IIT Bombay

Computer Programming

Dr. Deepak B Phatak
Dr. Supratik Chakraborty
Department of Computer Science and Engineering
IIT Bombay

Session: Associative Arrays for Histogram Equalization

Quick Recap

- We looked at the digital representation of images, and use of matrices to represent images
- We discussed
  - Histogram and Cumulative Distribution Function
  - Histogram equalization technique to improve contrast

Overview

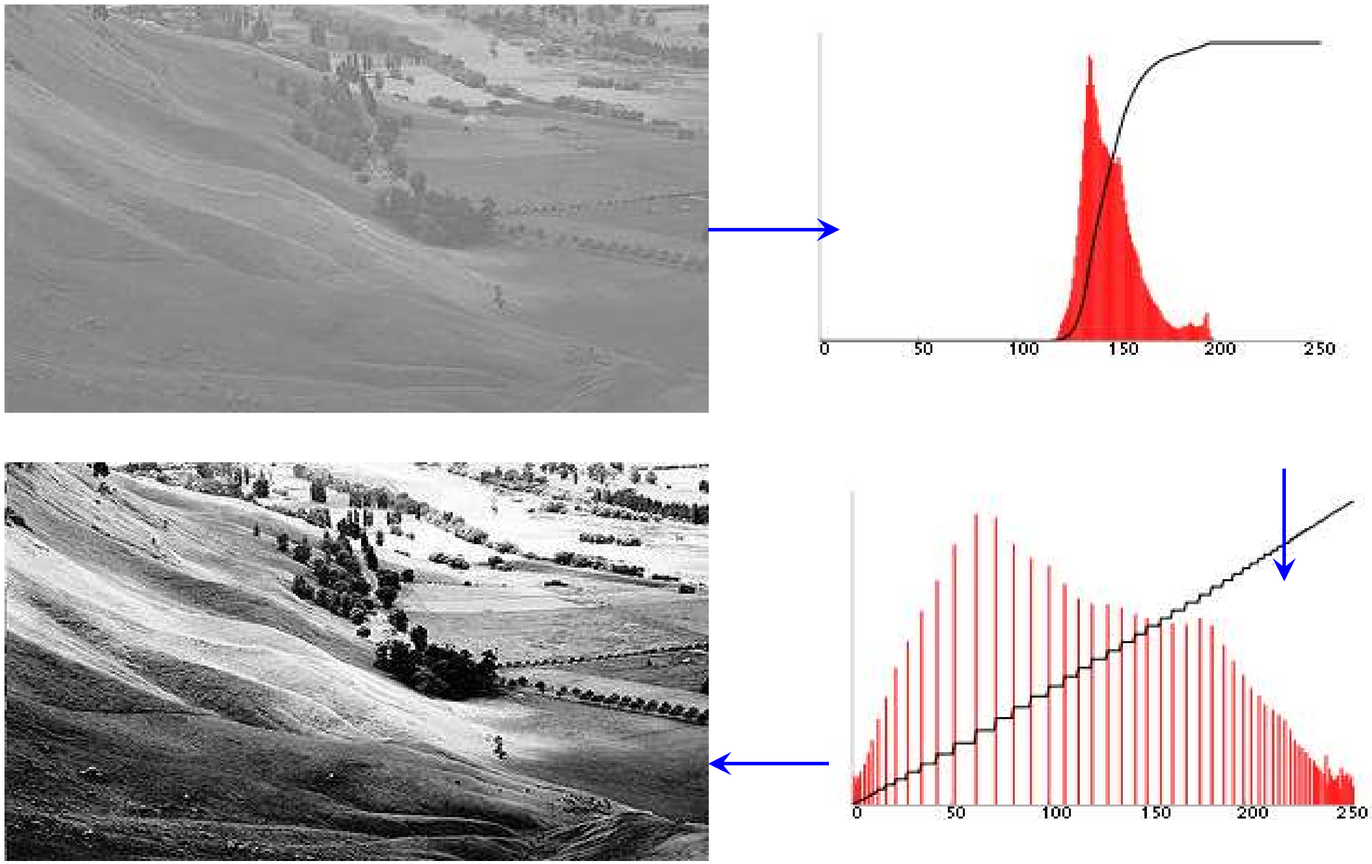
IIT Bombay

- In the context of histogram calculations, we will look at different computations to be performed
- We will discuss the important concept of "Associative Array" for use in our program, to be written for contrast enhancement

[Note: The histogram equalization technique described here, and the digital images used, are directly based on a wikipedia article:

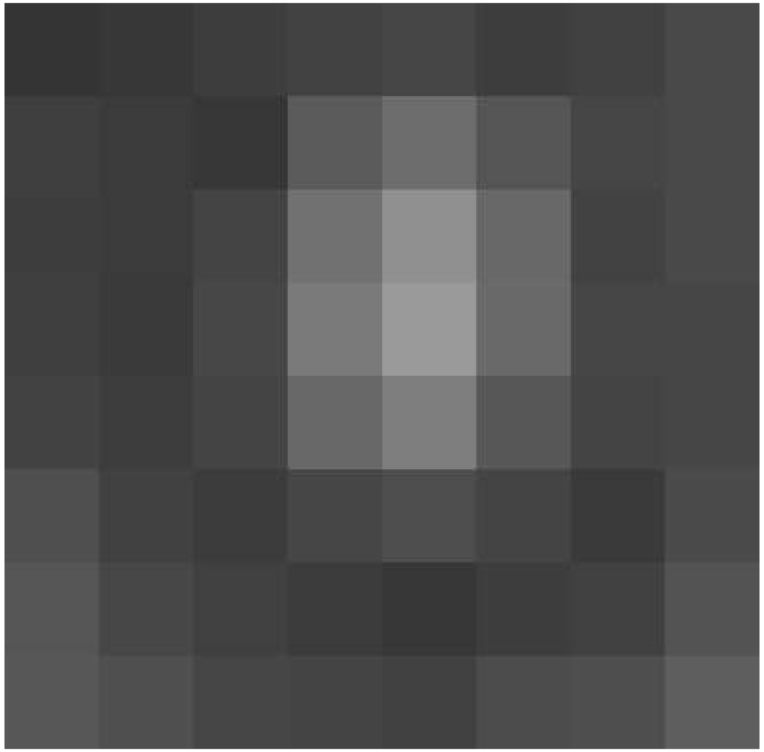
http://en.wikipedia.org/wiki/Histogram_equalization]

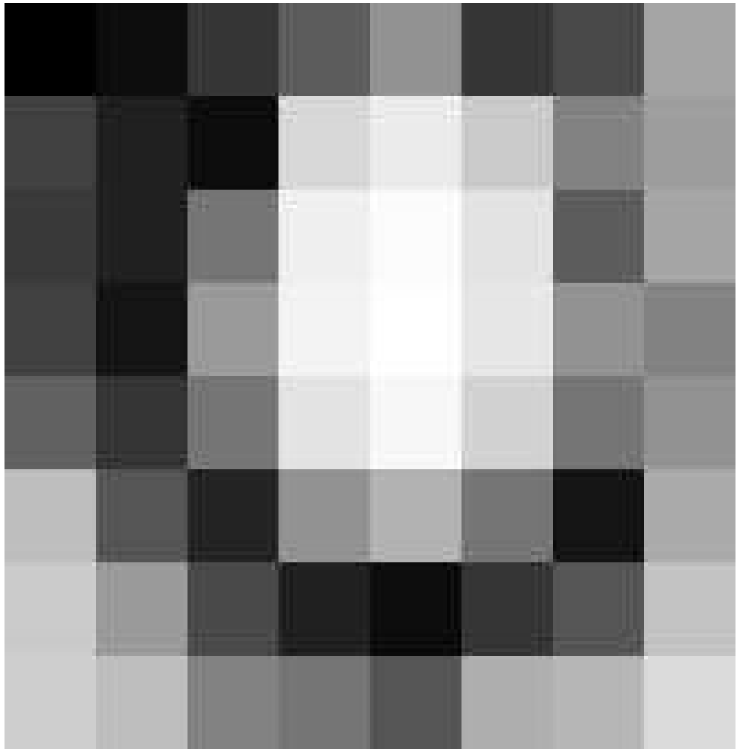
Original picture, equalization, modified picture



Original and contrast-enhanced pictures

IIT Bombay





Pixel values for the image

IIT Bombay

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

Histogram values (shown for non-zero pixels)

IIT Bombay

Val	n	Val	n	Val	n	Val	n	Val	n
52	1	64	2	72	1	85	2	113	1
55	3	65	3	73	2	87	1	122	1
58	2	66	2	75	1	88	1	126	1
59	3	67	1	76	1	90	1	144	1
60	1	68	5	77	1	94	1	154	1
61	4	69	3	78	1	104	2		
62	1	70	4	79	2	106	1		
63	2	71	2	83	1	109	1		

Histogram Calculations

IIT Bombay

- Histogram for a particular pixel value, is the total number of pixels in the image which have the same value
- Best way to represent the histogram, is to use a one dimensional array of 256 elements
  - a pixel can only be between 0 to 255
- We need to accumulate the count of each pixel value found in the image, in the corresponding element of the histogram array

Histogram Calculations

IIT Bombay

Index	Initial Value
0	0
1	0
.	.
68	0
69	0
.	.
255	0

Pixel values in the image

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

Histogram Calculations

IIT Bombay

Index	Initial Value
0	0
1	0
.	.
68	$\boxed{5}$
69	0
.	.
255	0

Pixel values in the image

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & \boxed{68} & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & \boxed{68} & 104 & 126 & 88 & \boxed{68} & 70 \\ 79 & 65 & 60 & 70 & 77 & \boxed{68} & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & \boxed{68} & 65 & 76 & 78 & 94 \end{bmatrix}$$

Histogram Calculations

IIT Bombay

Index	Initial Value
0	0
1	0
.	.
68	5
69	$\boxed{3}$
.	.
255	0

Pixel values in the image

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & \boxed{69} & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & \boxed{69} \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & \boxed{69} & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

Cost of Computing Histogram Values

IIT Bombay

- If this logic is used to compute histogram entries, what is the number of operations the computer has to perform?
- Assume array name IM[8][8] for image, and H[256] for histogram
  - For each pixel value p, we scan all 8x8 (= 64) pixel values in IM, and compare each with p.
  - We assign the total count of those pixel values which match, to the appropriate element of H
- We repeat this for each of the 256 possible values (256 x 64 comparisons)
- In a large image of size 500 by 300, there will be 1,50,000 pixels!

A list of students

- A list is available containing roll numbers of students attending a MOOC, and the cities they belong to

10001	Mumbai	10009	Tokyo	10017	Paris
10002	Delhi	10010	Kathmandu	10018	Delhi
10003	Tokyo	10011	Delhi	10019	Mumbai
10004	Karachi	10012	Dhaka	10020	Dhaka
10005	Delhi	10013	Mumbai	10021	Tokyo
10006	Kathmandu	10014	Delhi	10022	Delhi
10007	Dhaka	10015	Karachi	---	---
10008	Mumabi	10016	Washington	---	---

City Statistics

- We can count the number of students from each city, and store these counts in an array (table) which looks like

City	Count
Delhi	4024
Dhaka	1729
Dubai	572
Karachi	984
Kathmandu	431
Mumbai	5102
Paris	301
Tokyo	659
Washington	850

Cost of Calculating City Statistics

IIT Bombay

- If we use our approach to do such counting
  - Take a city, say Mumbai; scan the complete list to count students who belong to that city, and put the count value against that city in our count table
  - Repeat this for each city
- If there are 50,000 students taking a MOOC, and they belong to 200 different cities
  - We need to look at each of 50,000 entries, 200 times
  - We need to perform a total of 1,00,00,000 comparisons
- A very time consuming exercise

This is not how statisticians do their counting!

Associative Array

IIT Bombay

- It will be useful, if name of a city (which is a value in our list) , can be used as a key (as an index) to directly access the associated count, which is a value in our table
- An associative array is a set of key-value pairs.
  - The organization is such that the 'key' is the index of the array
- Assume that we have such an associative array for our example of city names and corresponding student population
  - If the key "Mumbai" is given, using that as an index of the array, we should directly get 5102
  - If the key "Kathmandu" is given, we should directly get 431
- We will like to use the value (name of the city) in our list of students, directly as a key for the table, to access the count for the city

Associative Array …

IIT Bombay

- In C++, the arrays we know are indexed by ‘keys’ which are only integer numbers, in the range of 0 to N- 1 (and not stings like city names)
- For the Histogram array, the key range is 0 to 255
- Pixel values in the image array are also integer numbers, and are precisely in this range
- Thus, the pixel value itself can be used as a ‘key’ or index of the Histogram array
    - This array can be treated as an associative array

Associative Array 'H'

IIT Bombay

- Suppose the histogram count for the image array IM, is stored in array H[256], counting process will require a single scan of the image matrix
- For each pixel value 'p' we look at, we know which element of the array H needs to be updated. It is the p^{th} element!
  - "Association" between the pixel value p, and the index of array H
- The count for a pixel in the image array element IM[i][j], which has value p, can simply be incremented by the assignment:

  p = IM[i][j];   H[p] = H[p] + 1;

  or more simply   H[IM[i][j]] = H[IM[i][j]) + 1
- Multiple scans of the image array, are not required.

Summary

IIT Bombay

- In the context of contrast enhancement of digital images, we studied the computation of the histogram
- We discussed the concept of an “associative array”, and found it to be a useful way to do efficient calculations