Lect 8

"Histogram Processing"

The histogram of a digital image with L total possible intensity levels in the range [0, G] is defined as the discrete function: $h(r_k) = n_k$

where r_k is the kth intensity level in the interval [0, G] and n_k is the number of pixels in the image whose intensity level is r_k, G: [255 for images of class uint8, 65535 for images class uint16 and 1.0 for images of class double].

* Normalized histograms: can be obtained by dividing all elements of $h(r_k)$ by the total number of pixels in the image:

$$P(r_k) = \frac{h(r_k)}{n} = \frac{n_k}{n}, \quad \text{for } k = 1, 2, \ldots, L,$$

from probability: n - total number of pixels

$P(r_k)$ - estimation of the probability of occurrence of intensity level r_k. (The sum of all components of a normalized histogram is equal = 1)

Matlab function: h = imhist(f, b).

f is the input image, h - its histogram $h(r_k)$, b - number of bins used in forming the histogram (b = 255 is the default)

b - is a subdivision of the intensity scale, if we work with uint8 images and we let b = 2, then the intensity scale is subdivided into two ranges: 0 to 127 and 128 to 255.

the resulting histogram will have two values h(1) and h(2).

We obtain the normalized histogram simply by:

p = imhist(f, b) / numel(f).

examples:

```
>> imhist(f); % plot the histogram of the image f
```

histograms often are plotted using bar graphs (1), we can use: bar(horz, v, width)

v: Vector containing the points to be plotted.

horz: Vector of the same size as v that contains the increments of the horizontal scale.

width: number between 0 and 1.

②

if horz is omitted, the horizontal axis is divided in units from 0 to length(v).
when width = 1 (the bars touch; when width = 0, the bars are simply vertical lines., the default 0.8).
example: write a Matlab code to display the histogram of an image, ~~...~~ using a bar graph, reduce the resolution of the horizontal axis into 10 bands (groups).

```
>> h = imhist(f);
>> h1 = h(1:10:256);
>> horz = 1:10:256;
>> bar(horz, h1);
>> axis([0 255 0 15000])
>> Set(gca, 'xtick', 0:50:255)
>> set (gca, 'ytick', 0:2000:15000)
```

The axis function has the syntax:

axis([horzmin horzmax vertmin vertmax])

which sets the minmum and the maximum values in the horizontal and vertical axes.

gca : get current axis (the axes of the figure last displayed).

xtick and ytick set the horizontal and vertical axes ~~of a graph using the function~~ ticks in the interval shown.

Axis labels can be added to the horizontal and vertical axes of a graph using the functions

xlabel('text string', 'fontsize', size)
ylabel('text string', 'font size', size).

Text can be added to the body of the figure using function:

text(xloc, yloc, 'text string', 'font size', size)

where: xloc and yloc define the location where text starts.

* Note that functions that set axis values and labels are used after the function has been plotted.

a title can be added to a plot using:

title('title string').

(2) a **Stem graph** can be used to display the histogram

stem(horz, v, 'color_linestyle_marker', 'fill')

horz and v as in bar function.

'color_line_style_marker' - triple values for color, line style and marker. (See Matlab help documentation).

stem(v, 'r--s') - produces a stem plot where the lines and the markers are red, the line are dashed and the marker are square.

* 'fill' - the marker is filled with color.

example:

```
>> h = imhist(f);
>> h1 = h(1:10:256);
>> horz = 1:10:256;
>> stem(horz, h1, 'fill')
>> axis([0 255 0 15000])
>> Set(gca, 'xtick', [0:50:255])
>> Set(gca, 'ytick', [0:2000:15000])
```

a (3) **plot graph** can be used to display the histogram with straight lines, the syntax is:

plot(horz, v, 'color_linestyle_marker').

the arguments are defined previously.

example:

```
>> h = imhist(f);
>> plot(h) % Use the default values
>> axis([0 255 0 15000])
>> set(gca, 'xtick', [0:50:255])
>> Set(gca, 'ytick', [0:2000:15000])
```

We can set the limits and ticks automatically by using:

xlim('auto')

ylim('auto')

See Matlab help!!

Histogram Equalization :

a- for intensity levels that are ~~continuous~~ continuous quantities normalized to the range [0, 1].

Let $P_r(r)$ - probability density function (PDF) of the intensity levels.

the following transformation on the input levels to obtain output levels, s :

$$S = T(r) = \int_0^r P_r(w)\,dw$$ ⇒ w - dummy variable of integration

the PDF of the output levels is uniform :

$$P_s(s) = \begin{cases} 1 & 0 \leq s \leq 1 \\ 0 & \text{otherwise} \end{cases}$$ → image, whose intensity levels are equally likely.

and it cover the entire range [0, 1].

this transformation is called intensity- levels equalization process (and it's nothing more than the cumulative distribution function (CDF).

b- discrete case (quantities):

the equalization transformation becomes:

$$S_K = T(r_K) = \sum_{j=1}^{K} P_r(r_j) = \sum_{j=1}^{K} \frac{n_j}{n}, \quad \text{for } K = 1, 2, \ldots, L$$

S_K - intensity value of the output image corresponding to value r_K in the input image.

Histogram Equalization : is a method which increases the dynamic range of the gray-level in a low-contrast image to cover full range of gray-levels.

Histogram equalization is achived by having a transformation function T(r) which can be defined to be the cummulative distribution function (CDF) of a given probability density function (PDF) of a gray-levels in a given image. (the histogram of an image can be considered as the approximation of the PDF of that image

Matlab Implementation

Histogram equalization is implemented in the toolbox by function histeq, which has the syntax:

g = histeq(f, nlev).

where f is the input image, nlev - number of intensity levels specified for the output image: (nlev = L (total number of possible level), then histeq implement the transformation function T(rk) directly), nlev less than L - the histeq attempts to distribute the levels, so that they will approximate a flat histogram. (nlev = 64 is the default)

example1:

```
>> imshow(f)
>> figure, imhist(f)
>> ylim('auto')
>> g = histeq(f, 256);
>> figure, imshow(g)
>> figure, imhist(g)
>> ylim('auto')
```

example 2:

as noted earlier, the transformation function T(rk) is simply the cumulative sum of normalized histogram values, we can use the function cumsum to obtain the transformation function, type:

```
>> hnorm = imhist(f)./numel(f);
>> cdf = cumsum(hnorm);
```

example 3: a plot of cdf (for example 2:) can be obtained using the following commands:

```
>> x = linspace(0, 1, 256); % Intervals for [0,1] horiz scale
>> plot(x, cdf)        % plot cdf vs. x
>> axis([0 1 0 1])     % Scale, settings and labels
>> set(gca, 'xtick', 0:.2:1)
>> set(gca, 'ytick', 0:.2:1)
>> xlabel('Input Intensity values', 'fontsize', 9)
>> ylabel('output Intensity values', 'fontsize', 9)
>> % specify text in the body of the graph:
>> text(0.18, 0.5, 'Transformation function', 'fontsize', 9)
```

Example 4 : (Manual Calculation):

Consider an 8-bit grayscale image has the following values:

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

→ 8 × 8 subimage.

The histogram for this image is show in the following table. pixels values that have a zero count are excluded.

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

The cumulative distribution function (cdf) is show below.

Values	cdf	Values	cdf	Values	cdf	Values	cdf	Value	cdf
52	1	64	19	72	40	85	51	113	60
55	4	65	22	73	42	87	52	122	61
58	6	66	24	75	43	88	53	126	62
59	9	67	25	76	44	90	54	144	63
60	10	68	30	77	45	94	55	154	64
61	14	69	33	78	46	104	57		
62	15	70	37	79	48	106	58		
63	17	71	39	83	49	109	59		

This cdf table shows that the minimum value in the subimage is 52 and the maximum value is 154.

$cdf(52) = 1$, $cdf(154) = 64$.

The cdf must be normalized to [0, 255]. The general histogram equalization formula is:

$$h(v) = \text{round}\left(\frac{cdf(v) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L-1)\right)$$

where: cdf_{min} - the minimum value of the cdf

$M \times N$: image's number of pixels. (M-width, N-Height)

L - number of gray scale levels (in most cases, 256)

The equalization formula for this particular example, is

$$h(v) = \text{round}\left(\frac{cdf(v) - 1}{63} \times 255\right)$$

for example: the cdf of 78 is 46, the normalized value becomes:

$$h(78) = \text{round}\left(\frac{46-1}{63} \times 255\right) = \text{round}(0.714286 \times 255) = 182.$$

We use the above formula to calculate the normalized cdf and the values of the equalized image are directly taken from normalized cdf), the following table contains the normalized cdf.

0	12	53	93	146	53	73	166
65	32	12	215	235	202	130	158
57	32	117	239	251	227	93	166
65	20	154	243	255	231	146	130
97	53	117	227	247	210	117	146
190	85	36	146	178	117	20	170
202	154	73	32	12	53	85	194
206	190	130	117	85	174	182	219

* Notice that the minimum value (52) is now 0 and the maximum value (154) is now 255.