



SKILLPILLS

MATLAB

<< Computer Image Analysis >>

Ray X Lee
李昕睿

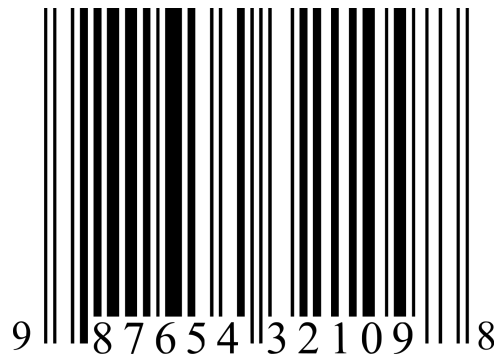
*Biological Physics Theory Unit &
Optical Neuroimaging Unit*

October 10th, 2017



Purposes and Applications

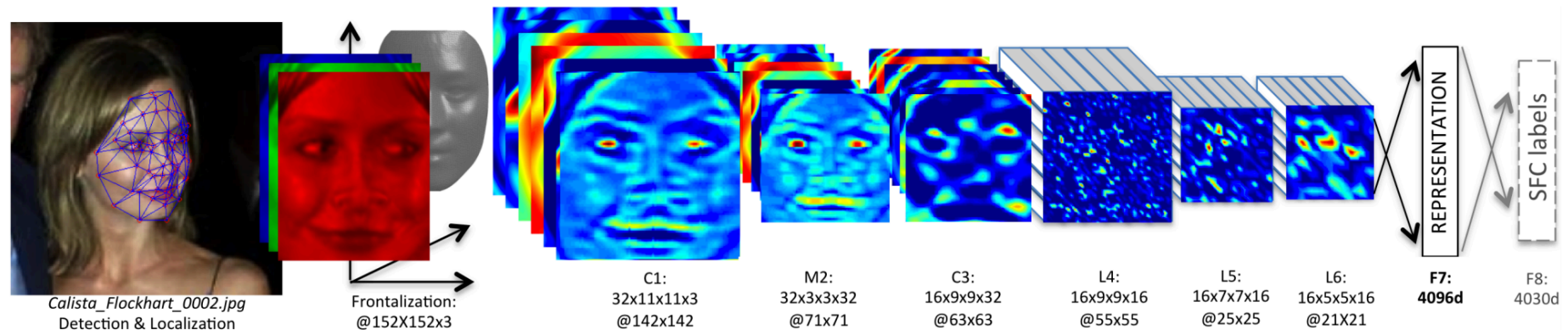
- Extract meaningful information from images



Universal Product Code (UPC)



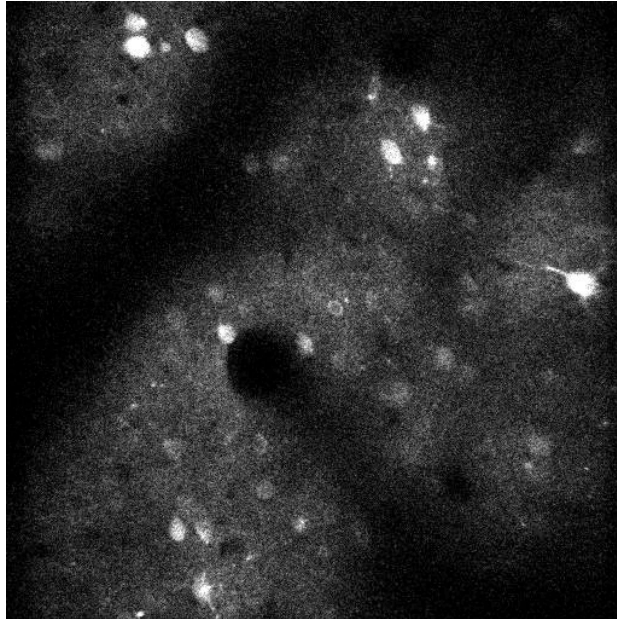
Quick Response (QR) Code



Facial Recognition

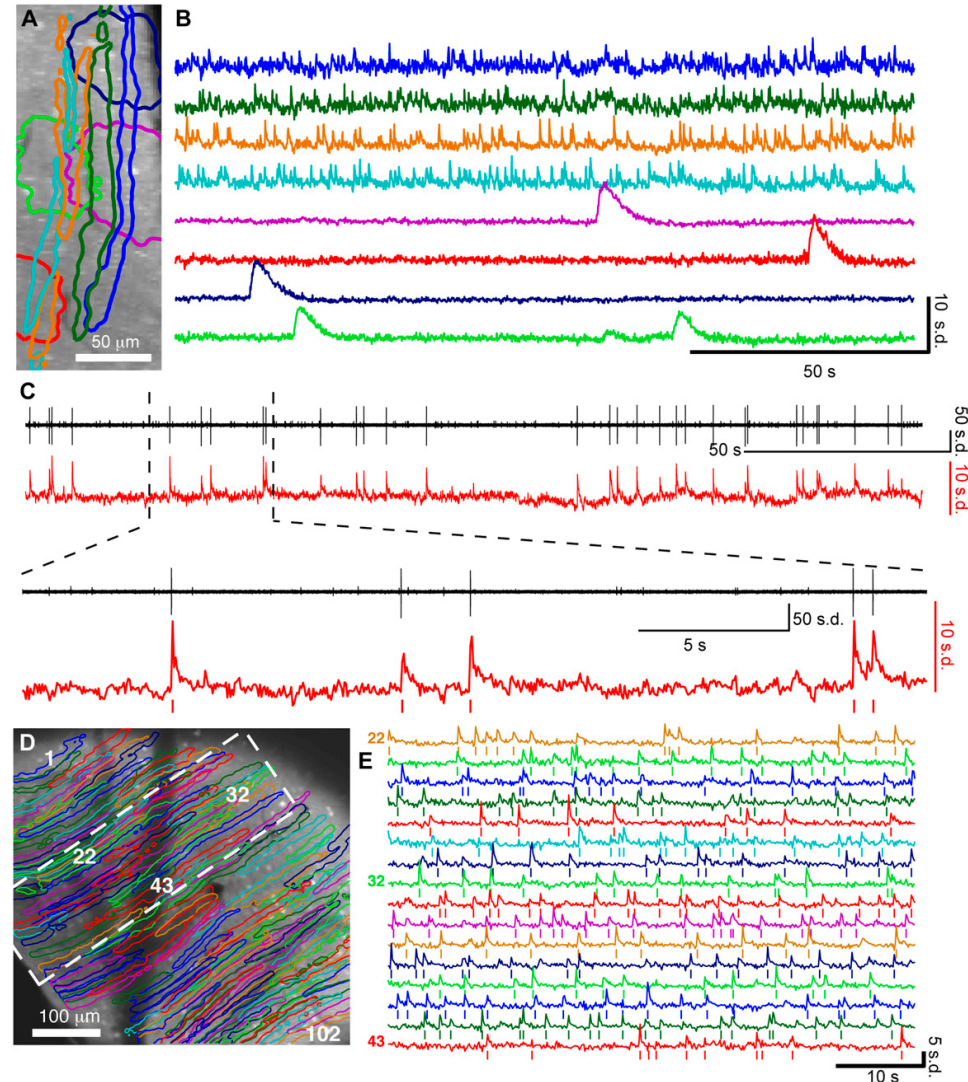
Taigman Y *et al* (2014) CVPR

- Extract meaningful information from images



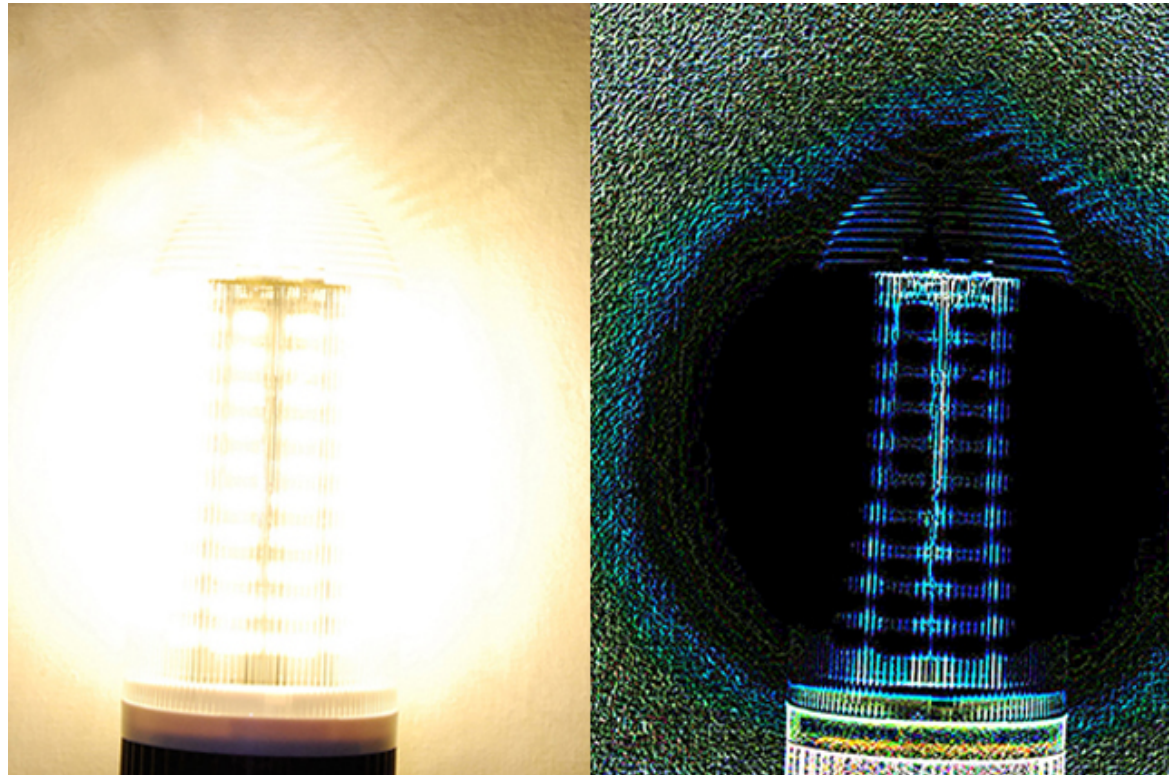
Demo not shown

(unpublished data)





Mukamel EA et al (2009) Neuron

- Brightness, Colors, and Geometric Transformations
- Segmentation
- Filtering
- Feature Detection
- Feature Matching



Phase stretch transform (PST)

FeaturesBusinessExploreMarketplacePricingXXX processSign in or Sign up

Search: File ExchangeRay | My Community Profile | Log Out

ProductsSolutionsAcademiaSupportCommunityEvents

File Exchange

MATLAB CentralFilesAuthorsTags

Search FilesAdvanced SearchXXX process

<https://www.mathworks.com/matlabcentral/fileexchange>

Languages

Python	19,611
Java	11,789
JavaScript	10,749
R	7,839
C++	7,250
Ruby	6,426
HTML	4,580
PHP	4,343
Jupyter Notebook	4,150
Matlab	X



<https://www.google.com>

The background of the entire slide is filled with numerous small, red and white capsules scattered in various orientations.

SKILLPILLS

Before getting start...



Array logical operations

```
A = randi(15,8,6);
```

```
A == 9
```

```
B1 = A<9
```

```
B2 = A(B1)
```

```
Ind = find(A<9)
```

```
B3 = A(:); B3 = B3(Ind)
```

```
[IndX,IndY] = ind2sub(size(A),Ind)
```

1	4	7
2	5	8
3	6	9



1,1	1,2	1,3
2,1	2,2	2,3
3,1	3,2	3,3

```
B4 = A(A<9 & A>2)
```

```
B5 = A; B5(B5<9) = 0
```

```
B6 = B5; B6(B6==0) = NaN; B6(~isnan(B6)) = 10
```

```
B7 = A; B7(B7(:,2)<9,3) = NaN
```

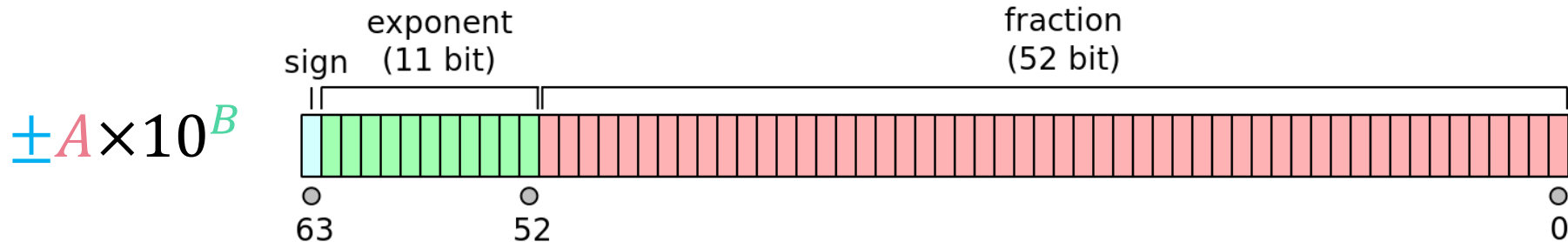
```
B8 = A; B8(B8(:,2)<9,:) = []
```

```
C1 = unique(A)
```

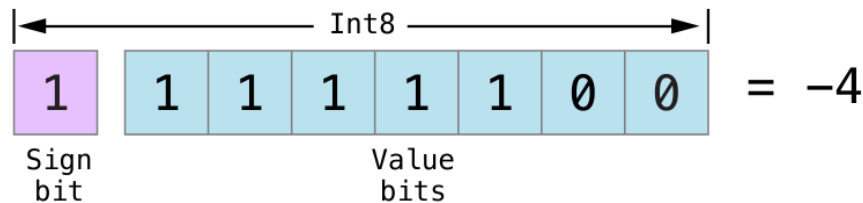
```
C2 = ismember(A,[1 3 5 7])
```



Double-precision floating-point format (64 bits)



Signed integer (8 bits)



$$(2^6 + 2^5 + 2^4 + 2^3 + 2^2) - 2^7 = 124 - 128 = -4$$

'single'	Single-precision number
'double'	Double-precision number
'int8'	Signed 8-bit integer
'int16'	Signed 16-bit integer
'int32'	Signed 32-bit integer
'int64'	Signed 64-bit integer
'uint8'	Unsigned 8-bit integer
'uint16'	Unsigned 16-bit integer
'uint32'	Unsigned 32-bit integer
'uint64'	Unsigned 64-bit integer

```
A{1,1} = rand(8,6);  
A{2,1} = rand(3,7);  
A{1,2} = 'Hello';  
A{2,2} = rand(2,5)>0.3;
```

```
A{1}.I = rand(8,6);  
A{2}.I = rand(3,7);  
A{1}.J = 'Hello';  
A{2}.J = rand(2,5)>0.3;
```

```
A.I = rand(8,6);  
A.J = rand(3,7);  
A.K = 'Hello';  
A.L = rand(2,5)>0.3;
```

```
A.I{1} = rand(8,6);  
A.I{2} = rand(3,7);  
A.J{1} = 'Hello';  
A.J{2} = rand(2,5)>0.3;
```

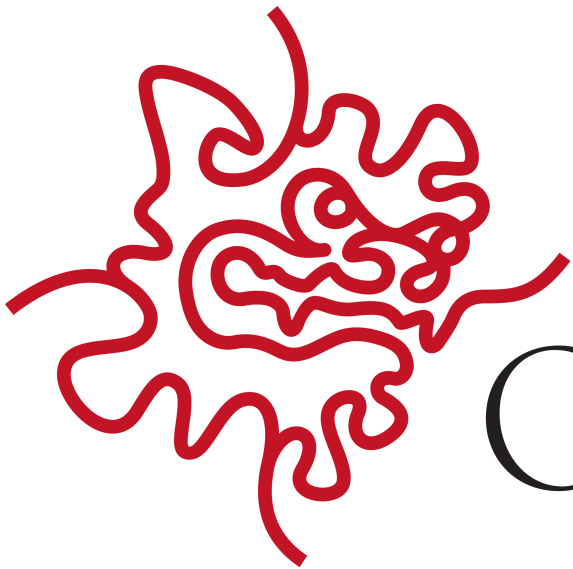
```
A(1,1).I = rand(8,6);  
A(1,2).I = rand(3,7);  
A(1,1).J = 'Hello';  
A(1,2).J = rand(2,5)>0.3;
```

```
A(1,3).J = nan(100,2);  
  
A(2,2).A = zeros(2,100);
```



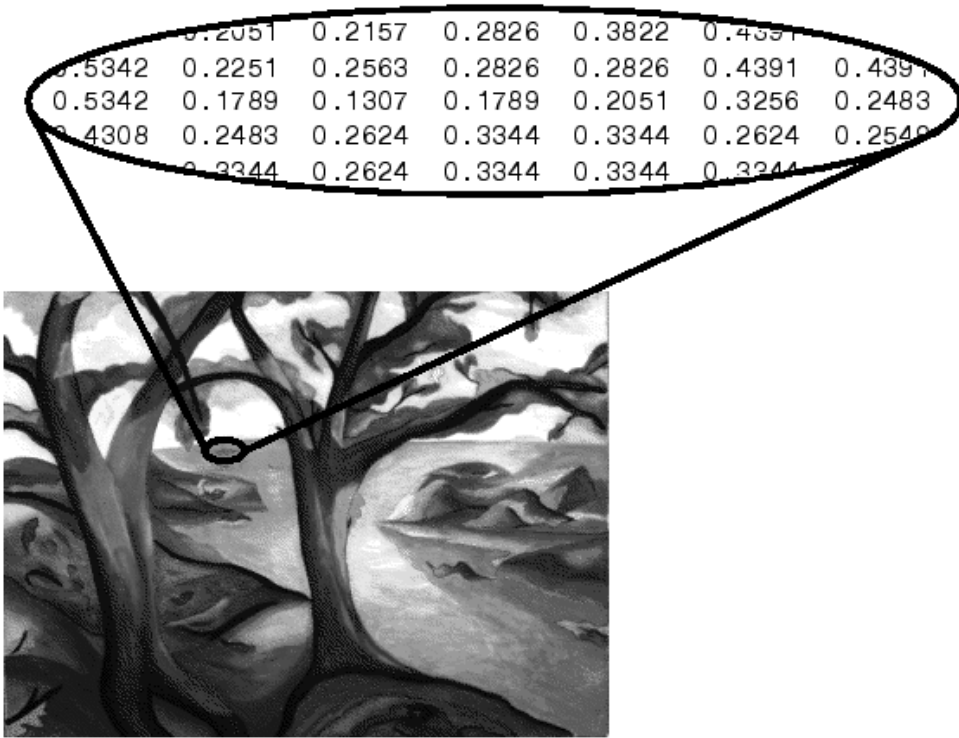
SKILLPILLS

Brightness, Colors, and Geometric Transformations



OIST

2-D matrix



3-D matrix

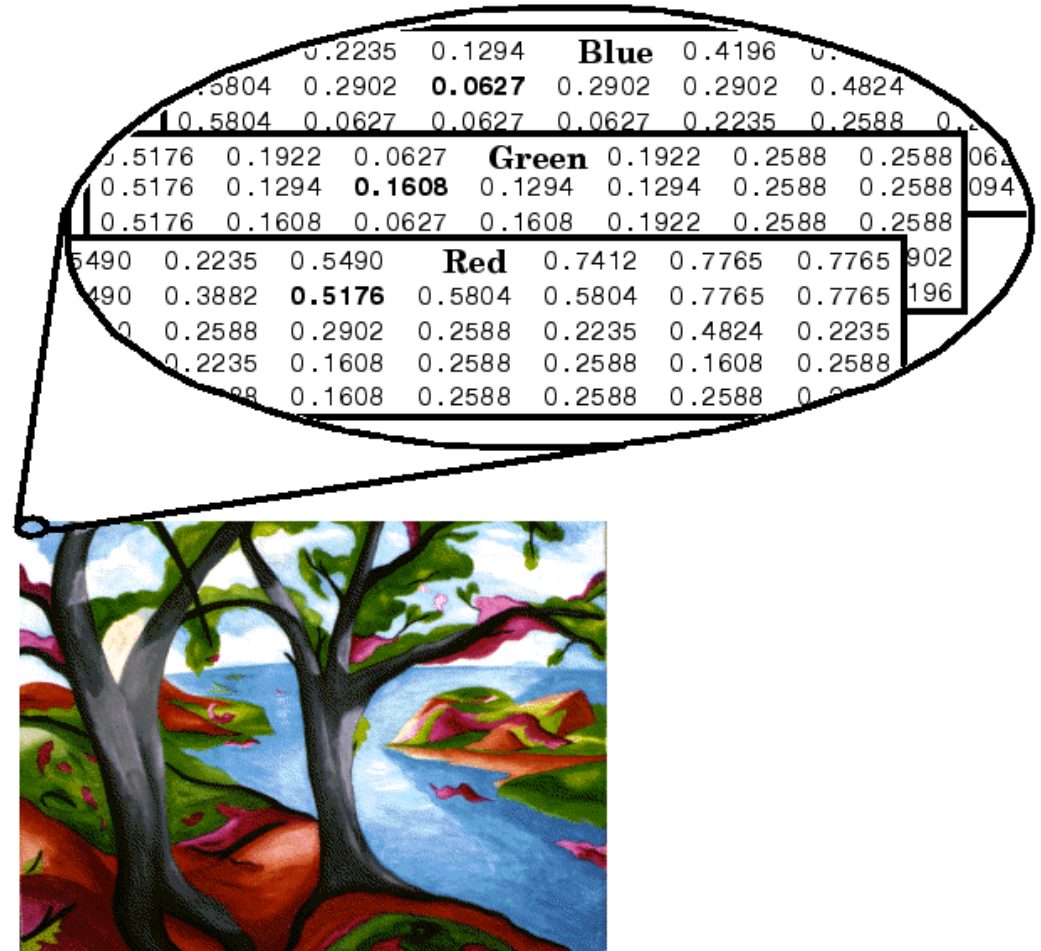


Image file

```
A = imread('ngc6543a.jpg');
```

```
imwrite(A, 'ngc6543a_2.jpg')
```

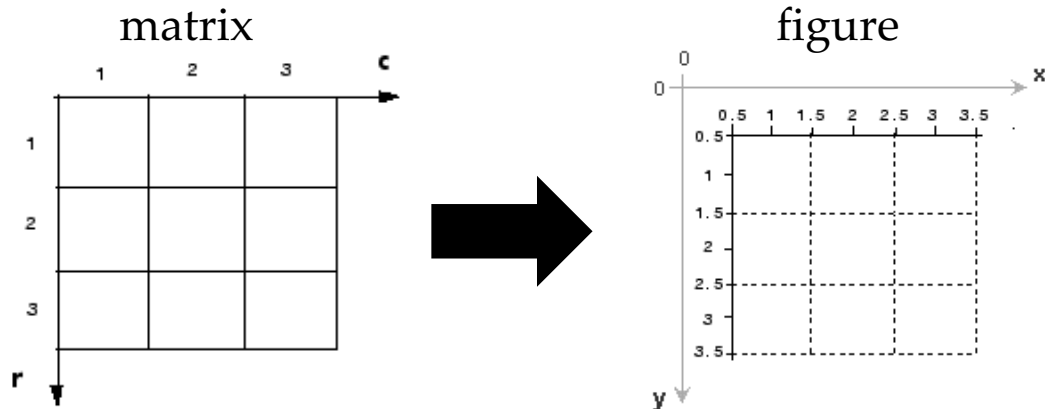
Video file

```
v = VideoReader('xylophone.mp4');  
c = 0;  
while hasFrame(v)  
    c = c + 1;  
    B(c).Frame = readFrame(v);  
end
```

```
v2 = VideoWriter('xylophone_2.mp4');  
open(v2);  
for frame = 1:length(B)  
    writeVideo(v2, B(frame).Frame);  
end  
close(v2);
```

Specifying locations in an image

```
imagesc(imread('corn.tif',2))
```



```
hold on  
plot(50,150,'g.','MarkerSize',50)
```



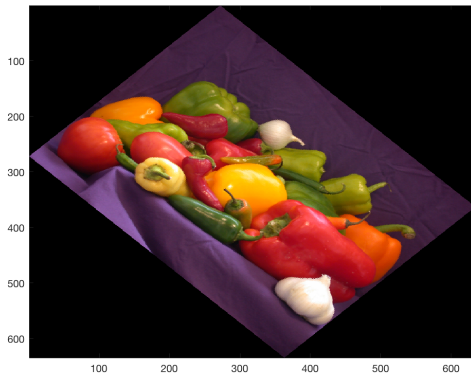
A



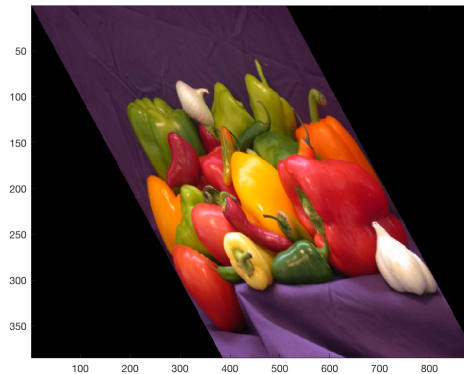
```
A2 = imresize(A,0.1);
```



```
A2 = imrotate(A,-45);
```



```
tform = affine2d([1 0 0;1 1 0;0 0 1]);  
A2 = imwarp(A,tform);
```



$$\begin{bmatrix} x_{\text{new}} & y_{\text{new}} & 1 \end{bmatrix} = \begin{bmatrix} x_{\text{original}} & y_{\text{original}} & 1 \end{bmatrix} T$$

- Create a 20-Hz SP03-01.avi video where the frames were 30°-clockwise-rotated images in the SP03-01.tif file.

Hint

1. `imread('File.tif',k)` reads the k^{th} image (total 141 images).
2. Frame rate is contented in `v2 = VideoWriter('File.mp4');`.

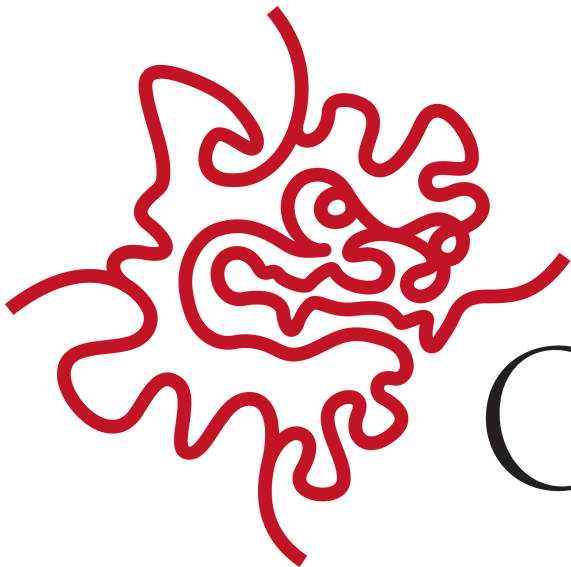
Further consideration

1. The total image number is usually unknown.



SKILLPILLS

Segmentation

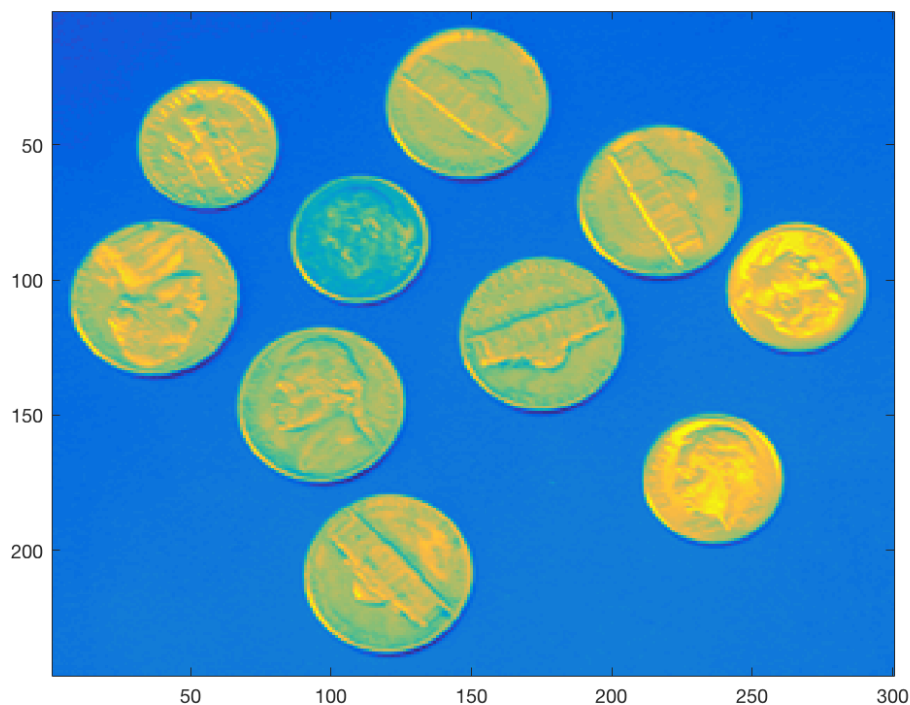


OIST

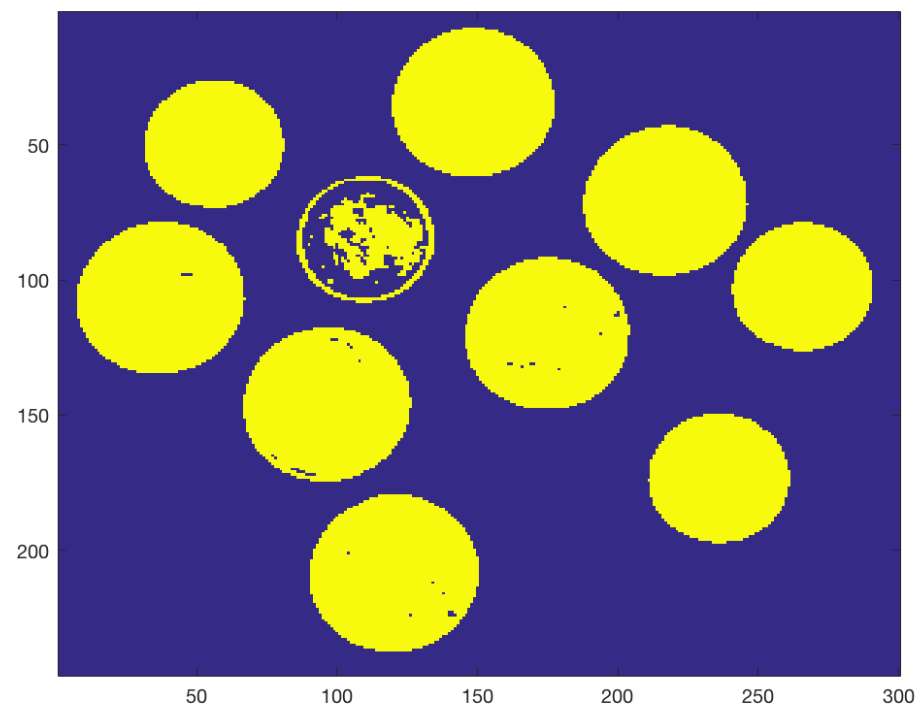
Binary Image Segmentation

```
A = imread('coins.png');  
BW = A; BW(BW<=max(BW(:))/2) = 0; BW(BW~=0) = 1;
```

A



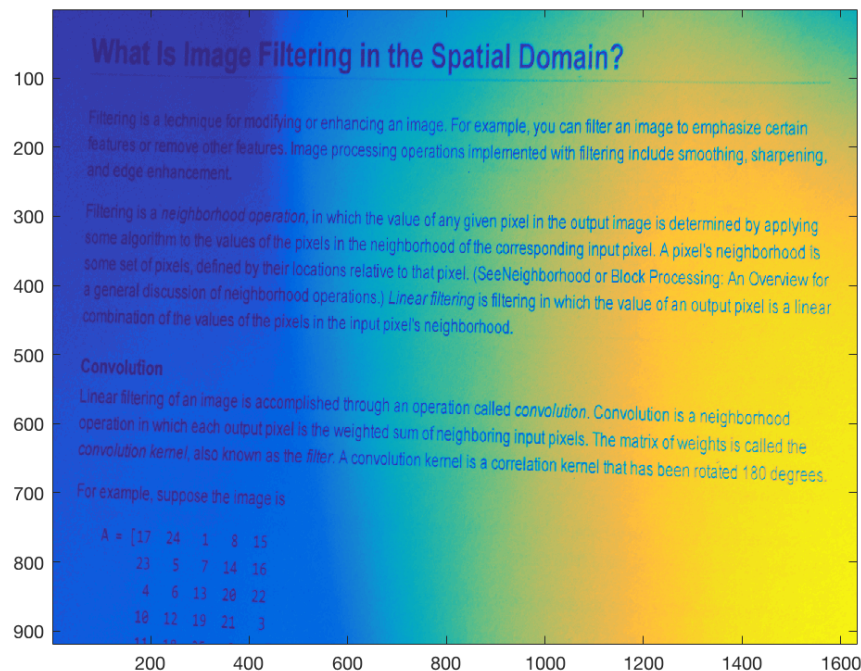
BW



Binary Image Segmentation

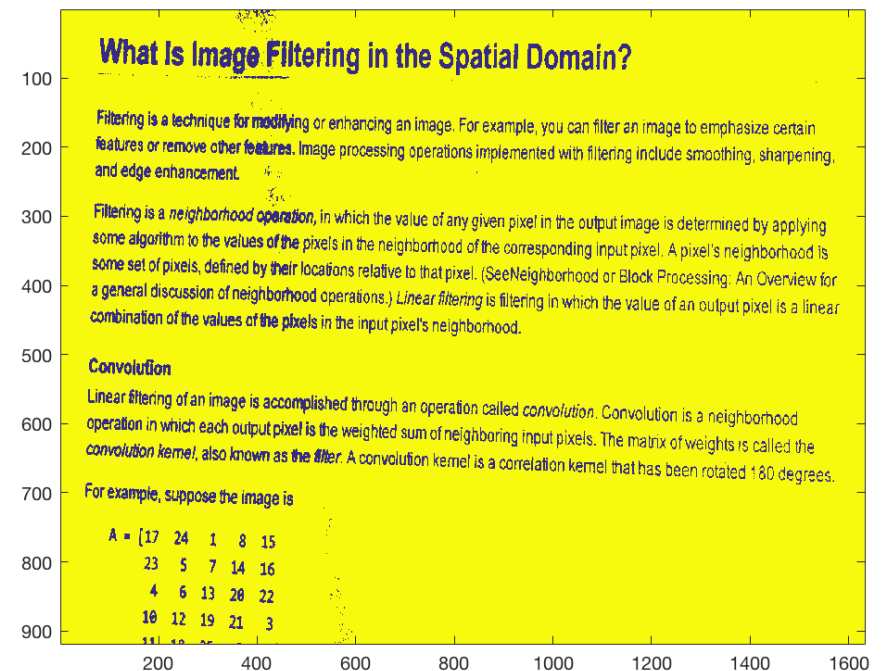
```
A = imread('printedtext.png');  
BW = imbinarize(A,'adaptive', ...  
                'ForegroundPolarity','dark', ...  
                'Sensitivity',0.4);
```

A



918x1632 unit8

BW

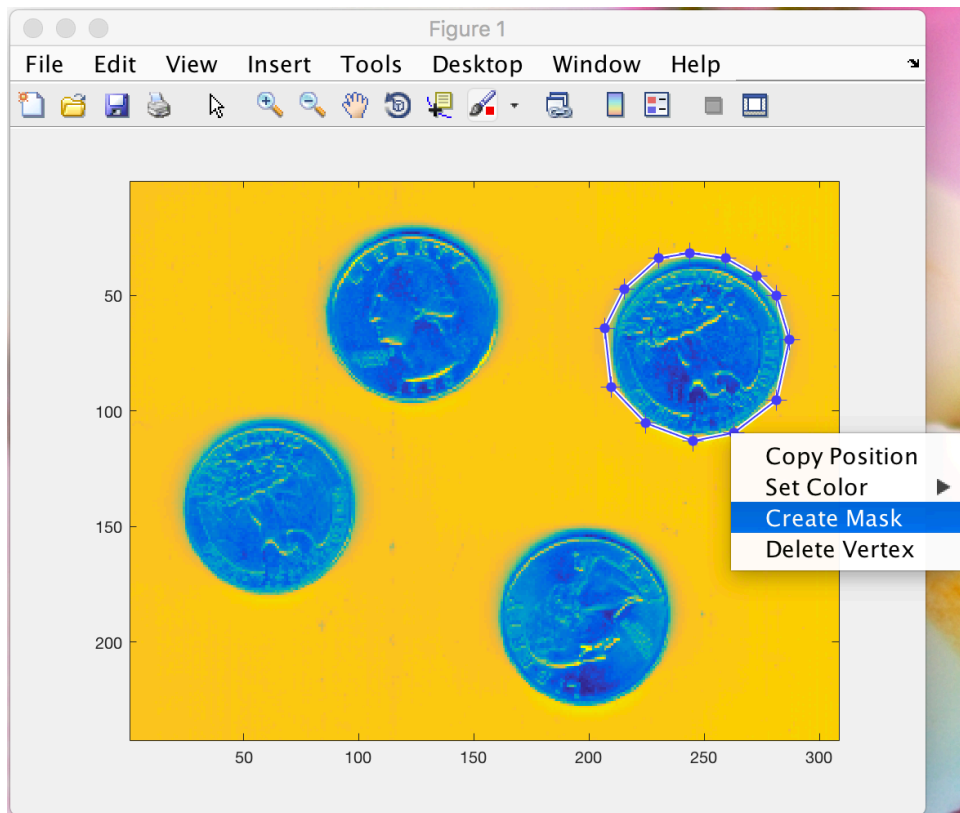


918x1632 logical

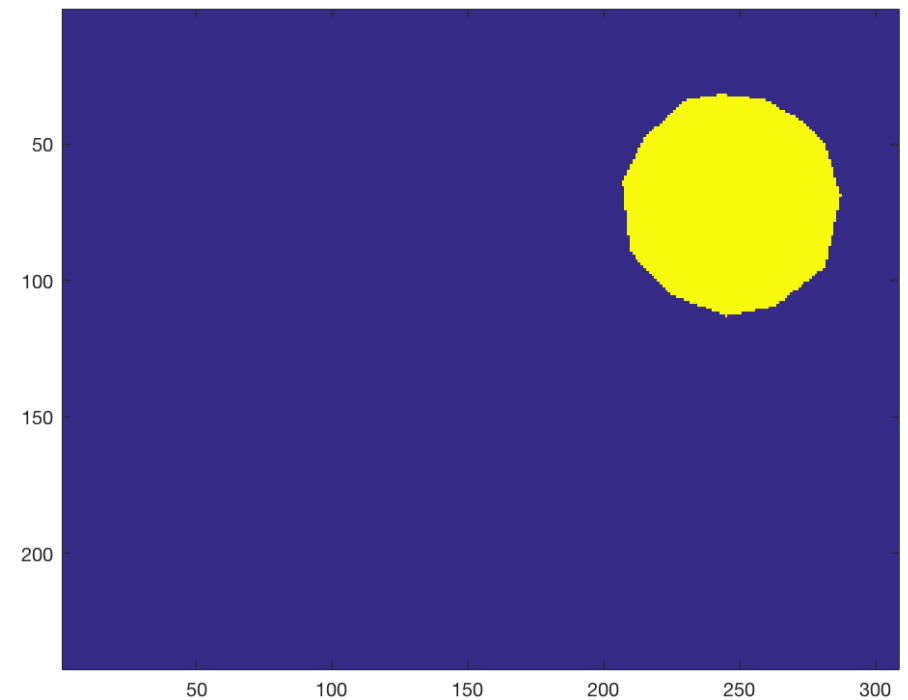
Polygon to Region Mask

```
A = imread('eight.tif'); imagesc(A)
BW = roipoly;
```

A

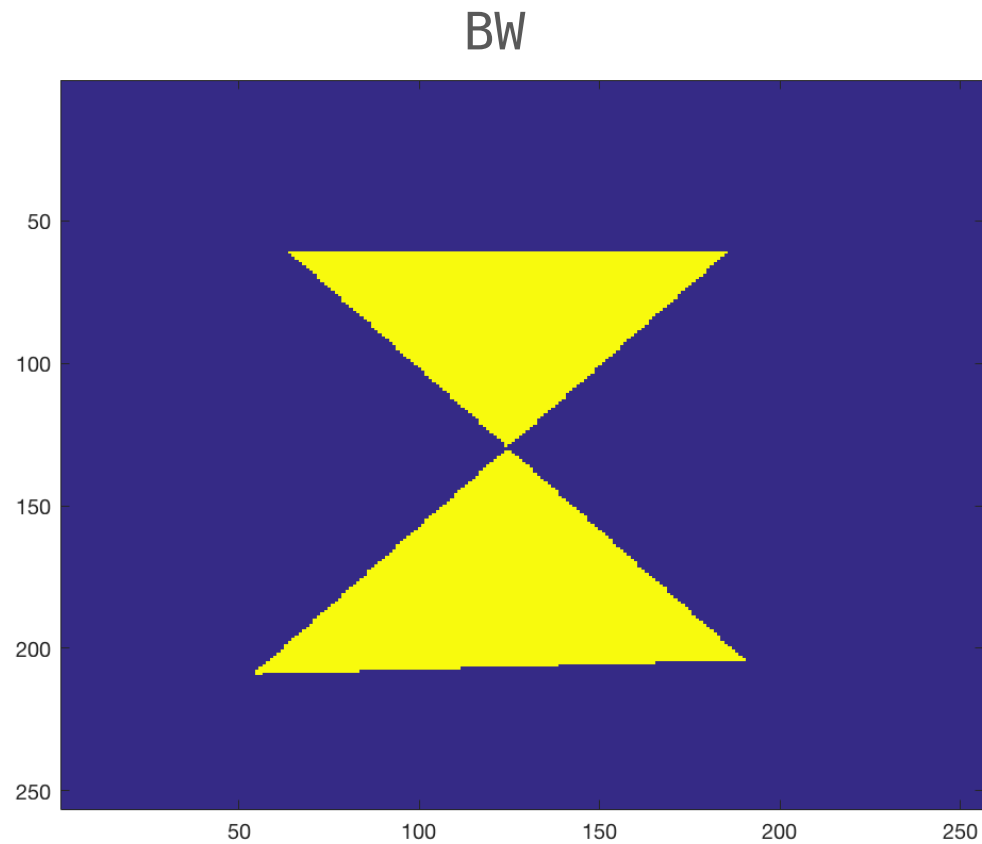


BW



Polygon to Region Mask

```
x = [63 186 54 190 63];  
y = [60 60 209 204 60];  
BW = poly2mask(x,y,256,256);
```



Labeling Segments

BW ☐

☒ 8x8 logical

	1	2	3	4	5	6	7	8
1	1	1	1	0	0	0	0	0
2	1	1	1	0	1	1	0	0
3	1	1	1	0	1	1	0	0
4	1	1	1	0	0	0	0	0
5	1	1	1	0	0	0	1	0
6	1	1	1	0	0	0	1	0
7	1	1	1	0	0	1	1	0
8	1	1	1	0	0	0	0	0

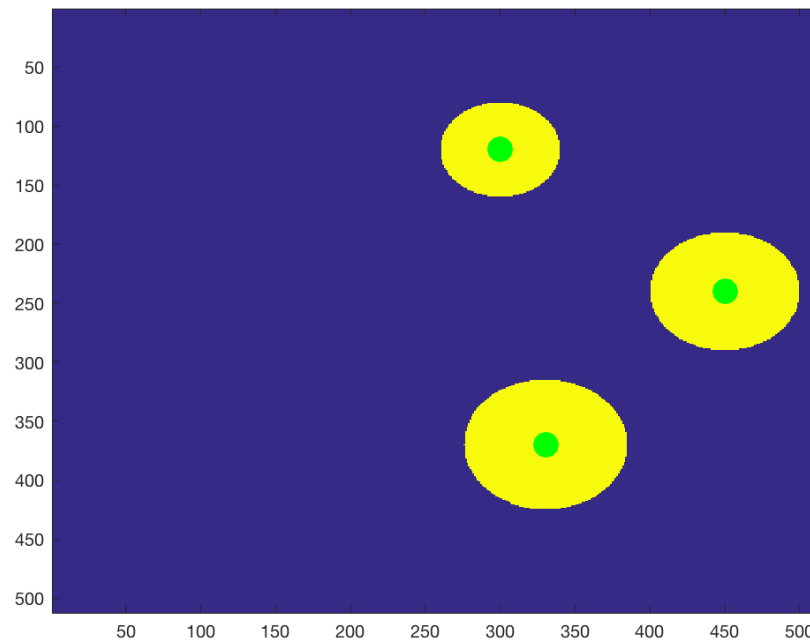
`L = bwlabel(BW);`

L ☐

☒ 8x8 double

	1	2	3	4	5	6	7	8
1	1	1	1	0	0	0	0	0
2	1	1	1	0	2	2	0	0
3	1	1	1	0	2	2	0	0
4	1	1	1	0	0	0	0	0
5	1	1	1	0	0	0	3	0
6	1	1	1	0	0	0	3	0
7	1	1	1	0	0	3	3	0
8	1	1	1	0	0	0	0	0


```
A = imread('circlesBrightDark.png');  
BW = A<60;  
s = regionprops(BW, 'centroid');  
imagesc(BW)  
hold on  
for x = 1:length(s)  
    plot(s(x).Centroid(1),s(x).Centroid(2),'g.','MarkerSize',50)  
end
```



- Get the rice number and their average length in the image SP03-02.png.

Hint

1. `regionprops(BW, 'MajorAxisLength')` gets the length in pixels of the major axis of an ellipse.

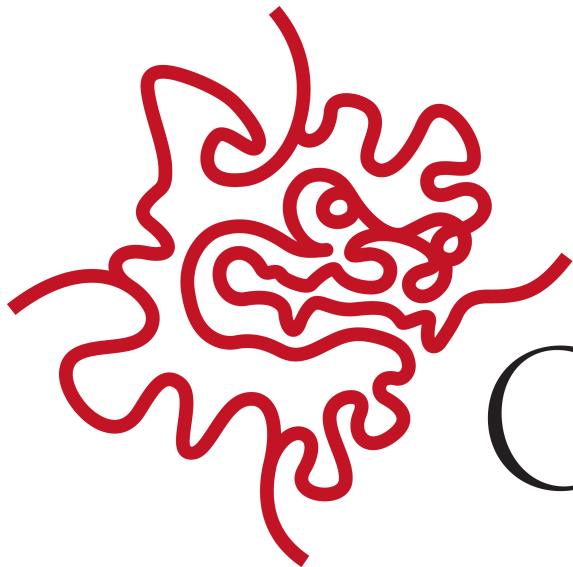
Further consideration

1. Some rice are not entirely shown in the picture, which bias the average length.



SKILLPILLS

Filtering



OIST

2-D Filters

```
A = imread('peppers.png');
```

```
filter1 = ones(30,30)/(30^2) ;  
FilteredA1 = imfilter(A,filter1);
```

```
filter2 = fspecial('motion',50,45);  
FilteredA2 = imfilter(A,filter2);
```

```
filter3 = fspecial('gaussian',30,3);  
FilteredA3 = imfilter(A,filter3);  
FilteredA4 = imgaussfilt(A,3);
```



A



FilteredA1



FilteredA2



FilteredA3
FilteredA4

```
A = imread('peppers.png');
```

```
filter2 = fspecial('motion',50,45);  
FilteredA2 = imfilter(A,filter2);
```

```
DeblurredFilteredA2 = deconvlucy(FilteredA2,filter2);
```



A



FilteredA2



DeblurredFilteredA2

- Improve the images in SP03-03.tif and SP03-04.tif files.

Hint

1. SP03-Exercise03.tif needs to be de-noised.
2. SP03-Exercise04.tif needs to be de-blurred.
3. Both need a Gaussian filter.

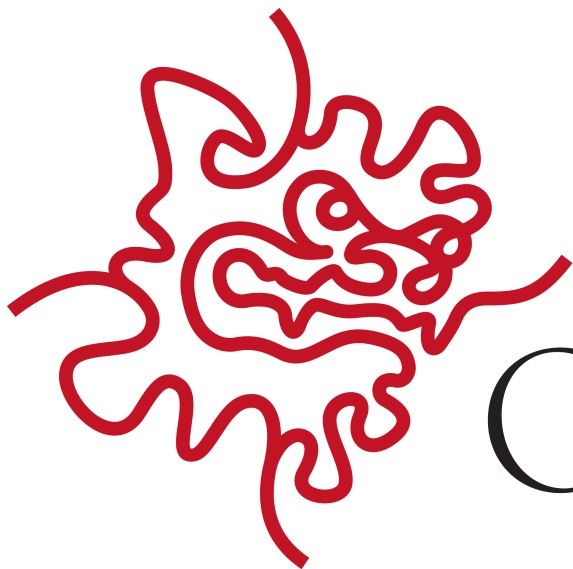
Further consideration

1. Approaches to find out a filter to recover an image.

The background of the slide is filled with numerous red and white capsules, some whole and some broken, scattered across the entire area.

SKILLPILLS

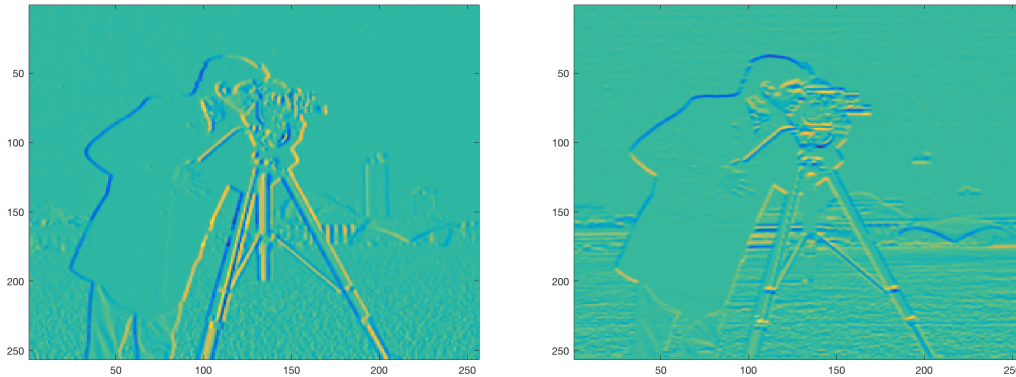
Feature Detection & Feature Matching



OIST

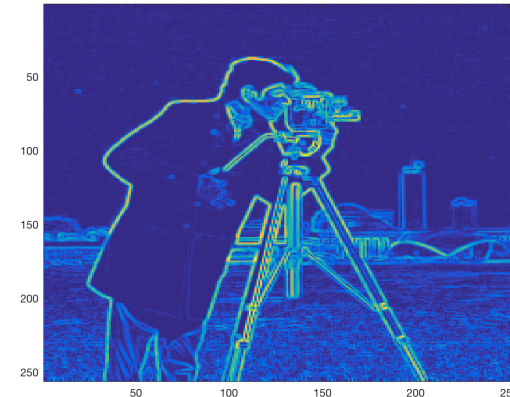
Gradients, Edges, and Corners

```
[Gx,Gy] = imgradientxy(A);
```



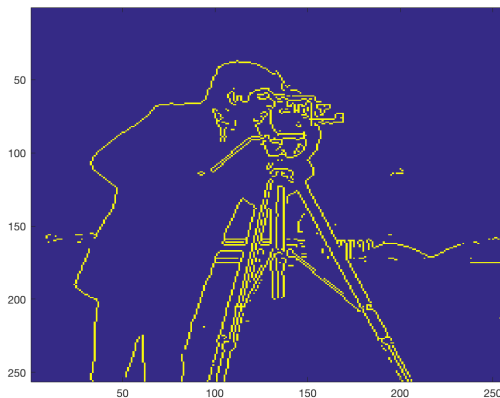
256x256 double

```
Gm = imgradient(A);
```



256x256 double

```
Edge = edge(A);
```



256x256 logical

```
corners = detectFASTFeatures(A);
```

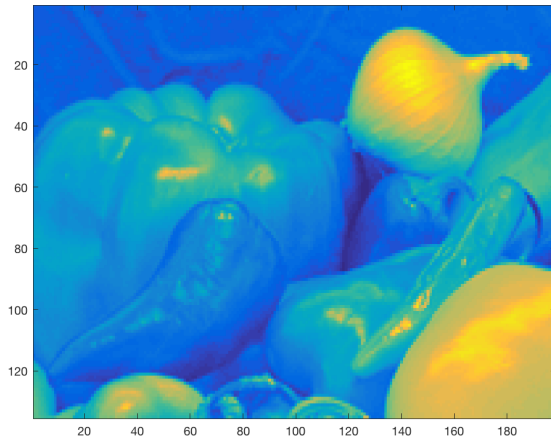


237x1 cornerPoints

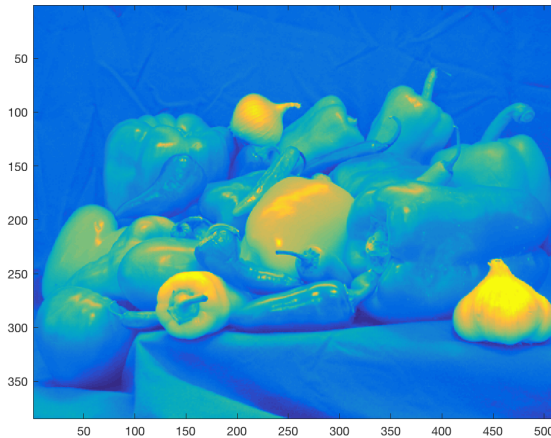
2-D Cross-Correlation

```
template = rgb2gray(imread('onion.png'));  
A = rgb2gray(imread('peppers.png'));  
xcorr = normxcorr2(template,A);
```

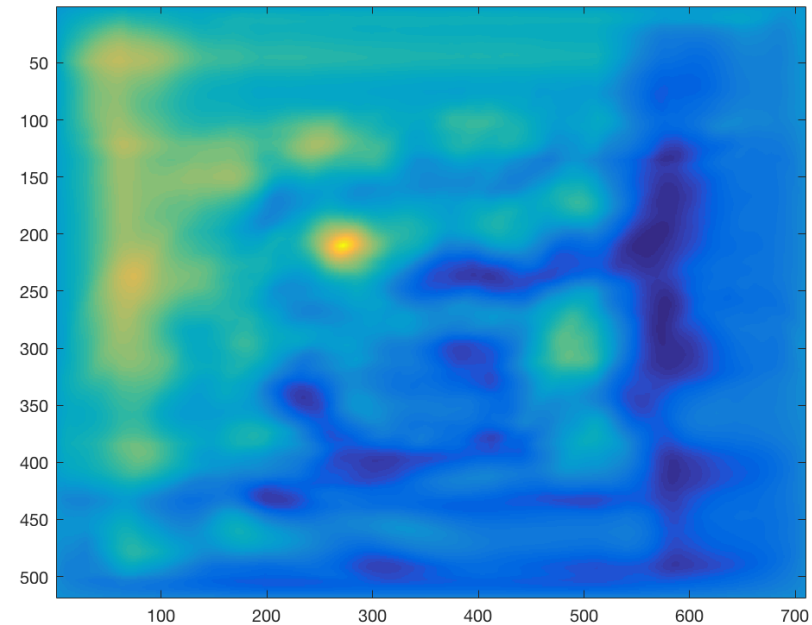
template
(135x198 uint8)



A
(384x512 uint8)



xcorr
(518x709 double)



```
[~,ind]=max(xcorr(:));  
[y,x] = ind2sub(size(xcorr),ind);  
y = y - ceil(size(xcorr,1)/2);  
x = x - ceil(size(xcorr,2)/2);  
 $(\Delta x, \Delta y) = (-83, -49)$ 
```

Matching Point Pairs

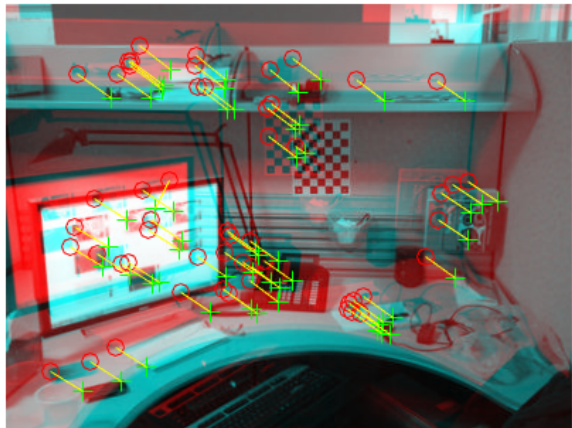
```
A1 = rgb2gray(imread('viprectification_deskLeft.png'));  
A2 = rgb2gray(imread('viprectification_deskRight.png'));
```

```
corners1 = detectFASTFeatures(A1);  
corners2 = detectFASTFeatures(A2);
```

```
[features1,validPoints1] = extractFeatures(A1,corners1);  
[features2,validPoints2] = extractFeatures(A2,corners2);
```

```
indexPairs = matchFeatures(features1,features2);
```

```
showMatchedFeatures(A1,A2,...
```

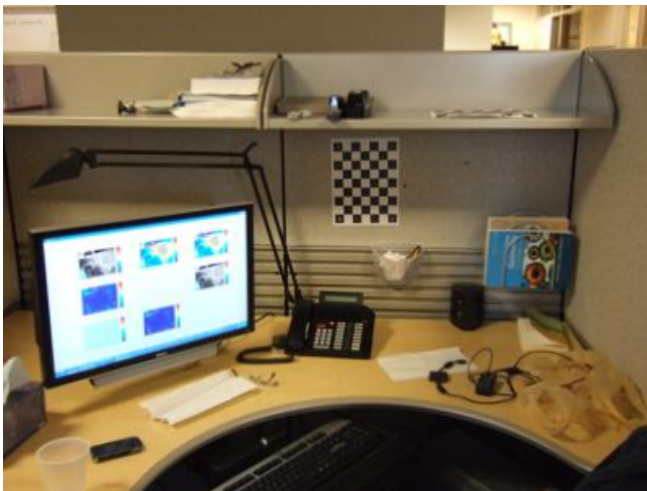


```
validPoints1(indexPairs(:,1),:),...  
validPoints2(indexPairs(:,2),:));
```

```
tform = estimateGeometricTransform(...  
    validPoints1(indexPairs(:,1),:),...  
    validPoints2(indexPairs(:,2),:),...  
    'similarity');
```


Exercise 4

- By referring SP03-05.tif, recover the distorted image SP03-06.tif.

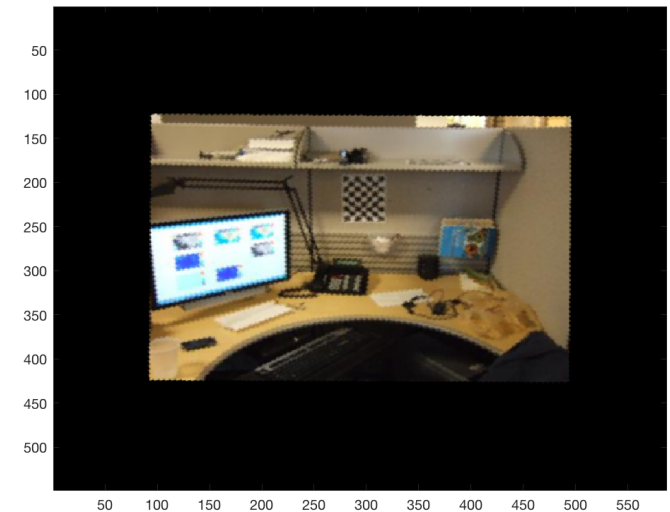


SP03-05.tif



SP03-06.tif

(Shifted + Resized + Rotated)



Recovered image