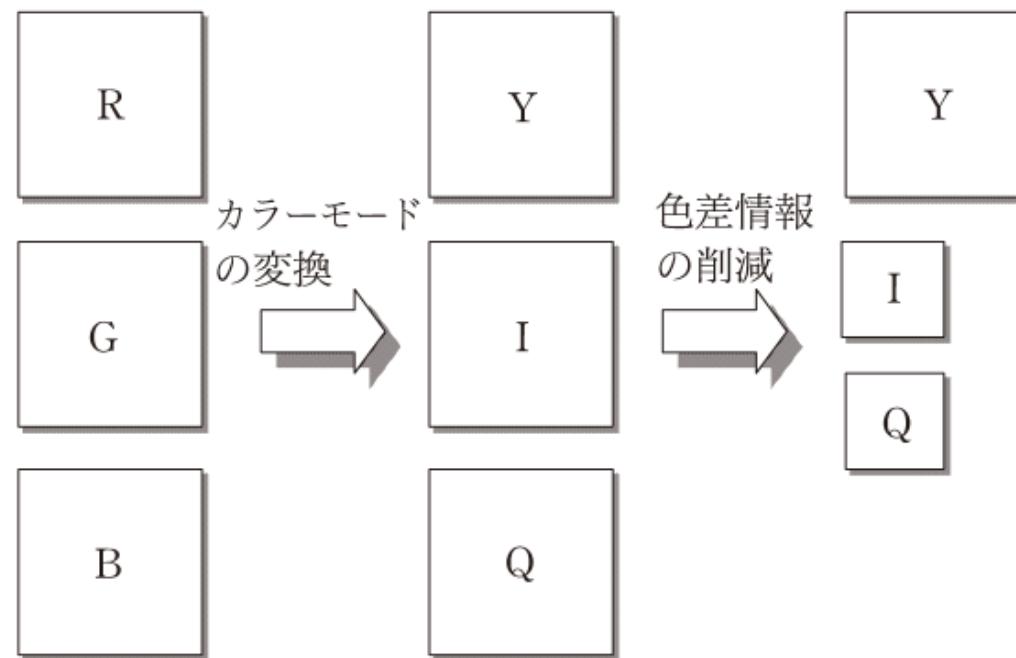
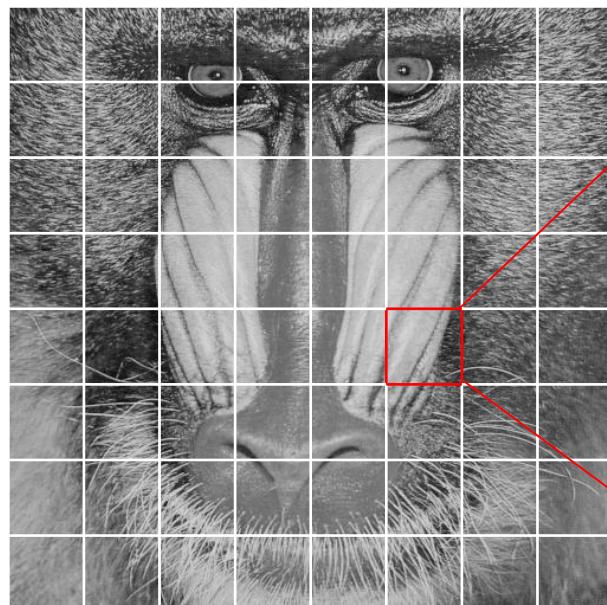


図 4.28 カラー モード の 変換 および 色差 情報 の 削減

# JPEGの非可逆符号化方式

2. YIQの各チャンネルの画像をブロック( $8 \times 8$ 画素)単位に分ける



72	64	62	59	71	64	52	61
62	60	56	69	65	61	60	68
59	57	66	62	52	59	62	52
61	52	49	51	58	53	52	52
52	55	51	70	52	59	66	62
56	52	61	58	62	56	52	41
54	41	52	54	64	51	39	33
44	52	55	62	52	44	28	31

$8 \times 8$  ブロックの画素値

# JPEGの非可逆符号化方式

## 3. ブロックごとにDCT変換し、周波数分布を計算

72	64	62	59	71	64	52	61
62	60	56	69	65	61	60	68
59	57	66	62	52	59	62	52
61	52	49	51	58	53	52	52
52	55	51	70	52	59	66	62
56	52	61	58	62	56	52	41
54	41	52	54	64	51	39	33
44	52	55	62	52	44	28	31

8×8ブロックの画素値

DC成分

160	31	-44	32	15	-11	-10	9
10	-20	30	-16	-9	-11	-3	-3
-61	28	-18	10	8	-13	19	-5
34	-9	3	8	-5	-9	0	-1
-8	2	8	-5	-9	0	3	2
-20	19	4	19	-10	-7	9	7
2	13	-11	-10	-10	11	3	-3
13	-7	2	-3	4	11	7	0

DCT係数

# JPEGの非可逆符号化方式

## 4. 量子化行列で割る

160	31	-44	32	15	-11	-10	9
10	-20	30	-16	-9	-11	-3	-3
-61	28	-18	10	8	-13	19	-5
34	-9	3	8	-5	-9	0	-1
-8	2	8	-5	-9	0	3	2
-20	19	4	19	-10	-7	9	7
2	13	-11	-10	-10	11	3	-3
13	-7	2	-3	4	11	7	0

DCT係数

3	5	7	9	11	13	15	15	17
5	7	9	11	13	15	17	19	
7	9	11	13	15	17	19	21	
9	11	13	15	17	19	21	23	
11	13	15	17	19	21	23	25	
13	15	17	19	21	23	25	27	
15	17	19	21	23	25	27	29	
17	19	21	23	25	27	29	31	

量子化行列

# JPEGの非可逆符号化方式

## 5. DC成分とAC成分を別々に圧縮



# MPEG

---

- 動画像圧縮に関する国際基準
- 静止画の場合と比較して、時間軸方向の圧縮も必要になる

MPEG1	ビデオCD, テレビ会議・電話等で採用 転送レートは約1～1.5 Mbps
MPEG2	DVD, Blue-ray, ディジタルテレビ等で採用 転送レートは約5～30Mbps
MPEG3	HDTV信号を目標とした規格として検討されたが、 MPEG2で実現可能なことがわかり消滅
MPEG4	元々は携帯電話等の低速度転送用だったが、 高い圧縮率が得られるため, Blue-ray等で採用

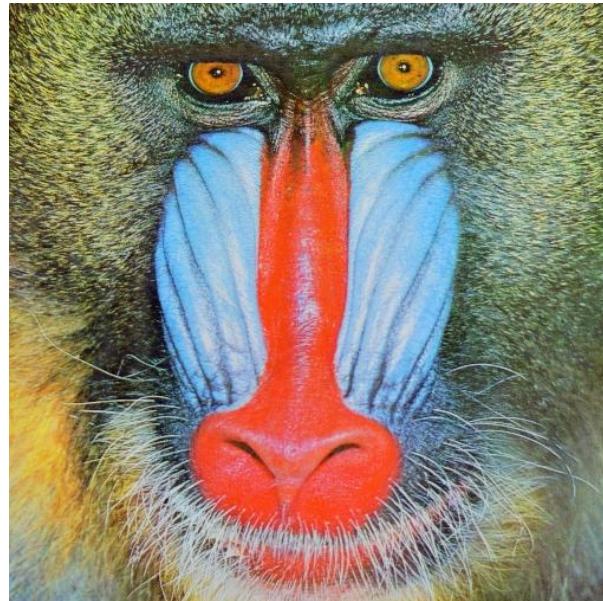
---

# 画像処理プログラミング7 JPEGによる画像データの圧縮

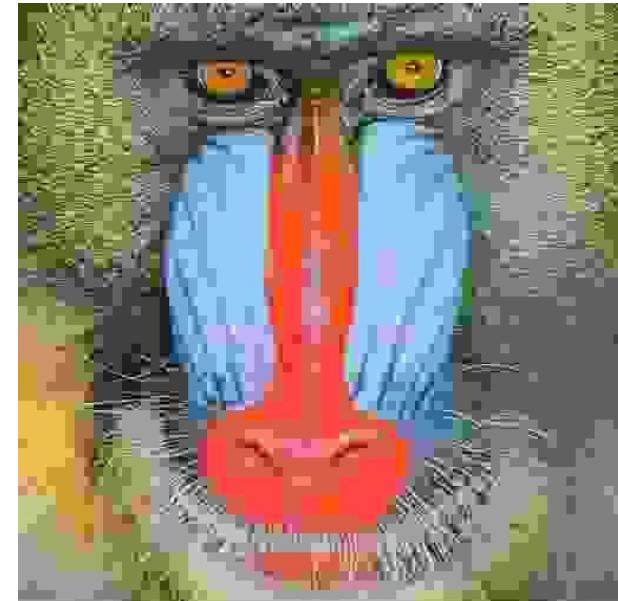
# JPEGによる非可逆圧縮

- qualityの値を0～100で変化させることで、  
画像の品質が変化

```
# JPEG圧縮を行う  
cv2.imwrite(output_path, img, [int(cv2.IMWRITE_JPEG_QUALITY), quality])
```



quality=100



quality=0

# 画像処理演習(レポート課題2に含める)

---

- カラー画像(color\_image.png)に対して, JPEGの圧縮品質を0~100まで10ずつ変化させながら, 非可逆圧縮を行い, 画像の品質とデータ圧縮率の関係を調査してください.
  - また, 上記の調査結果から, JPEG画像の品質をどの値に設定するべきか, 自分の考えを述べてください
- 
- データの圧縮率を求める際には, 可逆圧縮画像(color\_image.png)のデータサイズを基準とする
  - 画像の品質と圧縮率との関係は, グラフもしくは表として示せるとよい