

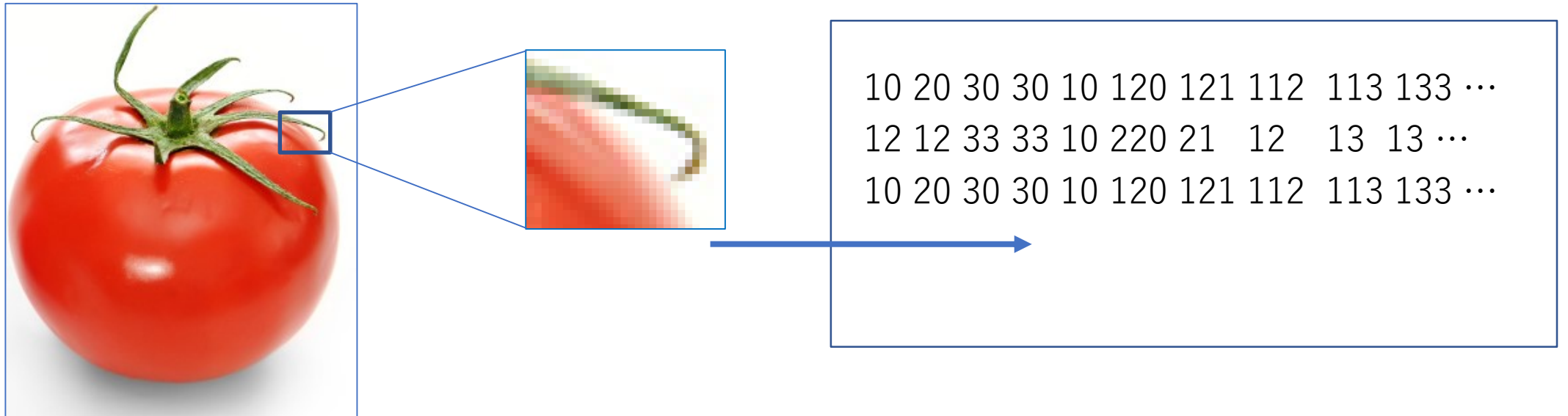
行列による画像処理

計算機演習 担当：劉 雪峰

2019年7月29日

画像ファイルと行列

- 画像ファイルはピクセルから構成される。例えば、 200×200 の画像は40000のピクセルがある。
- 実際の画像処理では、ピクセルの色の値は行列に格納される上、**行列の演算**によって画像処理が実施される。



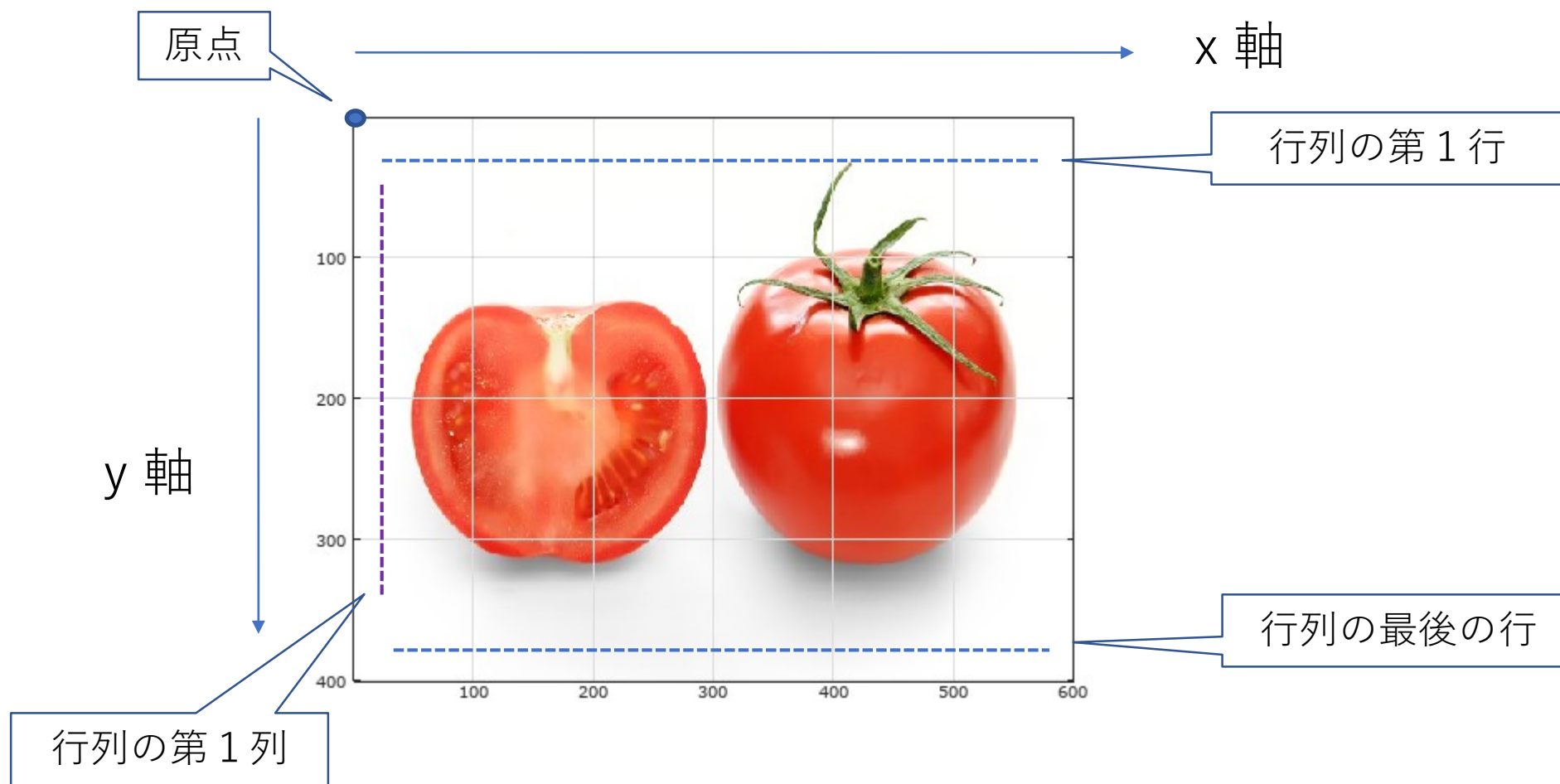
画像のピクセルと行列の成分

- カラーモードの画像ファイルの場合、各ピクセルの色の表現では、RGB三色の値を使用している。RGBの値を格納するために、3次元の行列が必要となる。
- 例： $M \times N$ の画像ファイルのとき、 $M \times N \times 3$ の行列が必要となる。
- 画像の行列 A の各成分
 - $A(:, :, 1)$: 赤
 - $A(:, :, 2)$: 緑
 - $A(:, :, 3)$: 青
 - 各色の値の範囲はそれぞれ0~255である。
- 例えば、 $A(12,30,1)$ は12行目13列目のピクセルの赤色の値を示している。

Octaveの画像処理の命令

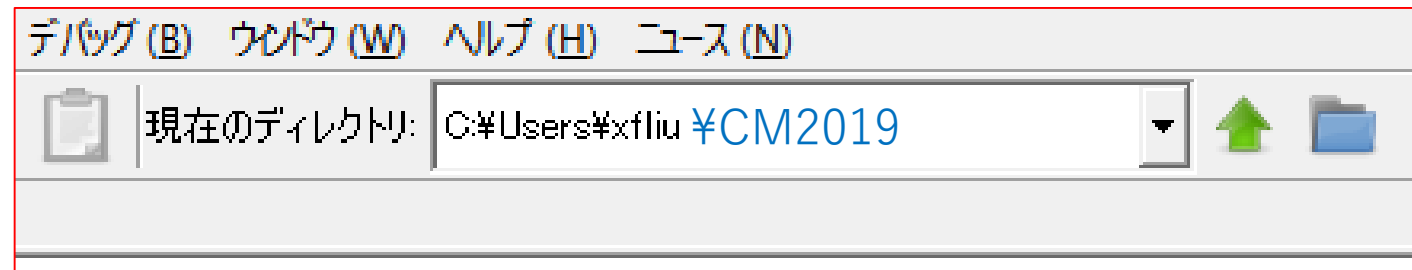
- 画像ファイルをロードして、行列 x に格納する。
 - `[x,map,alpha]=imread(“画像ファイルの名前”);`
- 画像の行列 x を表示する。
 - `image(x)`
 - `imshow(x)`
- 画像の行列をファイルに出力する。
 - `imwrite(x,“ファイル名”)`
- 画像の行列は普通の行列と同じのように演算できる。

画像におけるピクセルの座標系



演習 1

- 授業のHPからトマトの写真をダウンロードする。
 - ファイルの保存場所をC:/Users/[アカウント名]/CM2019/とする。
 - ファイル名を[tomato.jpg]とする。
- 計算機にインストールされるOctaveを起動して、現在のディレクトリを設定してください。



演習 1（続き）：起動後のOctave

The screenshot shows the Octave 4.2.1 interface. A red rectangle highlights the '現在のディレクトリ' (Current Directory) field, which displays 'C:\Users\xfliu'. A callout points to this field with the text '現在のディレクトリ'. To the left, a callout points to the file browser pane with the text 'ファイルのリスト'. The file browser shows a list of files and folders in 'C:/Users/xfliu'. The command window on the right shows the Octave startup message and the prompt '>> |'. A callout points to the command window with the text 'ここに命令を入力する。'. The command history pane at the bottom left shows a list of commands entered, including 'a=[2,0,0;1,1,0;0,2,0;1,0,1;0,1,1;0,0,2]', 'b=[2,0,0;1,1,0;0,2,0;1,0,1;0,1,1;0,0,2]', 'c=[2,0,0;1,1,0;0,2,0]', 'd=[a;b;c]', '[(1:15)', 'd]', and 'exit'.

現在のディレクトリ

ファイルのリスト

ここに命令を入力する。

演習 1 （続き）

- 以下の命令を実行しなさい。
 - `[x,map,alpha]=imread("tomato.jpg");`
 - `imshow(x,map)`
 - `size(x)`
- 行列xの R G B 成分をそれぞれ変数xr,xg,xbに格納する。
 - `xr = x(:, :, 1); xg = x(:, :, 2); xb = x(:, :, 3);`
 - `size(xr); size(xg); size(xb);`
- xr,xg,xbをそれぞれ画像ファイルとして表示しなさい。
 - `imshow(xr,map)`
 - `imshow(255-xr,map)`

RGB画像とグレイ画像

- RGBの各色の値をベースして、RGB画像ファイルをグレイ画像に変換することができる。
- たとえば、以下のコードはRGBの平均値を使って、グレイ画像を作成している。

```
gray_x = (x(:,:,1)/3 + x(:,:,2)/3 + x(:,:,3)/3) ;  
imshow(gray_x)
```

平均値の計算はmeanを使ってもよいです。すなわち：

- `gray_x = mean(x,3);` % 3次元目のR,G,Bの平均値を計算する。

注意：Aのデータは符号なし 8 bit整数であるので、各成分の値の範囲は0~255である。上記のコードを以下のように計算した結果は予測できない。三つの成分の値は2 5 5を超える可能性がある。

- `gray_x = (A(:,:,1) + A(:,:,2) + A(:,:,3))/3 ;` (よくない計算例)

演習 2 : さまざまな画像の加工

画像行列xを利用して、以下の画像の加工を試してみてください。

- 色の反転
 - `nx=255-x`
- 画像の一部を取る。
 - `nx = x(:, 1:300, :)`
- 画像ファイルのサイズを半分にする。
 - `nx = x(1:2:end, 1:2:end, :);`
 - `imwrite(nx,"tomato_half.jpg");`

上記のコードで得られる行列を画像として表示する：

- `imshow(nx,map);`

演習 3 :

- トマトの写真にある二つのトマトをそれぞれ取って、tomato_left.jpg, tomato_right.jpgに保存してください。
 - トマトの周りの余白を取り外すこと。

(チャレンジ)

- 以下の写真サイズの変更を考えてください。
 - 写真のサイズを2/3にする
 - 写真のサイズを2倍にする。

「ヒント」特に、写真を2倍するとい、もともとのピクセルの間の距離が大きくなって、そのピクセルの間に新しいピクセルを作成するのは必要である。

演習 3 のヒント

写真のサイズを2/3にすること

- 元々の行列Aのサイズは 400×600 である。
- Aのサイズの2/3になる行列B(266×400)を作成するために、以下のIDEAを参考しなさい。
 - Aの部分行列を使って、Bを作成する。
 - Aの行の一部を取るために、
 - Bの行の番号1:266
 - Aの行の番号1:400
 - Bの行の番号からAの行の番号までの変換を以下の式で計算できる。
$$1:266 \Rightarrow \text{int32}((1:266)*3/2)$$
 - 同じように、列の一部をとるために、以下の対応関係を利用する。
$$1:400 \Rightarrow \text{int32}((1:400)*3/2)$$
- よって、Bは次のように作成する。
 - $B = A(\text{int32}((1:266)*3/2), \text{int32}((1:400)*3/2), :)$

演習 4

- トマトの写真に対して、自由な加工を考えてください。
- たとえば、トマトを回転したり、複数にコピーしたりすること。
- 授業で提供しているトマトの以外の写真も自由に使ってください。

行列を画像ファイルに変換する

- 行列を画像として描画するとき、行列の成分を「符号なし8bitの整数」に変換するのは必要である。
- たとえば、乱数で作成した行列を画像として描くとき、以下のコードを参照しなさい。
 - `A=rand(100,100)*255;`
 - `imshow(uint8(A));`
 - `u: unsigned int : integer 8: 8bit`