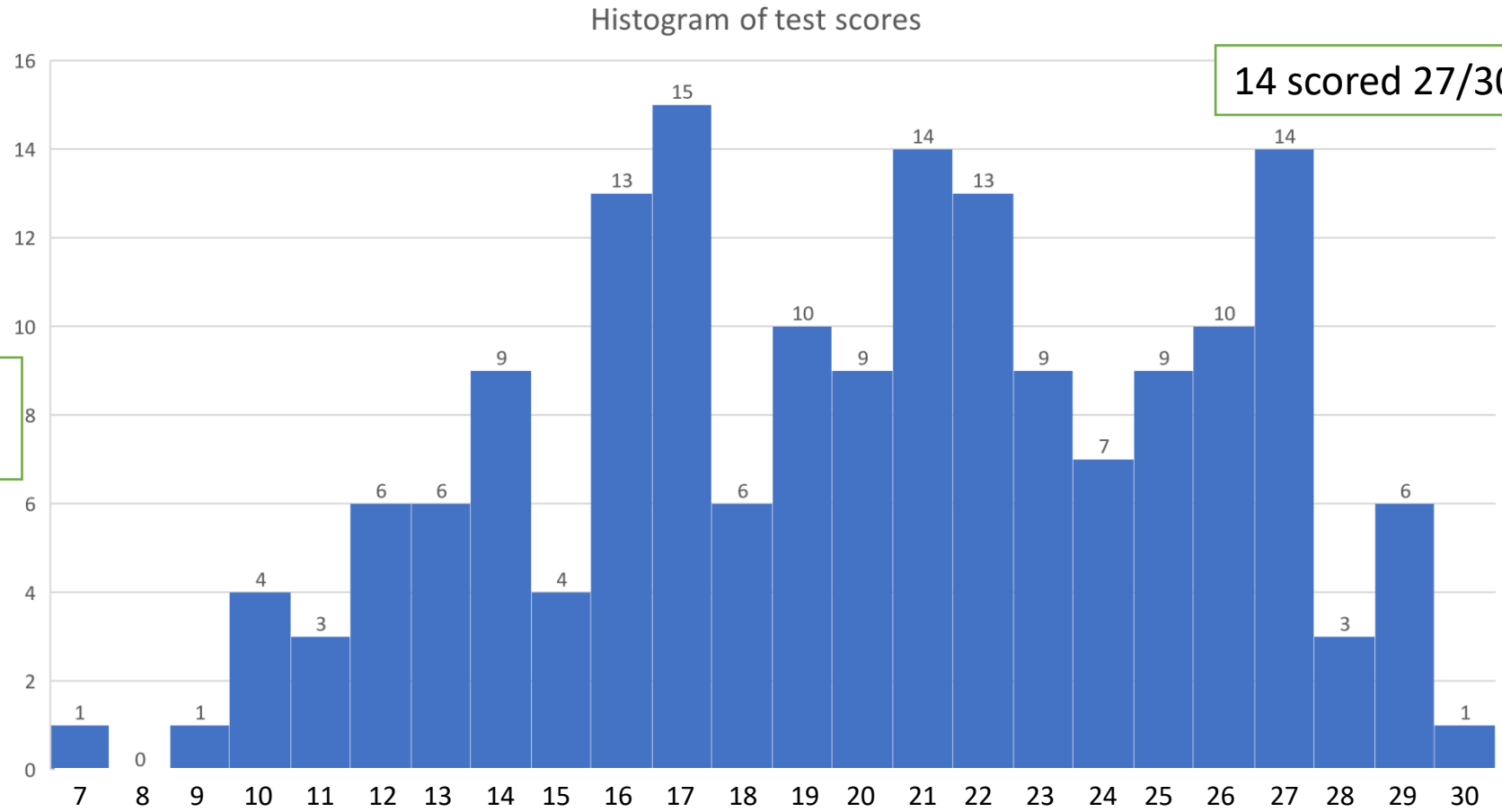
Histograms

13.00
18.00
19.
18.
14.00
16.00
17.00
16.00
27.00
26.00
16.00
16.00
10.00
13.00
12.00
19.00
20.00
17.00
17.00
24.00
17.00
16.00
12.00
12.00
21.00
14.00
17.00
21.00
20.00
18.00
16.00
13.00
27.00

Extract from test scores

Count of data value falling into bin



14 scored 27/30

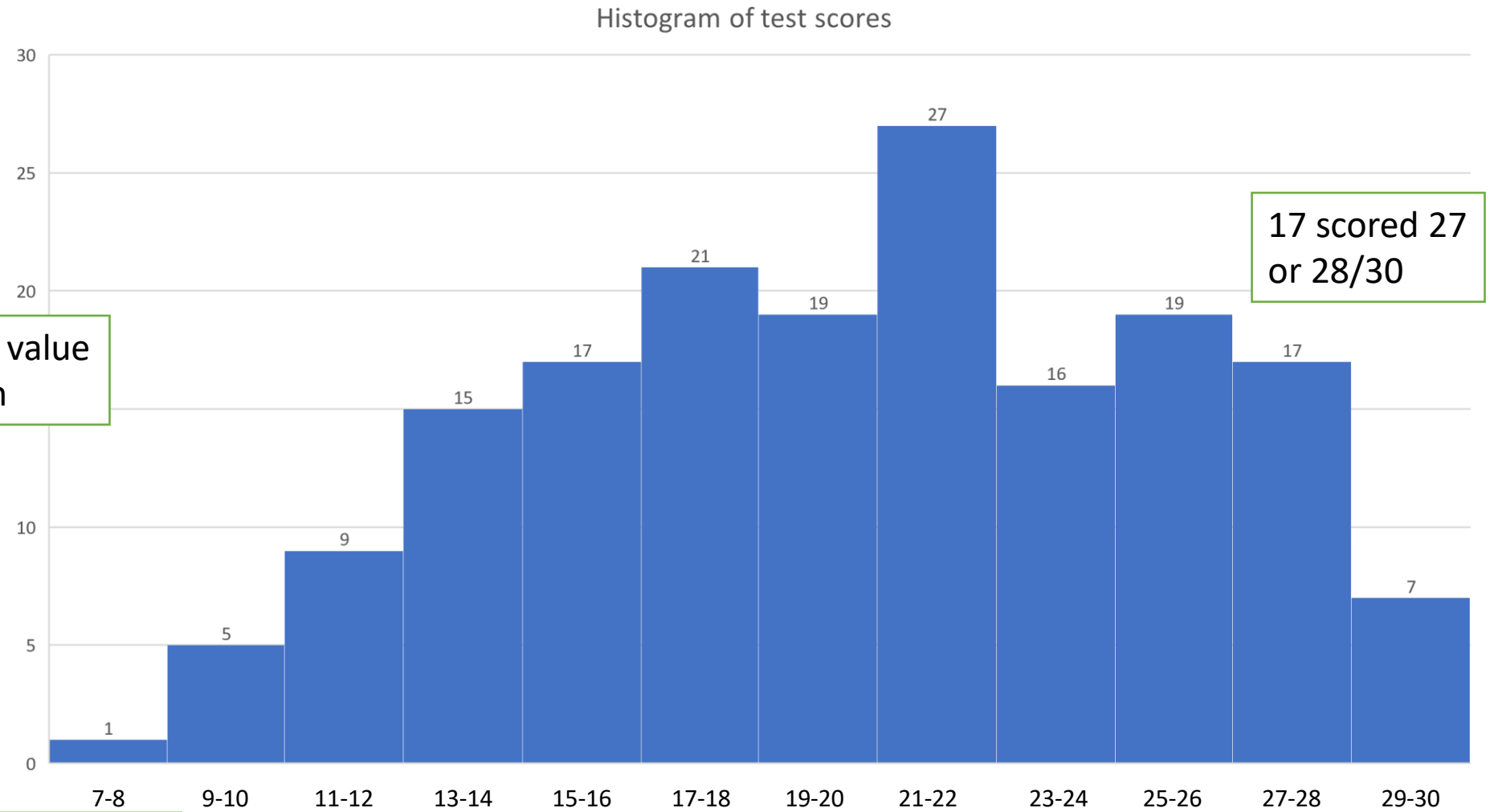
Bin size is 1

Histograms

13.00
18.00
19.
18.
14.00
16.00
17.00
16.00
27.00
26.00
16.00
16.00
10.00
13.00
12.00
19.00
20.00
17.00
17.00
24.00
17.00
16.00
12.00
12.00
21.00
14.00
17.00
21.00
20.00
18.00
16.00
13.00
27.00

Extract from test scores

Count of data value falling into bin

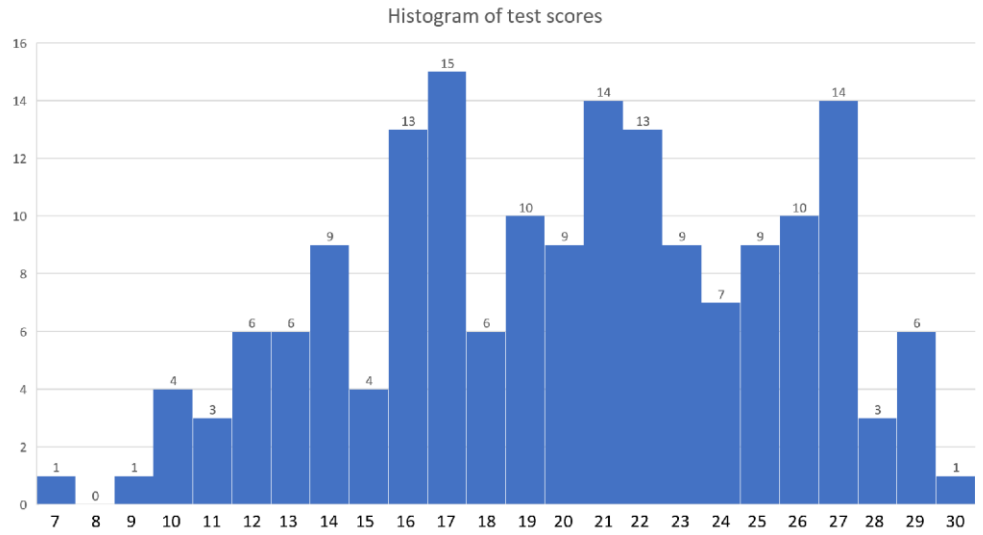


17 scored 27 or 28/30

Bin size is 2

Coding Histograms

- Step 1: Determine: The number of bins; The bin boundaries; How to map data value to bin.
- Step 2: Allocate and initialise to zero the frequency table that will count the number of data elements in each bin.
- Step 3: Loop through data elements, increasing the bin count for the bin of that data element.
- Step 4: Render the histogram.



Coding Image Histograms

- Step 1: Bin size usually 1. The pixel R, G, B or I (0-255) value will be the bin reference.
- Step 2: The R, G, B and I frequency tables are allocated to each be size 256.
- Step 3: Iterate through each pixel. Obtain R, G and B values. Compute I value (e.g., $I = (R + G + B) \div 3$). Add 1 to each respective frequency table at position R, G, B and I.
- Step 4: Use the R, G, B and I arrays to render the histograms.

- Note: It is a common student mistake to create the intensity histogram by adding the 256 entries from the three R, B and B histograms together and dividing by 3. **This is incorrect.**

Image Histogram

- An image histogram is a graph representing number of pixels at each intensity or colour level. This image is dark. The histogram is bunched on low values

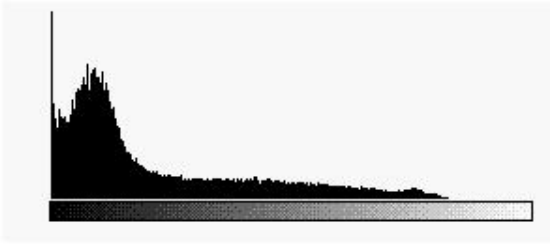
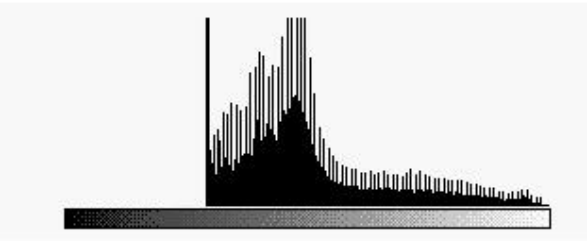


Image histogram

- We can make non-uniformly brighter using gamma correction, or uniformly brighter by adding constant intensity to each pixel



Histogram Calculation (example for red)

```
int[][] histogram;
histogram=new int[256][3];
//also need to declare/assign value, colour, h, w, etc.

for (j=0; j<h; j++)
   for (i=0; i<w; i++) {
       colour=pixelReader.getColor(i,j);
       value=colour.getRed();
       histogram[value][0]++; //[0] for red, [1] green
    }
//output the histogram to screen
for (i=0; i<256; i++)
    System.out.println(i+" r="+histogram[i][2]+" g="+histogram[i][1]+
                           " b="+histogram[i][0]);
```

Histograms

Typical Output

233 r=790 g=491 b=566

234 r=783 g=672 b=1214

235 r=704 g=796 b=1308

236 r=577 g=970 b=227

237 r=542 g=982 b=1278

238 r=614 g=1045 b=190

239 r=614 g=1093 b=1134

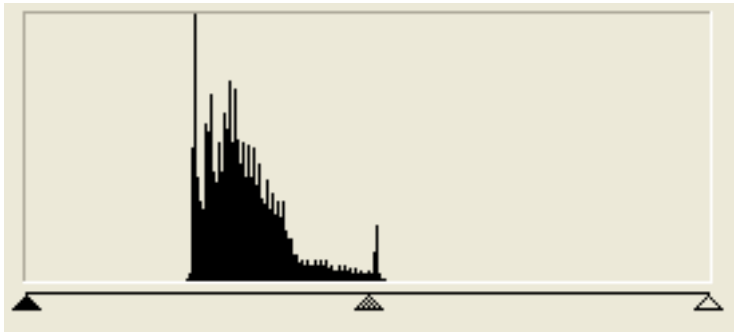
240 r=572 g=1065 b=240

241 r=700 g=844 b=828

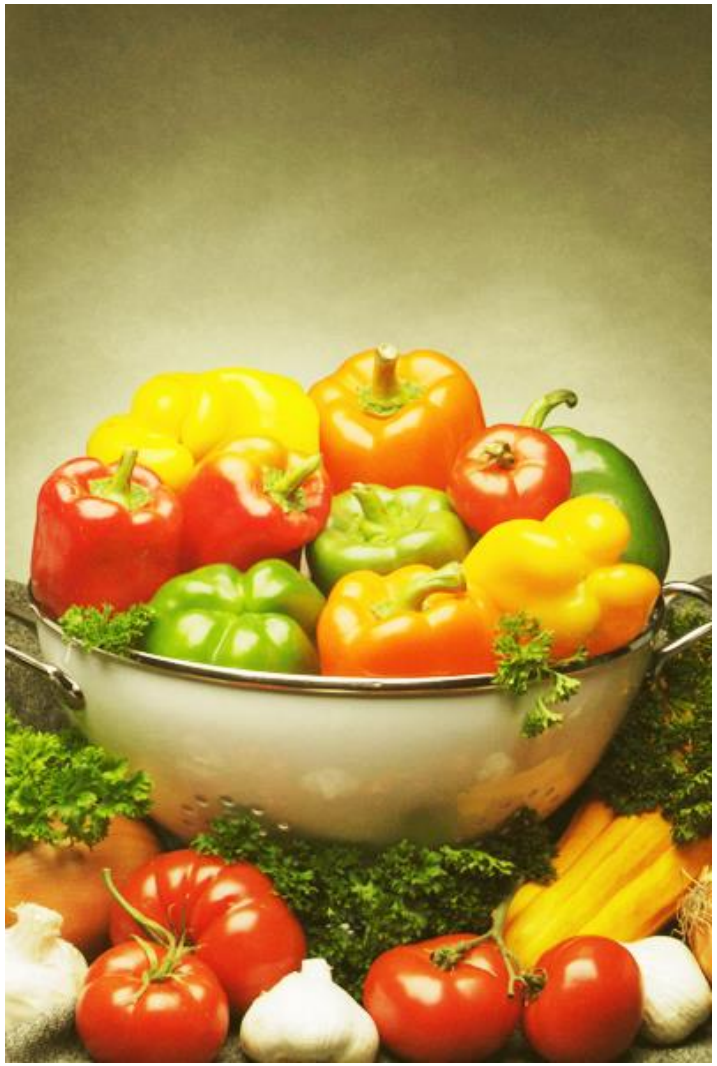
- e.g. 790 pixels have a red value of 233, 491 pixels have a green value of 233, and 566 pixels have a blue value of 233

Histograms

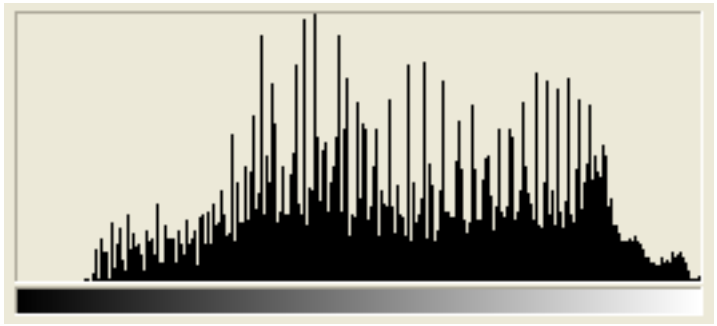
- Reduced contrast in Photoshop
- Displayed histogram (combined RGB into grey level image)



Histograms

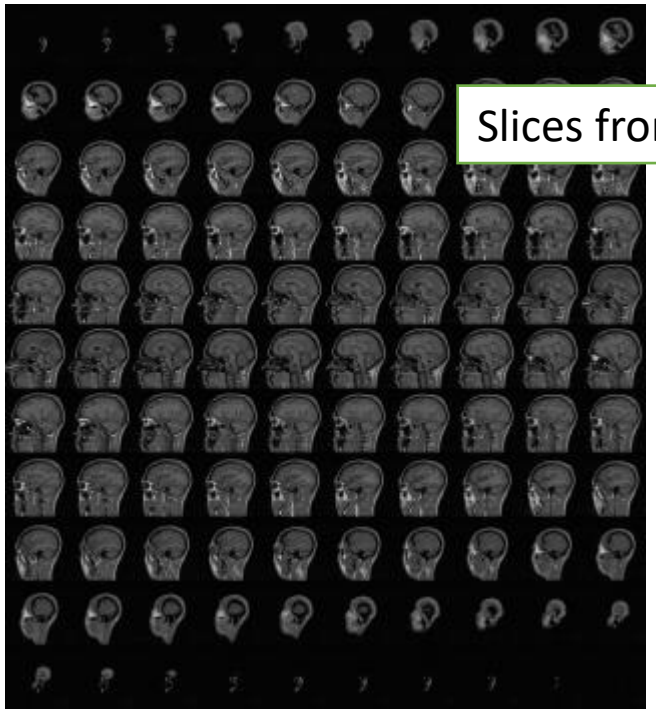


- Used Histogram Equalization
- Note how the histogram is more spread out

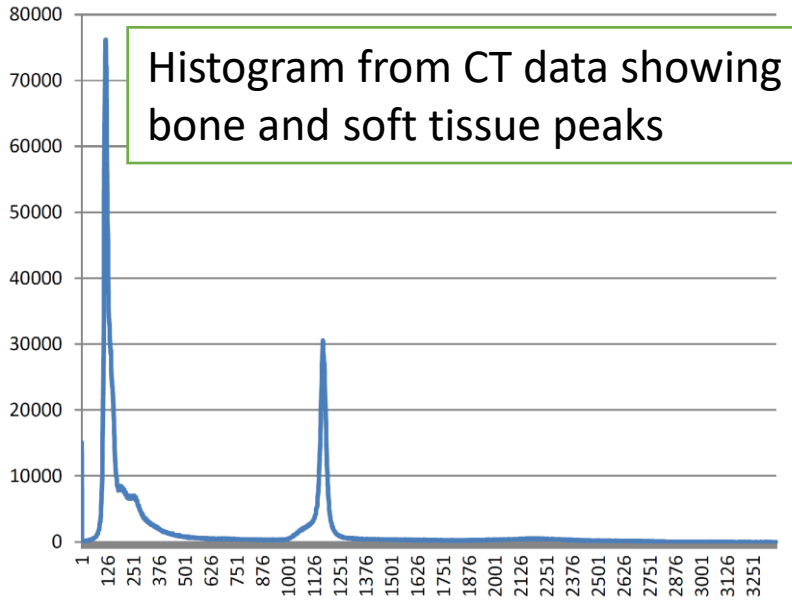


Histograms of Volumetric Data

CT Scanner

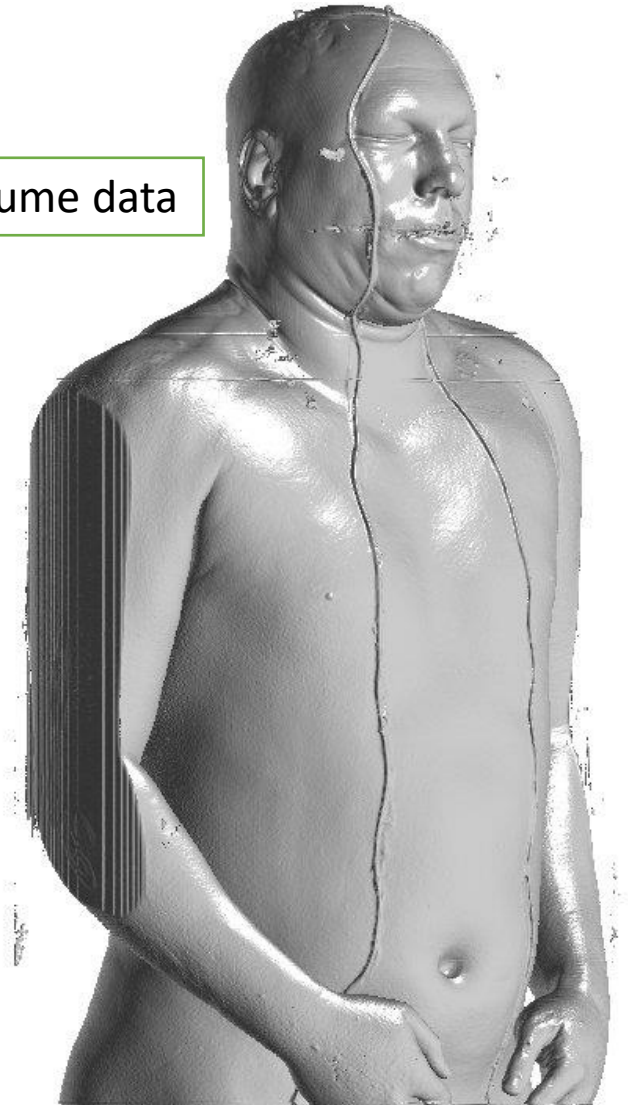


Slices from MRI data



Histogram from CT data showing bone and soft tissue peaks

3D render from volume data



Coding Continued

- Assuming bin size is 1, find max and min of input data.
- Number of bins for histogram is max-min+1.
- Initialise each entry in the histogram to zero count.
- For each input data value, *v*, increase the count of bin *v-min* by 1.
- Render the resulting histogram.

Histogram Equalization

- Contrast can be improved by spreading distribution of intensity evenly over the pixels
- Total number of pixels in image is height*width (h*w)
- Number of grey levels is g_levels
- Ideal histogram is flat
- Ideal number of pixels at each level:
- p_num=(h*w)/g_levels

Histogram Equalization

- Mapping F is defined as:
- *If t(i) is the actual number of pixels at all old grey levels up to and including grey level i, then*

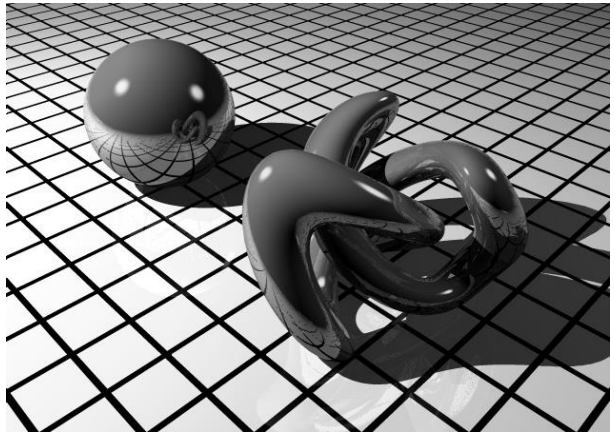
F(i)=max{0,round((g_levels*t(i))/(h*w))-1}

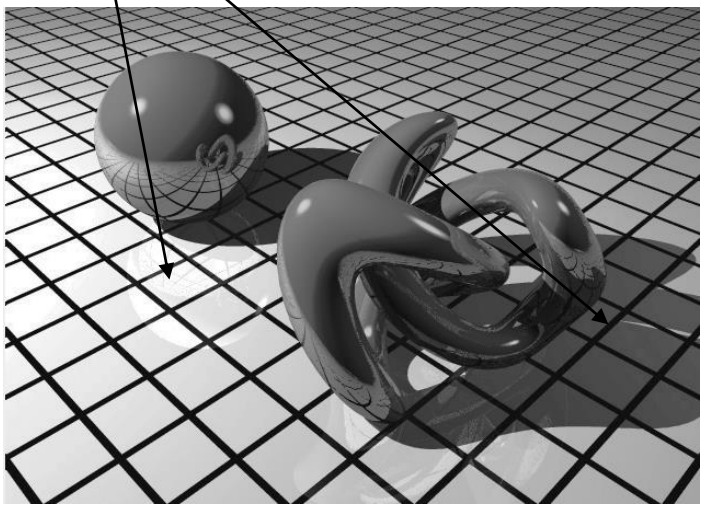
t(i) is known as the cumulative distribution function

Histogram Equalization

110 115
111 116
112 116
113 117
114 117
115 118
116 119
117 119
118 120
119 121
120 121
121 122
122 122
123 123
124 124
125 124

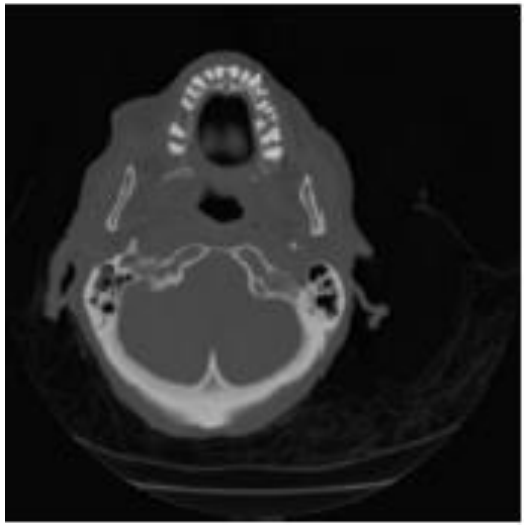
- e.g. old grey pixels of value 112 map to new grey pixel of value 116
- see contrast in shadows and reflections





Histogram Equalization

- Lowering the dynamic range in images
- e.g. 12 bit CT scanner – display on 8 bits?
- naïve – remove lowest 4 bits



- Better – use histogram equalization with g_levels=max_intensity-min_intensity=3365 (for this CT head)

Colour Histogram Equalization

- Convert RGB to HSV
- Build histogram on V (brightness)
- Equalize V
- Convert HSV back to RGB
- Display

Further reading

- https://en.wikipedia.org/wiki/Histogram
- https://en.wikipedia.org/wiki/Image_histogram
- https://en.wikipedia.org/wiki/Histogram_equalization