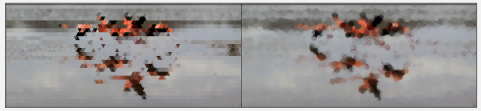
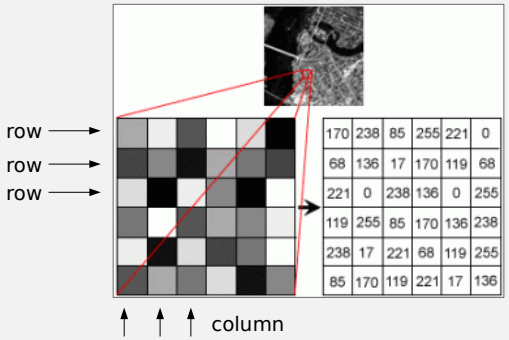
Images as a Source of Colors



- instead of (or in addition to) displaying an image, the image can be used as a source of colors for drawing shapes (rect, ellipse, triangle, etc)
  - imagine the shapes are being drawn on top of the image – use the color of the image at that spot as the color of the shape
  - the image itself doesn't have to be displayed

Images and Pixels

- an image is a 2D grid of pixel colors



https://naushadsblog.wordpress.com/2014/02/10/image-processing-and-computer-vision-in-java-point-operators-part-2histogram-equalization/

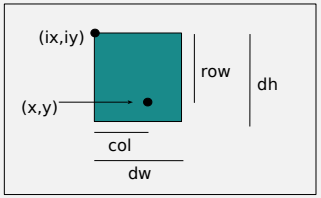
Using Images as a Source of Colors

Two steps –

- step 1 – figure out which image pixel we want the color of
  - imagine the image underlying (part of) the drawing window
  - convert coordinate (x,y) in the drawing window into a pixel coordinate (row,col) in the image
- step 2 – access the color of image pixel (row,col)

important: to avoid confusion, use (x,y) to refer to coordinates in the drawing window and (row,col) to refer to coordinates within an image

From (x,y) To (row,col)

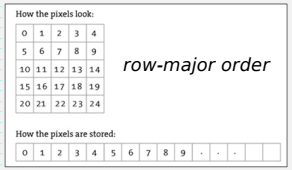


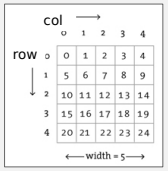
important: to avoid confusion, use (x,y) to refer to coordinates on the screen and (row,col) to refer to coordinates within an image

- assumptions/notation
  - image has size *img*`.width` × *img*`.height`
    - replace *img* with the name of the image variable
  - image is positioned in the drawing window with corner (`ix`,`iy`) and dimensions `dw` × `dh`
- to transform a drawing window coordinate (`x`,`y`) into an image pixel coordinate (`row`,`col`) –
  - `row = (y-iy)*img.height/dh`
  - `col = (x-ix)*img.width/dw`

Accessing Pixel Colors

- load pixels so they can be accessed
  `img.loadPixels();`
- pixels are stored in the array `img.pixels`
  - pixel at position (`row`,`col`) is stored in slot
    `loc = row*img.width+col`





- `img.pixels[loc]` is of type `color`
  - can use as is for `fill`, `stroke`
  - access color components with `red(img.pixels[loc])`, `green(img.pixels[loc])`, `blue(img.pixels[loc])`

https://processing.org/tutorials/pixels/


```
PImage img;          // the image to display

void setup () {
  size(420,640);

  // load image
  img = loadImage("marmot.jpg");

  // load image pixels so they can be accessed
  img.loadPixels();
}

void draw () {
  background(0);

  // draw image  takes the whole window
  image(img,0,0,width,height);

  // draw colored spot following the mouse - color is taken from the image pixel
  //  under the mouse
  {
    // (row,col) corresponding to the mouse's location in the image
    int row = (mouseY-0)*img.height/height;
    int col = (mouseX-0)*img.width/width;
    // location in pixels array corresponding to (row,col)
    int loc = row*img.width+col;

    fill(img.pixels[loc]);
    stroke(0);
    ellipse(mouseX,mouseY,40,40);
  }
}
```

declare a variable for the image

load the image

load the image pixels so they can be accessed

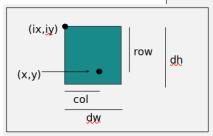
(ix,iy)

dw, dh

compute (row,col) in image corresponding to (x,y) in the window

compute the slot in the pixels array corresponding to (row,col)

get the color of that pixel




```
row = (y-iy)*img.height/dh
col = (x-ix)*img.width/dw
```