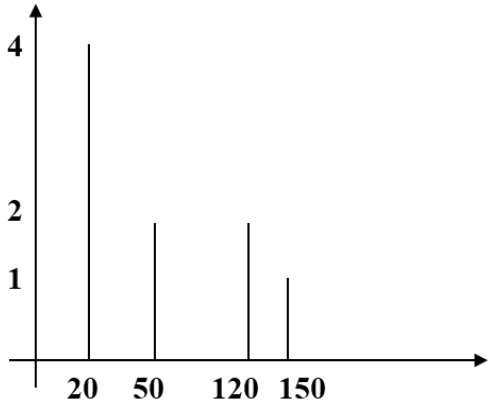
Histogram Equalization

Sung Soo Hwang

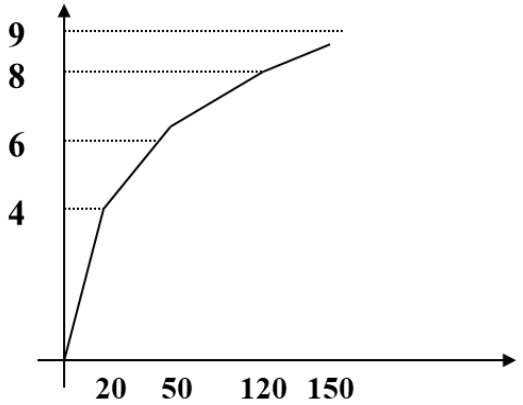
Introduction

- Definition of a histogram
  - Histogram of an image with intensity levels in the range [0, L-1]
    - $h(r_k) = n_k$
    - r_k: k^{th} intensity value n_k:the number of pixels with intensity r_k
    - The number of bin is L in this case

20	20	50
20	20	50
120	120	150



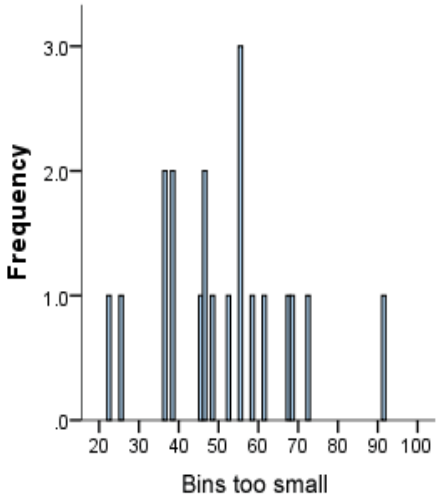
Histogram

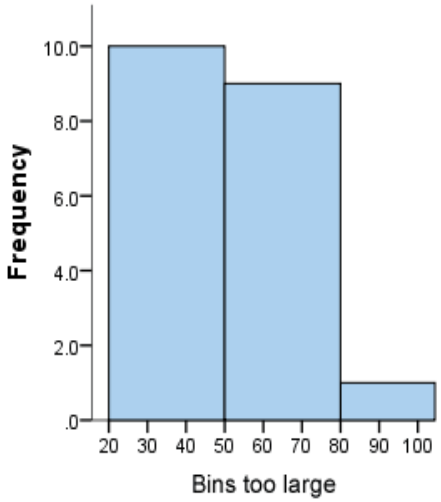


Cumulative histogram

Introduction

- Histogram normalization
  - By dividing each of its components by the total number of pixels
  - It can be considered as a probability function
- Histograms are the basis for numerous spatial domain processing techniques
  - Setting the proper number of bins is important

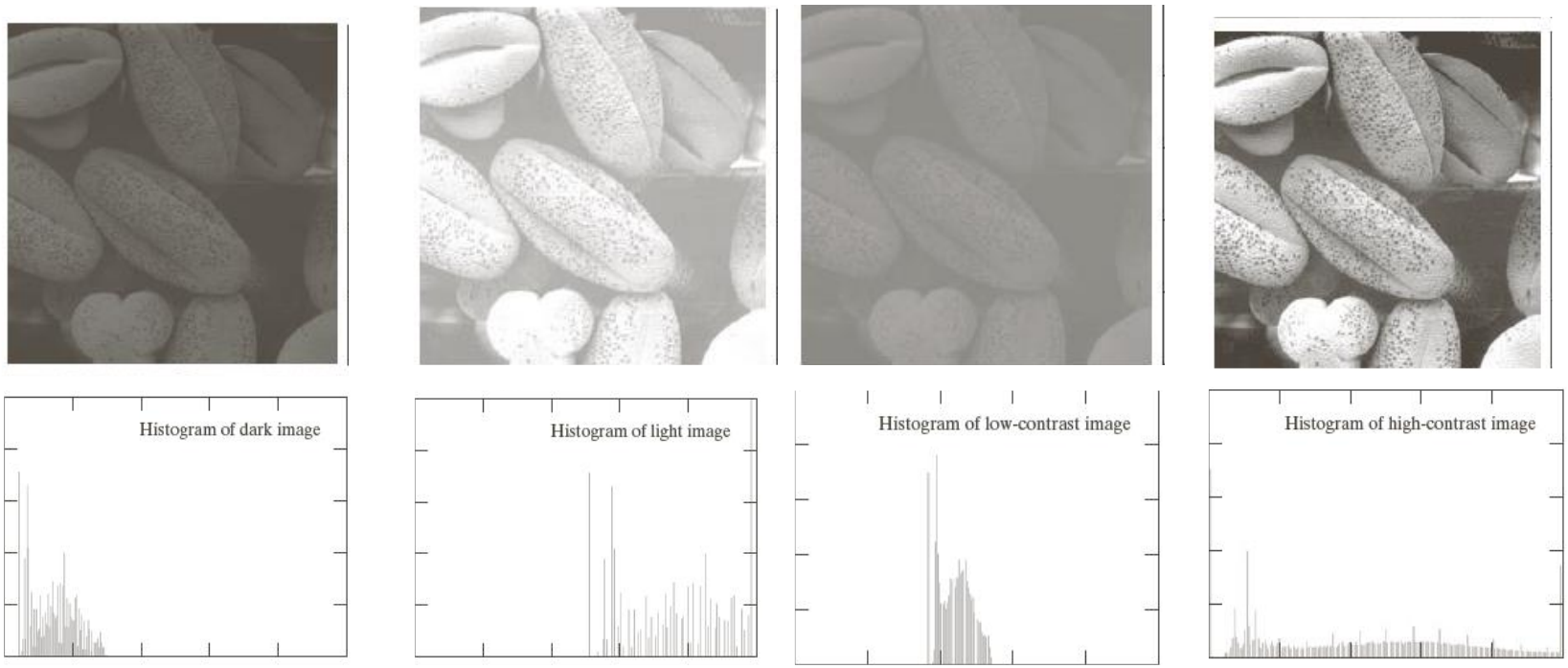




In both case, it is difficult to estimate the characteristics of data

Histogram equalization

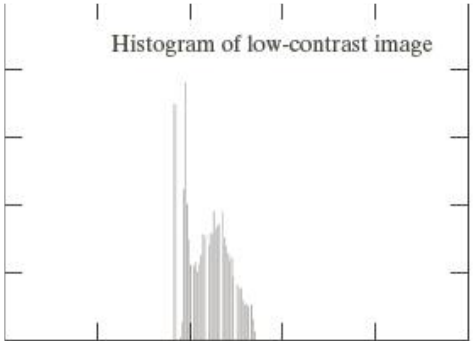
- Histogram equalization
  - A method which adjust contrast of an image
  - Contrast: The difference in brightness or color that makes an object distinguishable



Histogram equalization

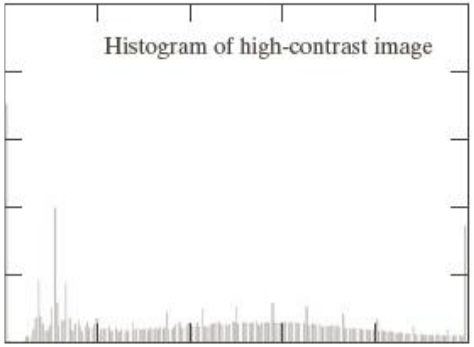
- Histogram equalization

1. Histogram computation



2. Find mapping function which distributes pixel values uniformly

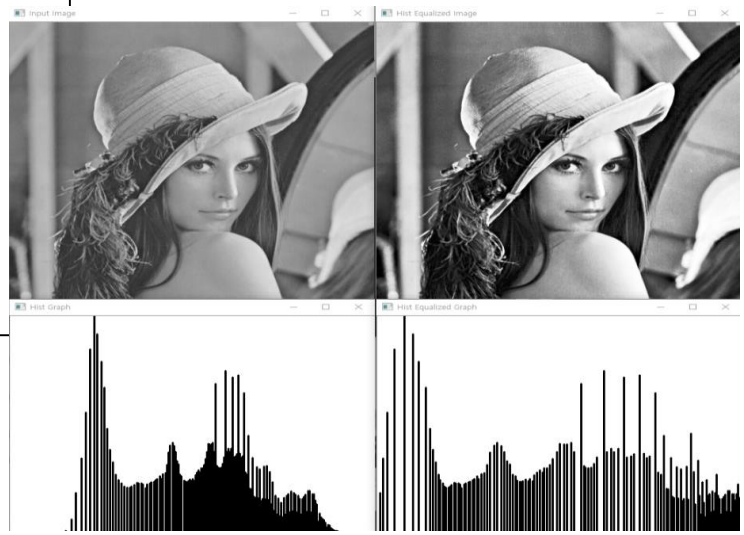
3. Apply the mapping function to an input image



Histogram equalization

- Example code

```
int main() {
	Mat image;
	Mat hist_equalized_image;
	Mat hist_graph;
	Mat hist_equalized_graph;

	image = imread("lena.png", 0);
	if (!image.data) exit(1); //Check image

	equalizeHist(image, hist_equalized_image); //histogram equlization

	hist_graph = drawHistogram(image);
	hist_equalized_graph = drawHistogram(hist_equalized_image);

	imshow("Input Image", image);
	imshow("Hist Equalized Image", hist_equalized_image);
	imshow("Hist Graph", hist_graph);
	imshow("Hist Equalized Graph", hist_equalized_graph);

	waitKey(0);
	return 0;
}
```



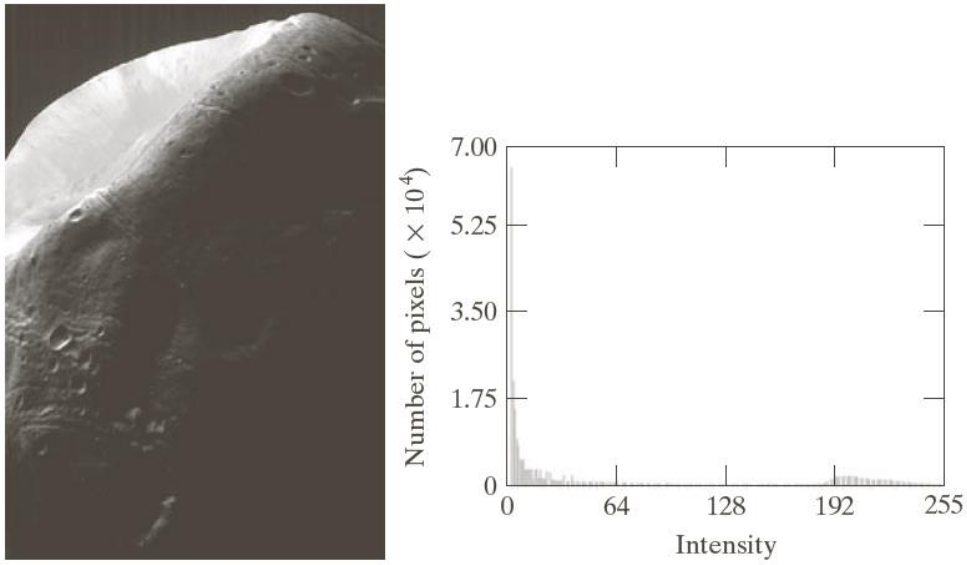
Histogram equalization

- Example code

```
Mat drawHistogram(Mat src){
	Mat hist, histImage;

	// establish the number of bins
	int i, hist_w, hist_h, bin_w, histSize;
	float range[] = { 0, 256 };
	const float* histRange = { range };

	hist_w = 512;
	hist_h = 400;
	histSize = 256;
	bin_w = cvRound((double)hist_w / histSize);

	// compute the histograms
	// &src: input image, 1: #of src image, 0: #of channels numerated from 0 ~ channels()-1, Mat(): optional mask
	// hist: output histogram, 1: histogram dimension, &histSize: array of histogram size,  &histRange: array of histogram's boundaries
	calcHist(&src, 1, 0, Mat(), hist, 1, &histSize, &histRange);

	//draw the histogram
	histImage = Mat(hist_h, hist_w, CV_8UC3, Scalar(255, 255, 255));

	// normalize the result to [0, histImage.rows]
	// hist: input Mat, hist: output Mat, 0: lower range boundary of range normalization, histImage.rows: upper range boundary
	// NORM_MINMAX: normalization type, -1: when negative, the ouput array has the same type as src, Mat(): optional mask
	normalize(hist, hist, 0, histImage.rows, NORM_MINMAX, -1, Mat());

	for (i = 1; i < histSize; i++) {
	    rectangle(histImage, Point(bin_w*i, hist_h), Point(bin_w*i, hist_h - cvRound(hist.at<float>(i))), Scalar(0, 0, 0), 2, 8, 0);
	}
	return histImage;
}
```

Histogram equalization

- Does histogram equalization provide an enhanced image?
  - Not always, especially when a certain range of data is dominant



Washed out appearance