Lecture 12

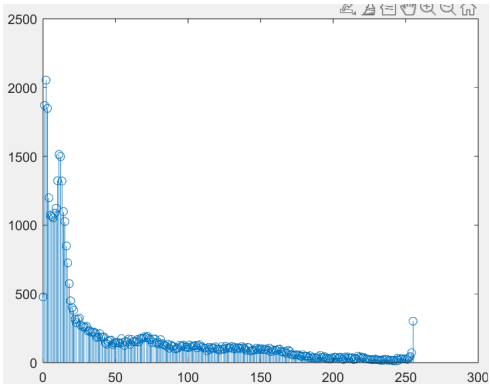
(cont'd) As a comparison, *imadjust* function in Matlab

```
>> imtool(I)
>> doc imadjust
```

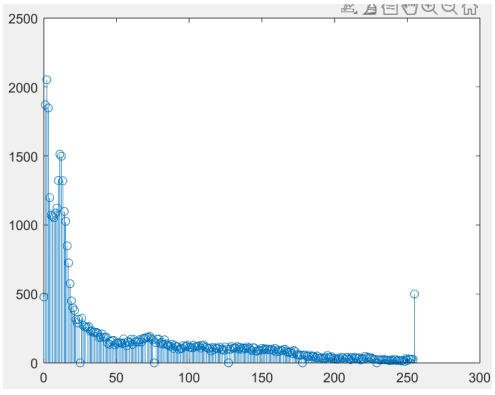
imadjust

Adjust image intensity values or colormap

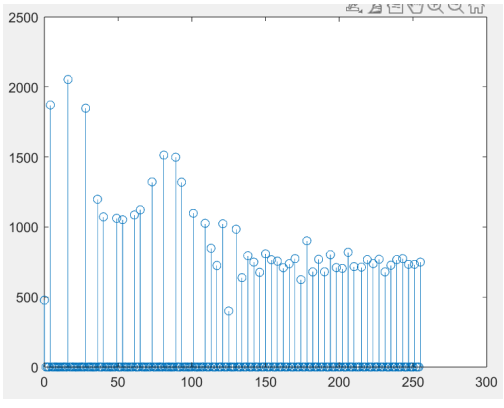
```
>> I = imread('tire.tif');
>> imtool(I)
>> [counter, bin] = imhist(I);
>> figure; stem(bin, counter);
```



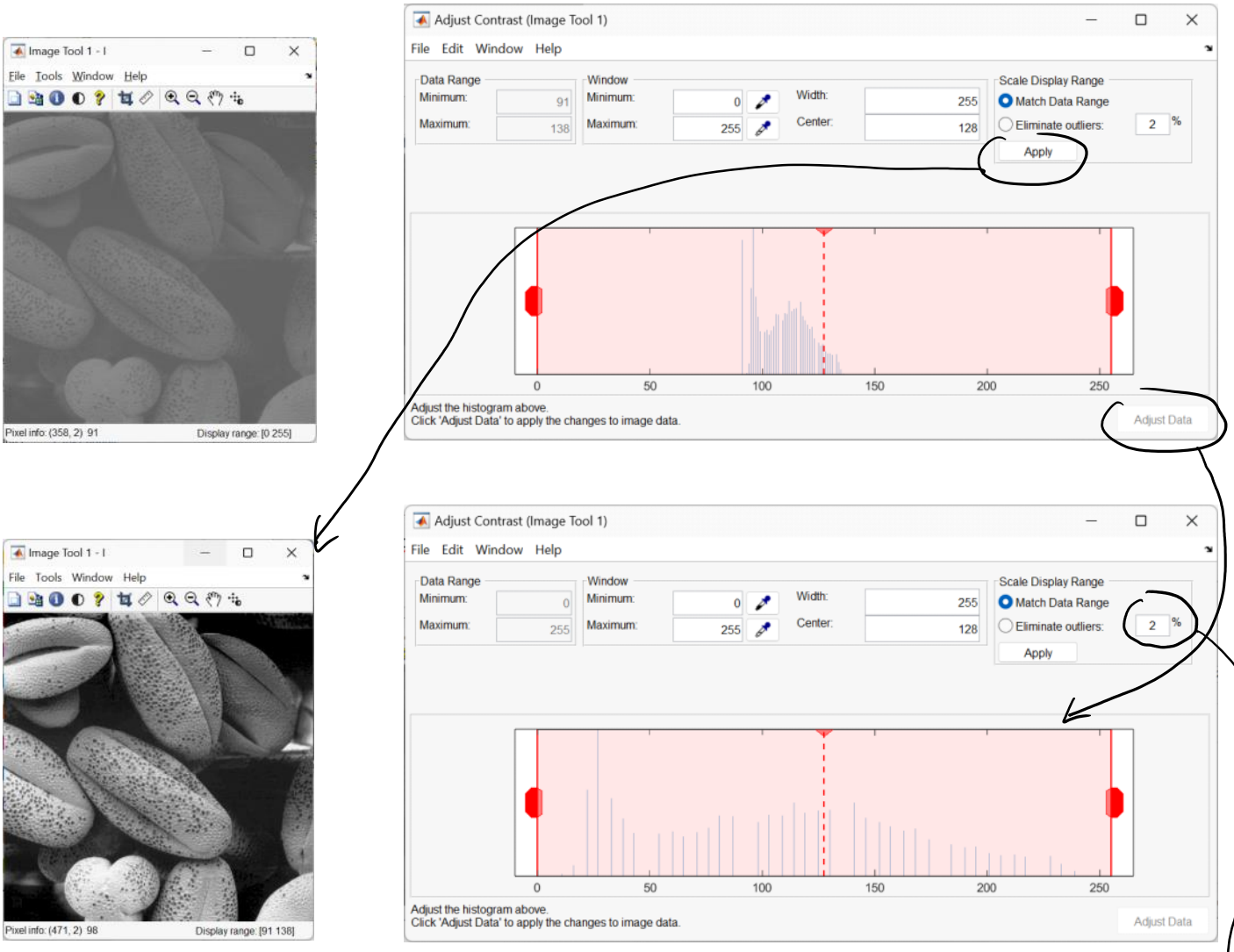

```
>> J = imadjust(I);
>> figure; imshow(J)
>> [counter, bin] = imhist(J);
>> figure; stem(bin, counter);
```




```
>> K = histeq(I);
>> figure; imshow(K)
>> [counter, bin] = imhist(K);
>> figure; stem(bin, counter);
```



```
>> I = imread('Fig0320(2)(2nd_from_top).tif');
>> imtool(I)
```



stretchlim

Find limits to contrast stretch image

lowhigh = stretchlim(I) computes the lower and upper limits that can be used for contrast stretching grayscale or RGB image I. The limits are returned in lowhigh. By default, the limits specify the bottom 1% and the top 1% of all pixel values.

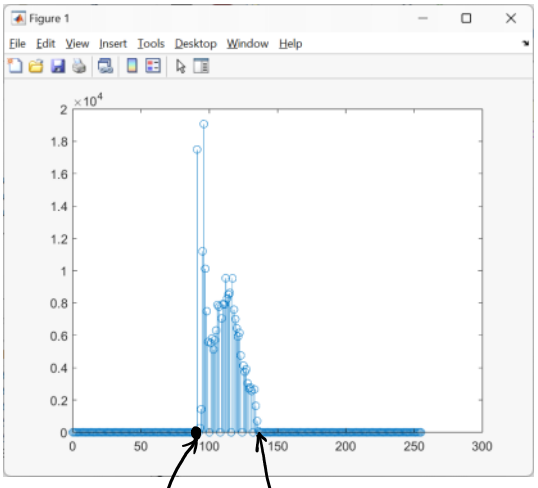
```
>> lowhigh = stretchlim(I)

lowhigh =
    0.3569
    0.5216
```

```
>> 91/255

ans =

    0.3569
```

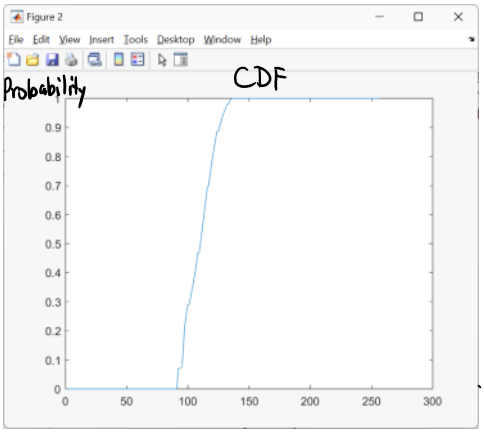
```
>> 133/255

ans =

    0.5216
```

J = imadjust(I) maps the intensity values in grayscale image I to new values in J. By default, imadjust saturates the bottom 1% and the top 1% of all pixel values. The function linearly maps pixel values between the saturation limits to values between 0 and 1. This operation increases the contrast of the output image J.

This syntax is equivalent to imadjust(I,stretchlim(I)).




```
>> [counter, bin] = imhist(I);
>> figure; stem(bin, counter);
>> cdf = cumsum((counter)/sum(counter));
>> figure; plot(cdf)
```




```
>> min(I(:))
ans =
  uint8
  91
```

```
>> max(I(:))
ans =
  uint8
  138
```

```
>> low = find(cdf > 0.01, 1, 'first')
low =
    92
```

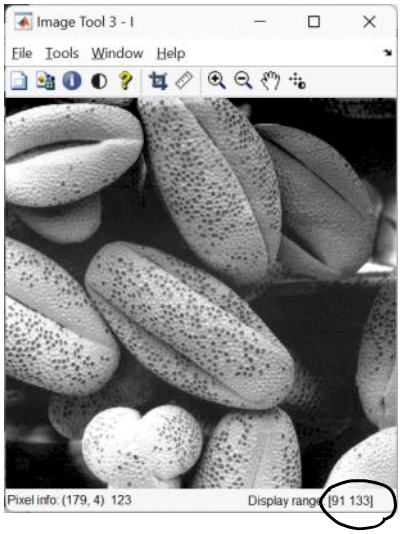
(bin index)

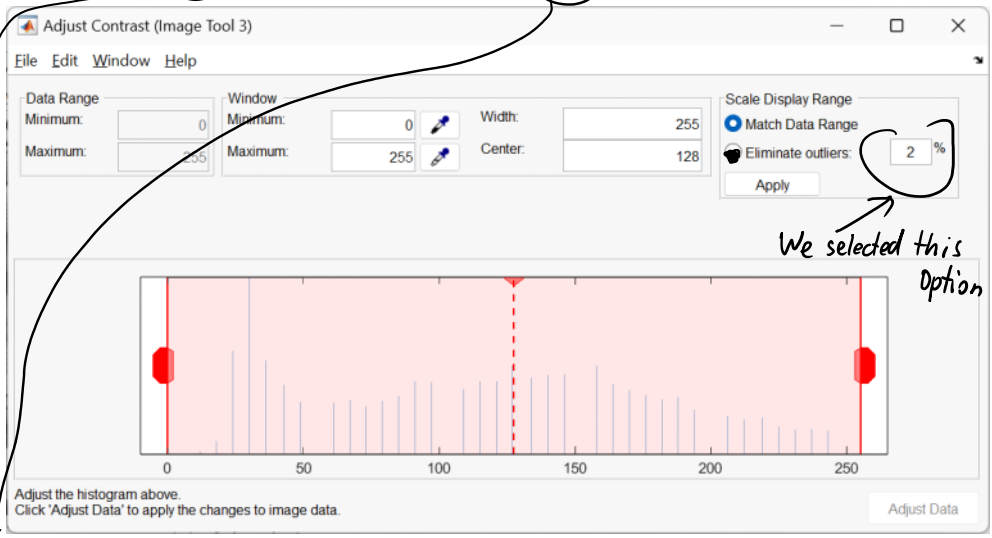
pixel value: 92 - 1 = 91

```
>> high = find(cdf >= 0.99, 1, 'first')
high =
   134
```

(bin index)

134 - 1 = 133





[91 133]

```
>> bin(1:10)'
ans = 0  1  2  3  4  5  6  7  8  9
```

pixel values

index: 1 2 3