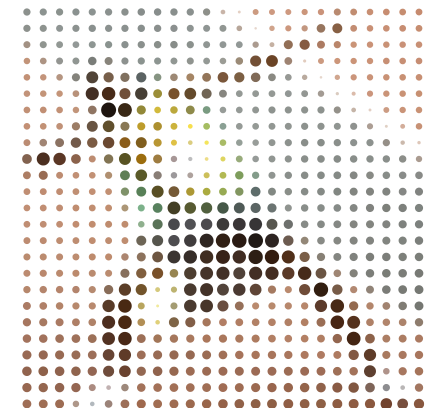


Announcements

- A2 resubmissions due today
- A3 released; due Oct 18
- Next (and last) lab: A3 work session
- E-mail me to schedule retest

Today's topic: image processing

- Image functions
- Type `uint8`
- 3D arrays
- Filtering



Arrays: What we didn't tell you

```
m= [ 2  3  5  7;  
    -2 1  0  7;  
     5 2 -1  8 ]
```

```
L= m>3  
[ 0 0 1 1;  
  0 0 0 1;  
  1 0 0 1 ]
```

```
P= m>3 | m<0  
[ 0 0 1 1;  
  1 0 0 1;  
  1 0 1 1 ]
```

```
a= m(m>3)  
[ 5; 5; 7; 7; 8]
```

```
[ 2  3  5  7;  
 -2  1  0  7;  
  5  2 -1  8 ]
```

```
b= (m>3) .* m  
[ 0 0 5 7;  
  0 0 0 7;  
  5 0 0 8 ]
```

Column-major storage

```
m= [ 2  3  5  7;  
     -2 1  0  7;  
     5  2 -1  8 ]
```

```
m(5) % Linear index  
     1
```

```
v= m(:) % Convert to vec  
[ 2; -2; 5; 3; 1; 2; 5;  
 0; -1; 7; 7; 8 ]
```

```
% Recover matrix  
mm= reshape(v,3,4);
```

```
% Reductions act on  
columns
```

```
cmax = max(m)  
     [ 5 3 5 8]
```

Permutation indexing

- **Logical indexing:** $a=v(p)$
 - $p(i) \in \{\text{false}, \text{true}\}$
 - $\text{length}(p) == \text{length}(v)$
 - $\text{length}(a) == \text{sum}(p)$
- **Vector indexing:** $a=v(x)$
 - $x(i) \in 1:\text{length}(v)$
 - $\text{length}(a) == \text{length}(x)$
- **Permutation**
 - $\text{length}(x) == \text{length}(v)$
 - $x(i)$ are unique

- **Shuffling**
 - $x = 2*(1:5);$
 - $y = \text{randperm}(\text{length}(x))$
[2 4 5 3 1]
 - $z = x(y)$
[4 8 10 6 2]
- **Sorting**
 - $[w, p] = \text{sort}(z)$
[2 4 6 8 10]
[5 1 4 2 3]
 - $w == z(p) == x$

Searching

- Is x in v ? If so, where?
 - Need to check every element of v
 - Vectorized assistant: `find`

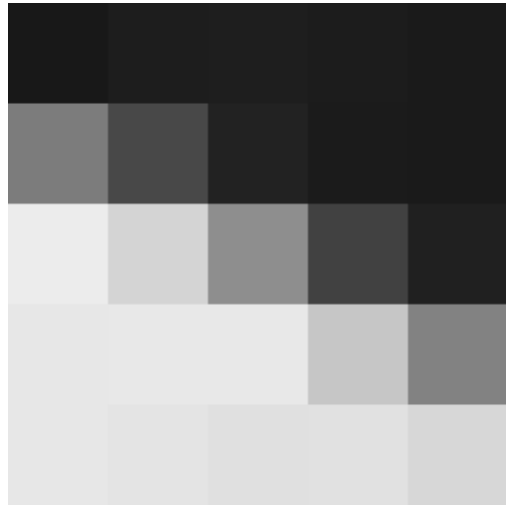
```
ids = [89, 4, 782];  
names= {'Curran', ...  
        'Aravind', ...  
        'Clive'};  
  
names{find(ids == 4)}
```

- What if v is sorted?
 - Can eliminate half of v without looking at its contents: “binary search”
 - Repetitions needed to check whole array: $\log_2(\text{length}(v))$
 - Even if v has 1000 sorted elements, searching takes only 10 steps
- **Recommendation:** If doing lots of searches, or if data is naturally sorted, avoid `find`, linear search

New topic: `uint8` and image processing

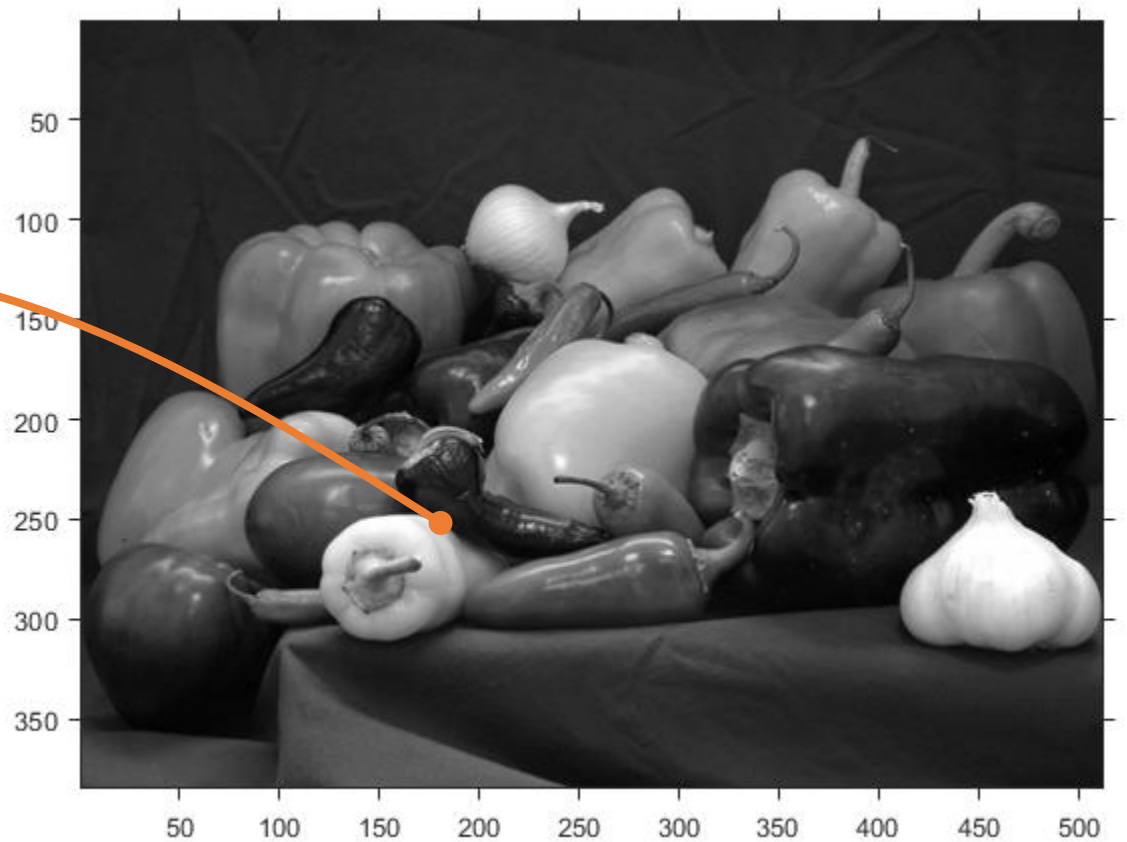


Pictures as matrices



```
img(248:252, 174:178)
```

24	29	30	28	26
124	72	34	27	26
236	212	142	65	32
231	232	232	198	130
231	228	224	225	215



“512 x 384” image \Rightarrow 384 x 512 array

Image files & raster data

File formats

- JPEG: Photographs, lossy
- PNG: Graphics, lossless
- TIFF: Technical

Others

- WebP, GIF, DNG, OpenEXR, ...

Properties

- Channels
 - RGB(A), YCbCr
- Bit depth, range
 - 8-bit, 10-bit, HDR
- Color space, “gamma”
 - sRGB, DCI-P3, raw
- Subsampling
 - 4:4:4, 4:2:0

MATLAB features

- % Read image file into matrix
`mat = imread('filename')`
- % Plot matrix as image
`imshow(mat)`
- % Write matrix to image file
`imwrite(mat, 'filename')`

New type: `uint8`

- Integer value between 0 and 255
 - 0=dark, 255=bright
- Can only operate with other `uint8` vars, or scalar doubles
- Semantics: rounding and saturation (different from most other languages)
- Conversion: `y = uint8(x)`

Example: Draw crosshairs

Objective

- User clicks on image
- Draw horizontal and vertical lines through clicked point
 - Lines replace image pixels



Tools

- `ginput(n)`
- `uint8` assignment
- For-loops, slice assignment