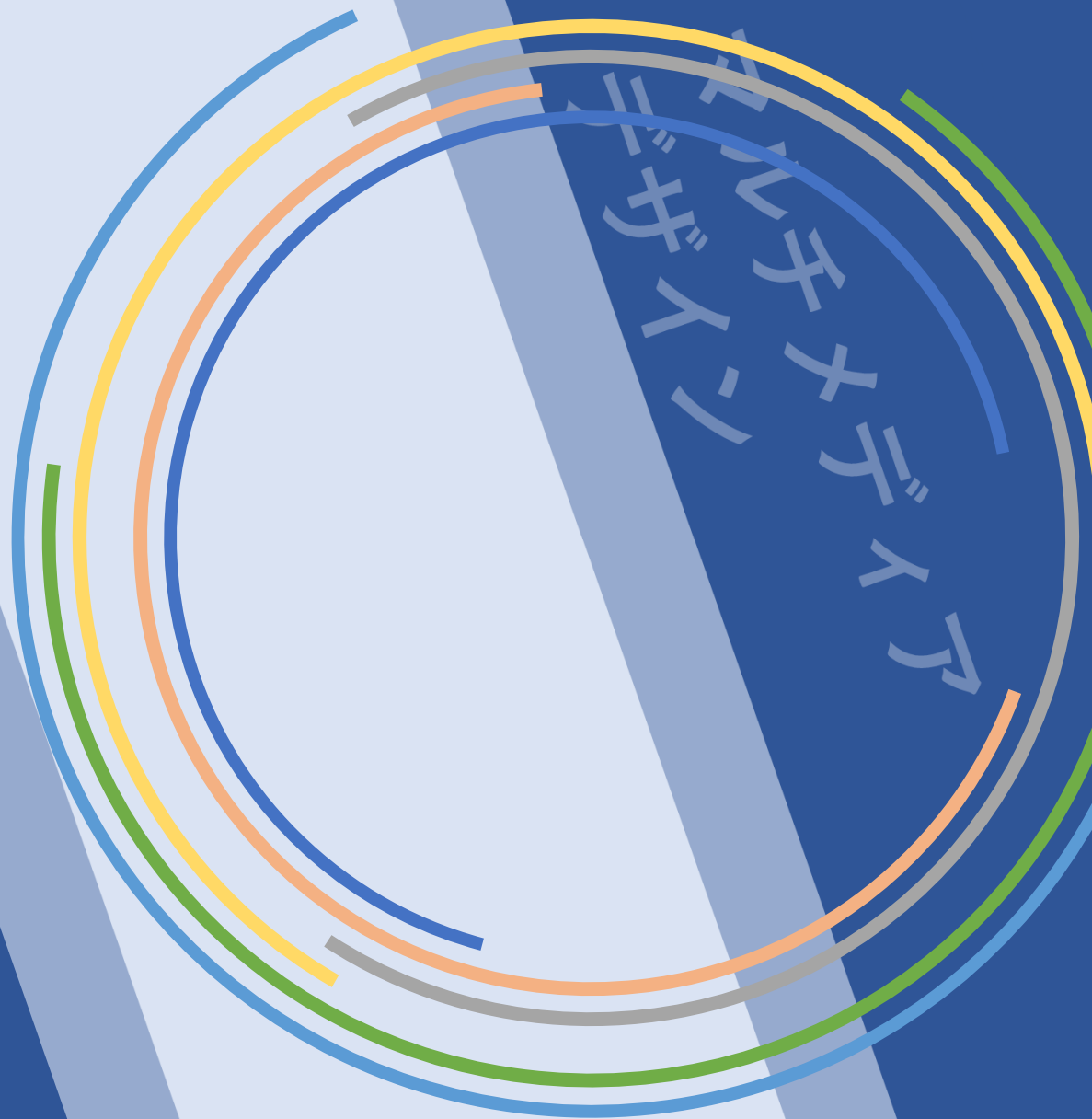


この授業のスライドはLMSで公開されます
実験の質問やレポートに関する質問などはSlackへ
<https://keio-st-multimedia.slack.com>



#8 圧縮・表現

担当： 西 宏章



領域抽出

44

- 4近傍または8近傍を利用したラベリング処理を行う

原画像(256x256)



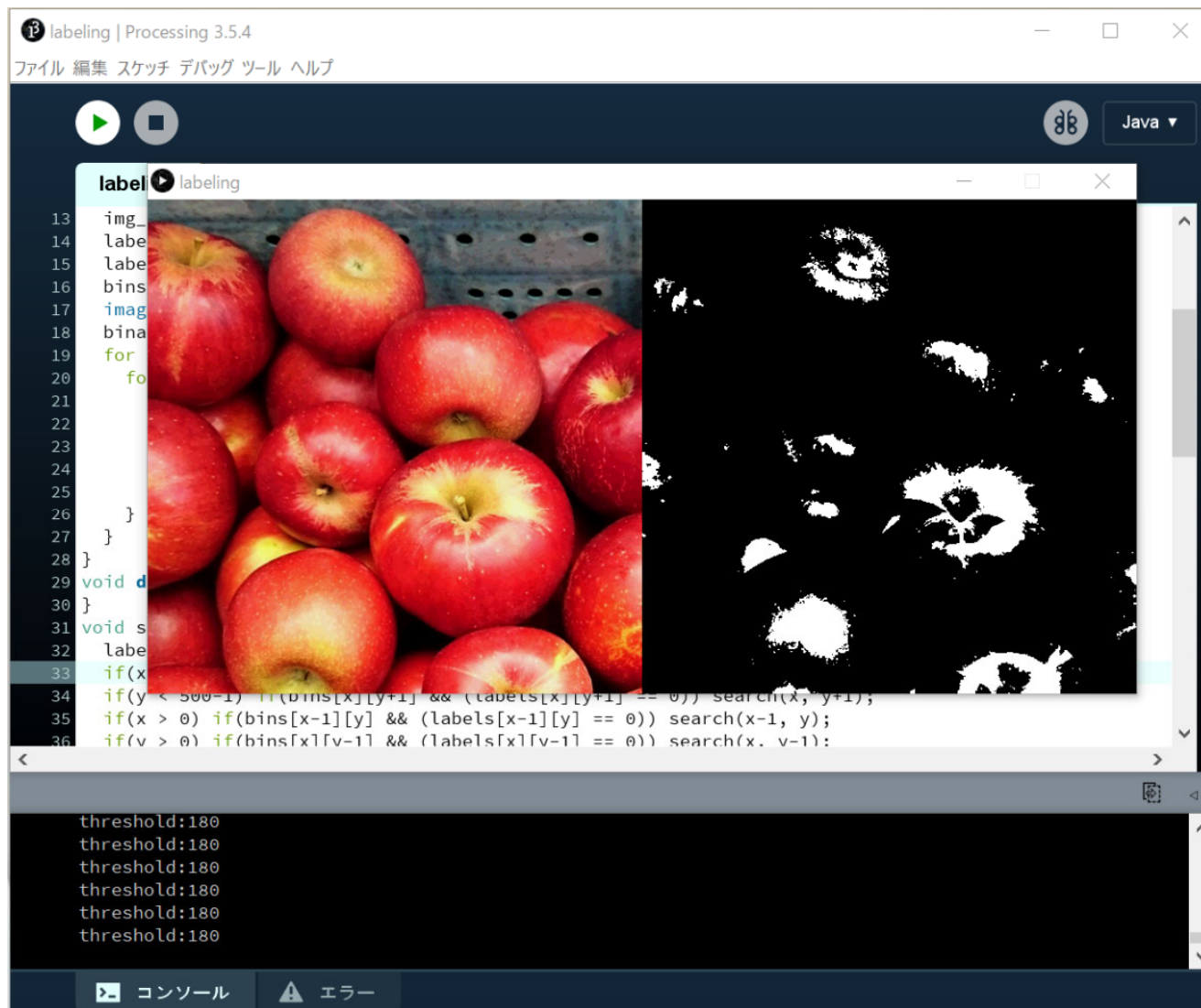
ラベリング結果





Processing: ラベリング

45





画像フォーマット

46

- 可逆変換と不可逆変換
 - 可逆変換：圧縮もしくは変換した符号から原画像を歪みなく（数値的な誤りなく）復元できる（ロスレス）
- 画像圧縮変換
 - ランレングス法
- カラーの表現
- 画像ファイルフォーマット
- 減色・パレット





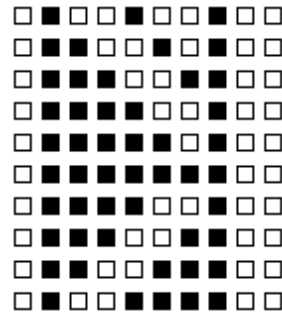
ランレングス法

47

- ファクシミリで送られる画像など、白色と黒色のみで構成された画像(2値画像)や、隣り合った画素の色の変化が少ない画像で有効
- 似た画素が続く冗長性を利用して「同じ色が連続する長さ」を符号化

例えば、右のような10x10の2値画像の場合

255	100	255	255	100	255	255	100	255	255
255	100	100	255	255	100	255	100	255	255
255	100	100	100	255	255	100	100	255	255
255	100	100	100	100	255	255	100	255	255
255	100	100	100	100	100	255	100	255	255
255	100	100	100	100	100	255	100	255	255
255	100	100	100	100	100	255	100	255	255
255	100	100	100	255	255	100	100	255	255
255	100	100	255	255	100	100	100	255	255
255	100	255	255	100	100	100	100	255	255



- 左上から順に右下へ連続して走査して圧縮すると次のようになる

255,1,100,1,255,2,100,1,255,2,100,1,255,3,
100,2,255,2,100,1,255,1,100,1,255,3,
100,3,255,2,100,2,255,3,
100,4,255,2,100,1,255,3,
100,5,255,1,100,1,255,3,
100,7,255,3,
100,4,255,2,100,1,255,3,
100,3,255,2,100,2,255,3,
100,2,255,2,100,3,255,3,
100,1,255,2,100,4,255,2

- 結果として、100ピクセル(100バイト)を86個の数値に圧縮したため、圧縮率86%
- なお、0と1が交互に並ぶためその数だけ表記して次のようになる
1,1,2,1,2,1,3, 2,2,1,1,1,3, 3,2,2,3, 4,2,1,3, 5,1,1,3, 7,3, 4,2,1,3, 3,2,2,3, 2,2,3,3, 1,2,4,2
- この場合の圧縮率は43%



- 2値
 - 0と1で明暗を示す
- グレースケール
 - 輝度を諧調表現する
- カラー
 - 16色
1980年代はこれが普通だった
 - 256色
1990年初めはこれが普通だった
 - High Color(16bitカラー)
65536色
 - True Color(24bitカラー)
1677万色
- カラー表現ではビット数以外にも次の様々な表現手法がある
- RGB
 - 光の3原色、赤、緑、青、それぞれで輝度を諧調表現
- YUV
 - 白黒からカラーへテレビ放送が映った際に両方で視聴できるように採用
 - 輝度信号Y、色差信号U(B-Y、Cb)、色差信号V(R-Y、Cr)で構成する
 - 人間の目は輝度変化に敏感かつ彩度変化に鈍感なため、色度よりも輝度により広い帯域を与えることで圧縮を効率化できる



その他の色の表現

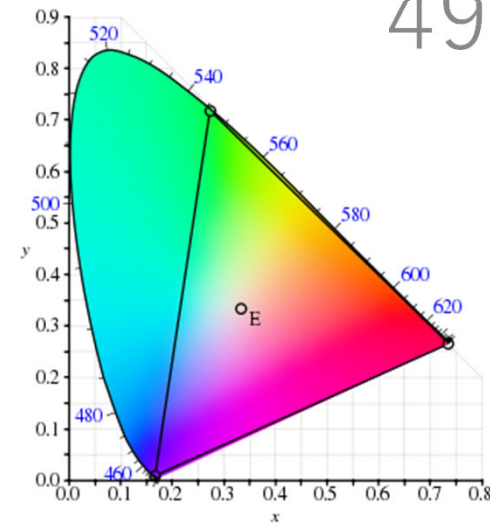
49

- CMY

- RGBと同様 3 元色の混ぜ合わせであるが、これは減色混合で全部混ぜると黒になる
- プリンタにおいて用いられる表現

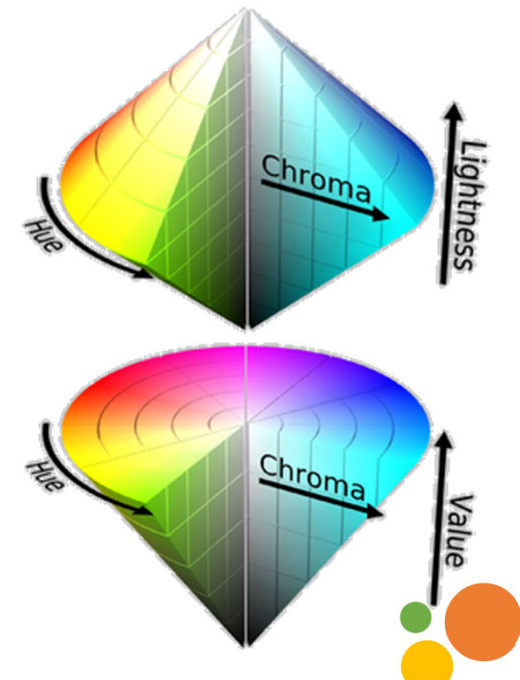
- CIE chromaticity diagram

- 右図がCIEダイアグラムで 太線に囲まれた領域が可視光領域
- ある 2 色(2 点)についてそれらを混合した色は 2 点を結ぶ線分の内分点になる
 - 外周曲線は純色でありその周波数が記載されている。中心は白となる



- HSI/HSV/HSB/HSL

- 人間の感覚によく適合する表現
- Hue(色)、Saturation(色の濃さ)、最後はIntensity(明るさ)、Value・Brightness・Lightnessとも表現される
- この色空間はdouble coneで表される






YUVにおける呼び方

50

- YUVyuv
 - Y:U:Vをy:u:vでサンプリングする。YUV444は、24bitで各8bitずつ
 - YUV422、YUV411等がある
- YUVs
 - 8bitでサンプリングしたときの1画素あたりのデータ量をsで示す
 $s = (8y + 8u + 8v) / y$
 - YUV12、YUV9等がある
- YUV420、YUV410
 - YUV420は、2x2のブロックでUとVが1サンプル
 - YUV410は、4x4のブロックでUとVが1サンプル





YUVの格納フォーマット

51

- YUV422を格納する場合
 - UYVY(UYNV) U0, Y0, V0, Y1の順に格納するフォーマット
 - YUY2(YUNV V422) Y0, U0, Y1, V0
 - YVYU Y0, V0, Y1, U0
- YUV411を格納する場合
 - IYU1 U0, Y0, Y1, V0, Y3, Y4 (IEEE1394 Digital Camera 1.04規格)
- YUV444を格納する場合
 - IYU2 U0, Y0, V0, U1, Y1, V1 (IEEE1394 Digital Camera 1.04規格)
- MPEG-1(YV12)では、
 - YUV420についてY, V, Uで配置している





RGBとYUVの変換

52

- YUVおよびRGBの値の範囲について
 - Yは、0と255は同期信号として利用するため、1～254
 - U、Vは、128を無彩色とし、16～240で表現
 - RGBは、ITU-Rで16～235でサンプリングするように規定されているが、これを守る場合も守らない場合もある
- たとえば、
 - RGB から YCbCr への変換
$$Y = 0.257R + 0.504G + 0.098B + 16$$
$$Cb = -0.148R - 0.291G + 0.439B + 128$$
$$Cr = 0.439R - 0.368G - 0.071B + 128$$
 - YCbCr から RGB への変換
$$R = 1.164(Y-16) + 1.596(Cr-128)$$
$$G = 1.164(Y-16) - 0.391(Cb-128) - 0.813(Cr-128)$$
$$B = 1.164(Y-16) + 2.018(Cb-128)$$

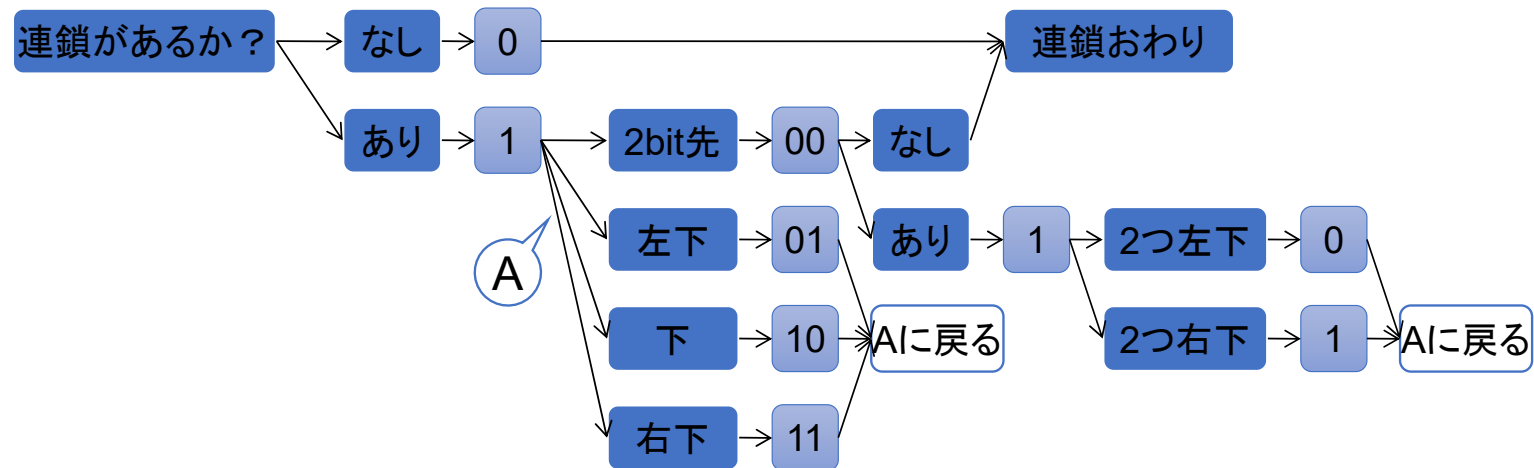
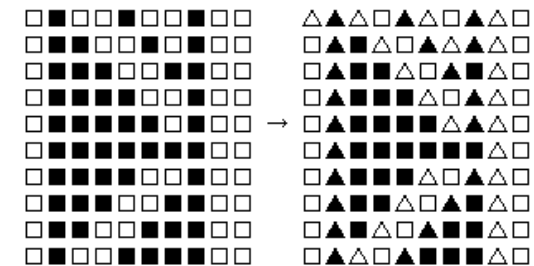




PICフォーマット

53

- 日本で編み出された画像圧縮方式
 - 当時はFAX以外、画像圧縮がそれほど存在しなかった
- 基本的にはランレングス法、正し縦方向にも拡張
- 手順
 - 色の変化点をチェック（図中の三角印）
 - 同じ色の変化点を探していく
 - 探すのは下方向で左右2bit







例えば、2つ目の黒は、11010101010101010000



PICの性能

54

- 自然画には適さず、人工的な絵は良い圧縮率を示す
- 少なくとも、ランレングス法よりは優れる。

画像名	ランレングス 法圧縮率	PIC法圧縮率	
image1	3.90%	0.82%	
image2	200.00%	35.99%	
image3	188.74%	149.29%	
image5	5.90%	3.29%	





減色・パレット化

55

- 減色
 - たとえば、24bit TrueColor(8x3)から、16bit HighColor(5x3)に変換する場合、単純に下位3bitを省略すればよい
 - 減色アルゴリズム
R:10010010 G:11010010 B:11001001ならば、
R:10010 G:11010 B:11001 とする
- パレット
 - 昔はメモリーが非常に高価であったことから、フルカラーを表現したいがメモリーの占有面積は減らしたいという要求があった
 - そこで、フルカラーのうち256色だけを利用して、管理は256種類の色番号にするという方法（昔のX11R5やSUN WSはこの方法を標準としていた）
 - カラーパレット番号0は黒、1は白とおおよそ定められており、残り254種類はフルカラーから自由に色を指定できる
 - パレット化アルゴリズムは木構造LRU法、マップ法など様々あるが、計算途中には膨大なメモリーが必要であり、当時としては実は本末転倒な面も






演習問題（6）

56

- Processingを用いて画像を圧縮、圧縮後のサイズと圧縮率を表示した後、圧縮された画像から元の画像を再現しなさい
 - 好きなイメージ（300x300から500x500）の画像を準備する
 - カラーでも2値でもよい
 - 圧縮手法は自由でよいが、与えた画像に対して必ず圧縮されている、つまり圧縮率は100%未満であることが必要である
 - ある特定の画像だけ圧縮できる手法ではなく、なるべく汎用性のある手法であることが望ましい。
 - 例えば、基準配点（合格点）を狙うのであればランレングス法がシンプルでよいであろう
 - 次回の内容となるが、差分符号もシンプルである





注意事項

- これらの設問に対する回答を、Microsoft Wordファイルで作成
- LMSで提出すること
- A4で作成すること
 - どのような入力値を使ったかや、実行結果を示せば十分です。なお、常識的な範囲でページ数を超過することは問題ありません
- **ソースコード、動作画面をキャプチャして貼り付けること**
 - キャプチャはキャプチャしたいWindowを選択してAlt+Print Screenでできます。そのまま、Wordファイルに張り付けることができます（+は押しながらの意味）
 - MacOSは、Control+Command+Shift+4とするとカーソルがカメラ型に変わります。特定のウインドウやメニューバーをクリックしてキャプチャ、画像はクリップボードに転送されます
 - 記述したソースコードをレポートに張り付けてください。
- 最初にタイトルとして「演習問題（6）」と書き、名前と学籍番号を記載すること
このフォーマットに従っていないレポート答案は受け取らない
- 締め切りなど詳細はLMSを確認すること

