

تجزیه و تحلیل داده ها در نساجی

پردازش تصویر

دکتر پدرام پیوندی

بخش سوم

www.pedram-payvandy.com

انواع تصویر

1. Intensity Images (Gray-Scale)
2. Binary Images
3. RGB Images (True color)
4. Indexed Images

www.pedram-payvandy.com

2

انواع تصویر

- Intensity Images (Gray- Scale):

به تصویری گفته می‌شود که تنها دارای مقادیر روشنایی است و خصوصیات رنگ مانند فام و خلوص را ندارد. در Matlab این تصاویر توسط ماتریس‌های دوبعدی تعریف می‌شوند به‌طوری‌که مقدار هر عنصر از این ماتریس معرف میزان روشنایی پیکسل متناظرش در تصویر مربوطه است. دامنه تغییرات عناصر این ماتریس ممکن است بین ۰ تا ۱ و یا بین ۰ تا ۲۵۵ تغییر کند.

www.pedram-payvandy.com

3

Intensity Images - Sample

Im_Gray =

0	91	123	218	84	155	110	43	192	127
0	91	123	218	84	155	110	43	192	127
0	91	123	218	84	155	110	43	192	127
0	91	123	218	84	155	110	43	192	127
0	91	123	218	84	155	110	43	192	127
0	91	123	218	84	155	110	43	192	127
0	91	123	218	84	155	110	43	192	127
0	91	123	218	84	155	110	43	192	127
0	91	123	218	84	155	110	43	192	127
0	91	123	218	84	155	110	43	192	127
0	91	123	218	84	155	110	43	192	127



www.pedram-payvandy.com

- Binary Images:

یک تصویر باینری به تصویری گفته می‌شود که هر پیکسل از آن تنها بتواند دارای یکی از دو مقدار ممکن (معمولاً ۰ و ۱) باشد، یعنی هر پیکسل می‌تواند یکی از دو رنگ سیاه (برای پس‌زمینه) و سفید (برای نمایش لبه‌ها) را داشته باشد. از آن‌جاکه برای ذخیره هر پیکسل تنها به یک بیت نیاز داریم، این‌گونه تصاویر به راحتی ذخیره می‌شوند و مقادیر آنها می‌تواند، (۰ و ۱) و یا (۰ و ۲۵۵) باشد.

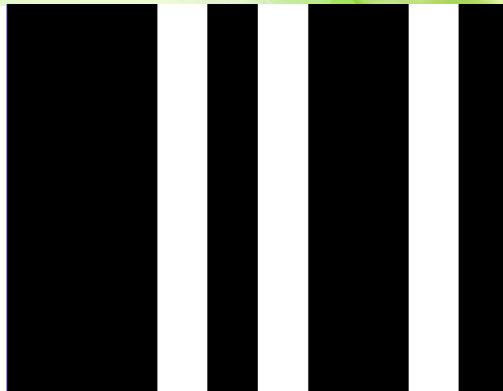
www.pedram-payvandy.com

5

Binary Images - Sample

Im_BW =

0	0	0	1	0	1	0	0	1	0
0	0	0	1	0	1	0	0	1	0
0	0	0	1	0	1	0	0	1	0
0	0	0	1	0	1	0	0	1	0
0	0	0	1	0	1	0	0	1	0
0	0	0	1	0	1	0	0	1	0
0	0	0	1	0	1	0	0	1	0
0	0	0	1	0	1	0	0	1	0
0	0	0	1	0	1	0	0	1	0
0	0	0	1	0	1	0	0	1	0



www.pedram-payvandy.com

- RGB Images (True color):

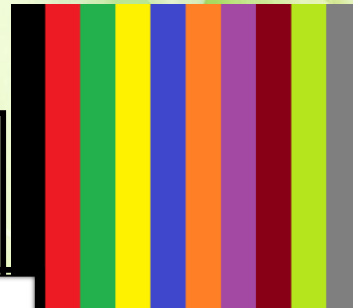
یک تصویر RGB به تصویری گفته می‌شود که به ازای هر پیکسل از آن سه عدد بین ۰ تا ۲۵۵ در حافظه کامپیوتر ذخیره می‌شود. این اعداد معرف شدت هر یک از رنگ‌های قرمز، سبز و آبی است. مثلاً برای یک پیکسل سفید سه عدد (۲۵۵، ۲۵۵، ۲۵۵) و برای یک پیکسل سبز سه عدد (۰، ۲۵۵، ۰) به ترتیب معرف شدت رنگ‌های قرمز، سبز و آبی است. بنابراین برای هر نقطه از تصویر بیش از ۱۶ میلیون (۲۵۶*۲۵۶*۲۵۶) حالت رنگی مختلف امکان‌پذیر خواهد بود. واضح است که یک تصویر RGB سه برابر یک تصویر شدتی هم‌اندازه با آن، حافظه کامپیوتر را اشغال خواهد کرد و به همان نسبت هم به زمان پردازش بیشتری نیاز دارد.

www.pedram-payvandy.com

```
val(:, :, 1) =
0 237 34 255 63 255 163 136 181 127
0 237 34 255 63 255 163 136 181 127
0 237 34 255 63 255 163 136 181 127
0 237 34 255 63 255 163 136 181 127
0 237 34 255 63 255 163 136 181 127

val(:, :, 2) =
0 28 177 242 72 127 73 0 230 127
0 28 177 242 72 127 73 0 230 127
0 28 177 242 72 127 73 0 230 127
0 28 177 242 72 127 73 0 230 127
0 28 177 242 72 127 73 0 230 127

val(:, :, 3) =
0 36 76 0 204 39 164 21 29 127
0 36 76 0 204 39 164 21 29 127
0 36 76 0 204 39 164 21 29 127
0 36 76 0 204 39 164 21 29 127
0 36 76 0 204 39 164 21 29 127
0 36 76 0 204 39 164 21 29 127
0 36 76 0 204 39 164 21 29 127
0 36 76 0 204 39 164 21 29 127
0 36 76 0 204 39 164 21 29 127
0 36 76 0 204 39 164 21 29 127
```



RGB Images -
Sample

www.pedram-payvandy.com

R: 0 G: 0 B: 0	R:237 G: 28 B: 36	R: 34 G:177 B: 76	R:255 G:242 B: 0	R: 63 G: 72 B:204	R:255 G:127 B: 39	R:163 G: 73 B:164	R:136 G: 0 B: 21	R:181 G:230 B: 29	R:127 G:127 B:127
R: 0 G: 0 B: 0	R:237 G: 28 B: 36	R: 34 G:177 B: 76	R:255 G:242 B: 0	R: 63 G: 72 B:204	R:255 G:127 B: 39	R:163 G: 73 B:164	R:136 G: 0 B: 21	R:181 G:230 B: 29	R:127 G:127 B:127
R: 0 G: 0 B: 0	R:237 G: 28 B: 36	R: 34 G:177 B: 76	R:255 G:242 B: 0	R: 63 G: 72 B:204	R:255 G:127 B: 39	R:163 G: 73 B:164	R:136 G: 0 B: 21	R:181 G:230 B: 29	R:127 G:127 B:127
R: 0 G: 0 B: 0	R:237 G: 28 B: 36	R: 34 G:177 B: 76	R:255 G:242 B: 0	R: 63 G: 72 B:204	R:255 G:127 B: 39	R:163 G: 73 B:164	R:136 G: 0 B: 21	R:181 G:230 B: 29	R:127 G:127 B:127
R: 0 G: 0 B: 0	R:237 G: 28 B: 36	R: 34 G:177 B: 76	R:255 G:242 B: 0	R: 63 G: 72 B:204	R:255 G:127 B: 39	R:163 G: 73 B:164	R:136 G: 0 B: 21	R:181 G:230 B: 29	R:127 G:127 B:127
R: 0 G: 0 B: 0	R:237 G: 28 B: 36	R: 34 G:177 B: 76	R:255 G:242 B: 0	R: 63 G: 72 B:204	R:255 G:127 B: 39	R:163 G: 73 B:164	R:136 G: 0 B: 21	R:181 G:230 B: 29	R:127 G:127 B:127
R: 0 G: 0 B: 0	R:237 G: 28 B: 36	R: 34 G:177 B: 76	R:255 G:242 B: 0	R: 63 G: 72 B:204	R:255 G:127 B: 39	R:163 G: 73 B:164	R:136 G: 0 B: 21	R:181 G:230 B: 29	R:127 G:127 B:127
R: 0 G: 0 B: 0	R:237 G: 28 B: 36	R: 34 G:177 B: 76	R:255 G:242 B: 0	R: 63 G: 72 B:204	R:255 G:127 B: 39	R:163 G: 73 B:164	R:136 G: 0 B: 21	R:181 G:230 B: 29	R:127 G:127 B:127
R: 0 G: 0 B: 0	R:237 G: 28 B: 36	R: 34 G:177 B: 76	R:255 G:242 B: 0	R: 63 G: 72 B:204	R:255 G:127 B: 39	R:163 G: 73 B:164	R:136 G: 0 B: 21	R:181 G:230 B: 29	R:127 G:127 B:127
R: 0 G: 0 B: 0	R:237 G: 28 B: 36	R: 34 G:177 B: 76	R:255 G:242 B: 0	R: 63 G: 72 B:204	R:255 G:127 B: 39	R:163 G: 73 B:164	R:136 G: 0 B: 21	R:181 G:230 B: 29	R:127 G:127 B:127

www.pedram-payvandy.com

• Indexed Images

این تصاویر توسط دو ماتریس زیر مشخص می‌شوند:

1. ماتریس اندیس: ابعاد این ماتریس برابر با ابعاد تصویر برحسب پیکسل است. مقادیر این ماتریس معمولاً بین ۱ تا ۲۵۶ تغییر می‌کند و مقدار هر درایه از این ماتریس معرف شماره سطر از ماتریس نقشه‌رنگ است.

2. ماتریس نقشه‌رنگ: این ماتریس دارای ۳ ستون است و هر سطر از آن معرف یکی از رنگ‌های موجود در تصویر است. به‌طوری‌که عنصر اول هر سطر معرف قرمز، عنصر دوم معرف سبز و عنصر سوم معرف آبی است.

یک تصویر اندیس‌شده بسته به مقادیر ماتریس نقشه‌رنگ، ممکن است رنگی یا Gray-Scale باشد.

www.pedram-payvandy.com

10

Indexed Images - Sample



Indx =

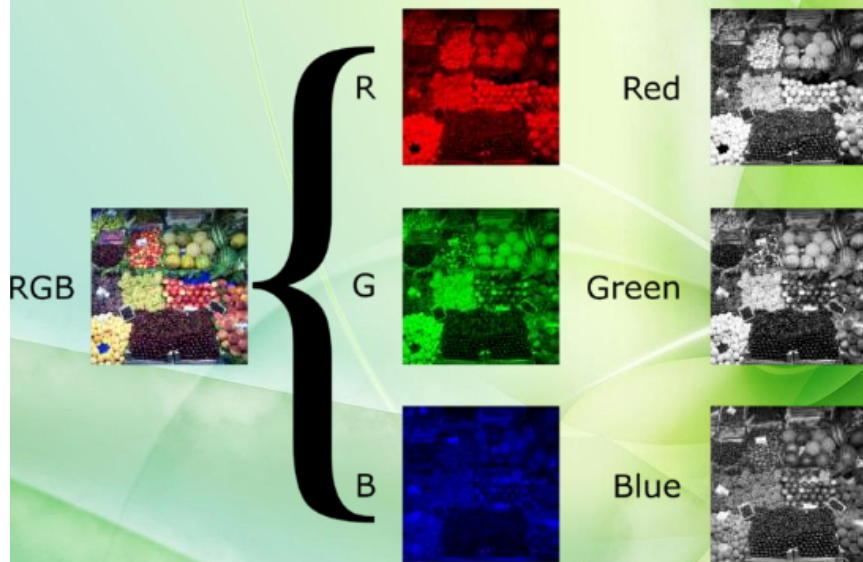
0	0	0
0.9294	0.1098	0.1412
0.1333	0.6941	0.2980
0.7098	0.9020	0.1137
0.2471	0.2824	0.8000
0.4980	0.4980	0.4980
0.5333	0	0.0824
1.0000	0.4980	0.1529
1.0000	0.9490	0
0.6392	0.2863	0.6431

Map =

0	1	2	8	4	7	9	6	3	5
0	1	2	8	4	7	9	6	3	5
0	1	2	8	4	7	9	6	3	5
0	1	2	8	4	7	9	6	3	5
0	1	2	8	4	7	9	6	3	5
0	1	2	8	4	7	9	6	3	5
0	1	2	8	4	7	9	6	3	5
0	1	2	8	4	7	9	6	3	5
0	1	2	8	4	7	9	6	3	5
0	1	2	8	4	7	9	6	3	5
0	1	2	8	4	7	9	6	3	5

www.pedram-payva

تبدیل تصویر رنگی به خاکستری



www.pedram-payvandy.com

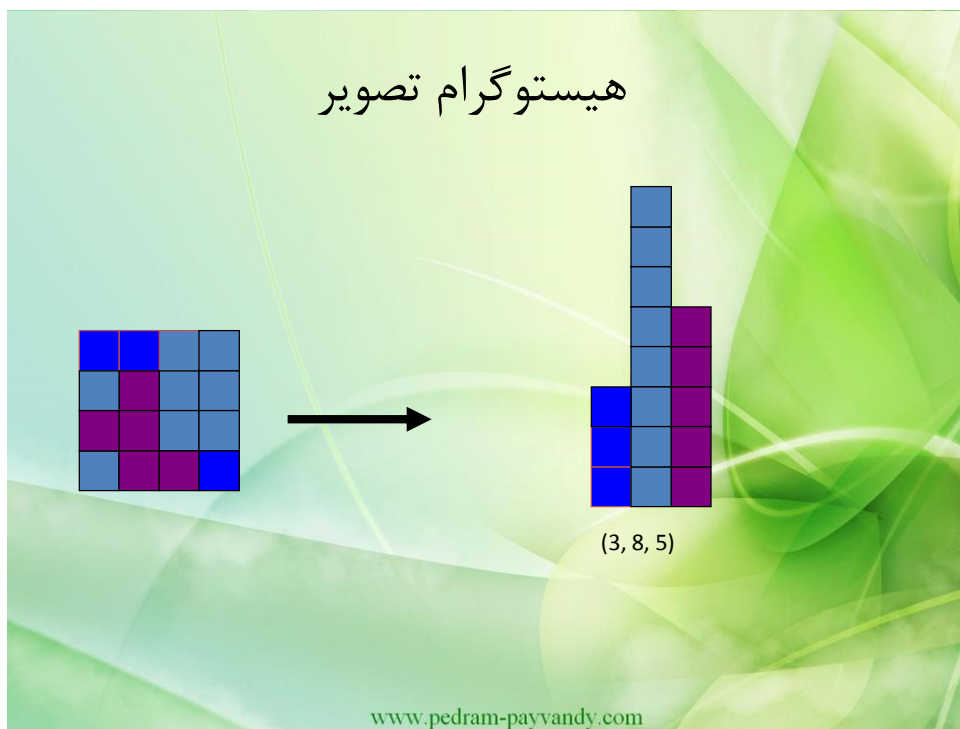


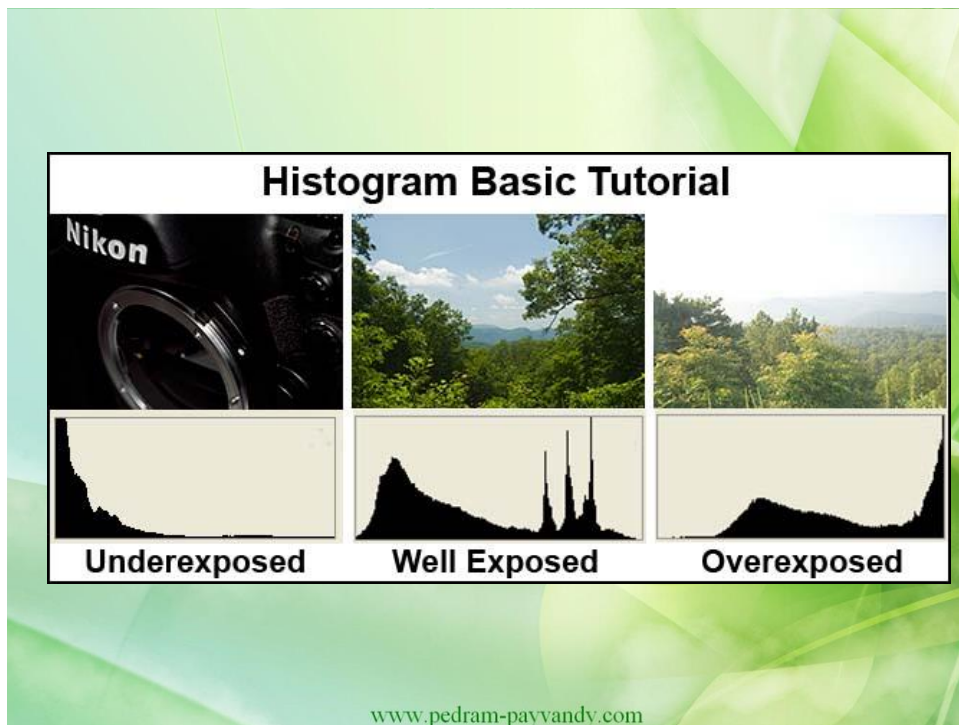
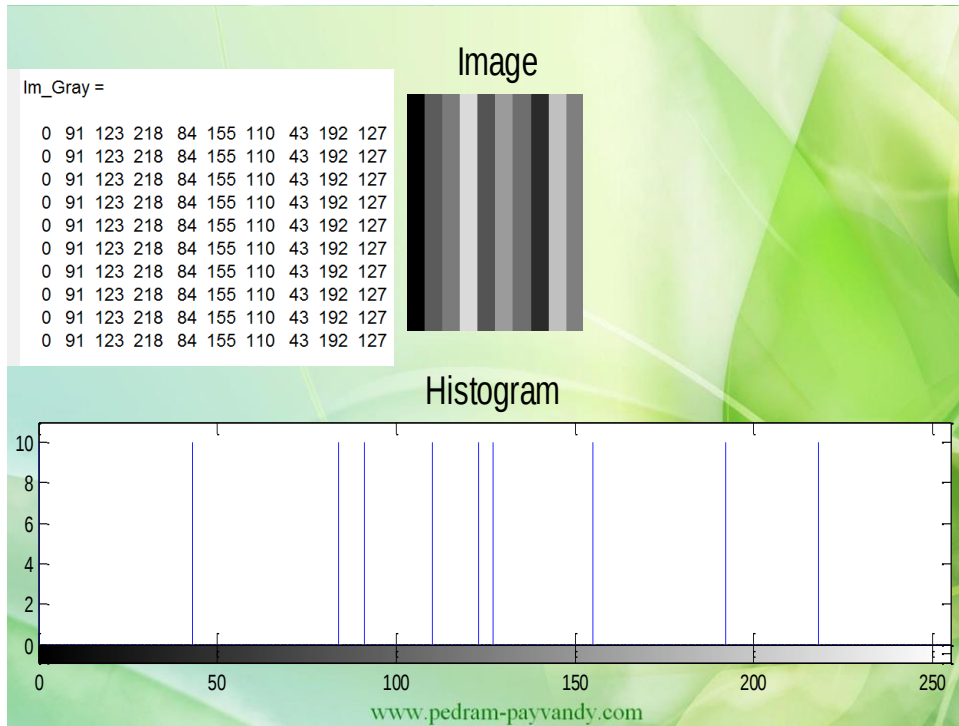
تبدیل تصویر رنگی به خاکستری

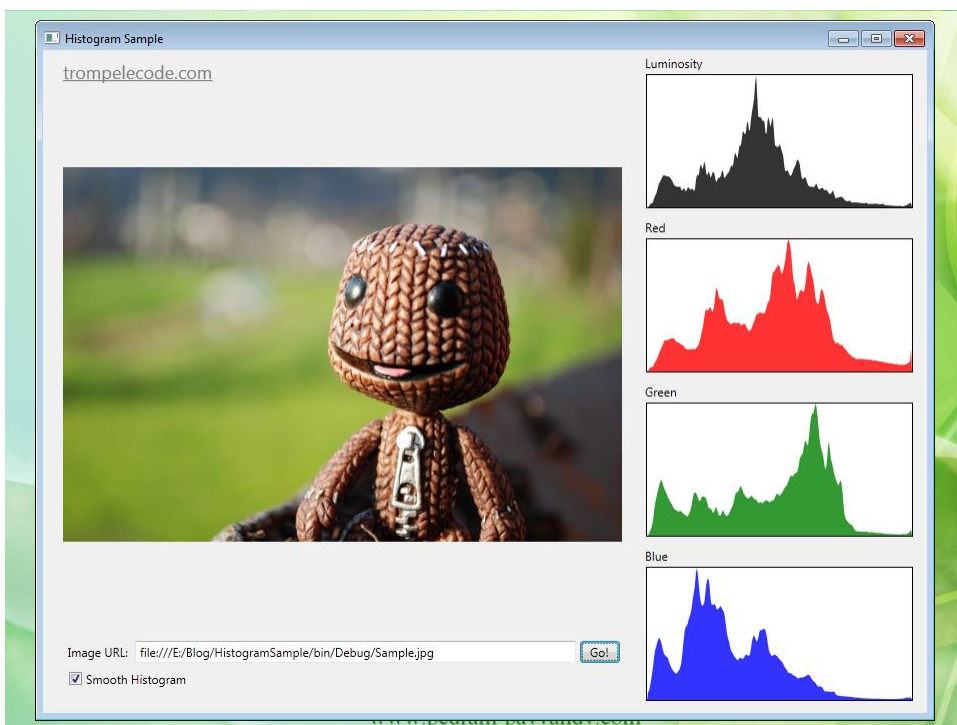
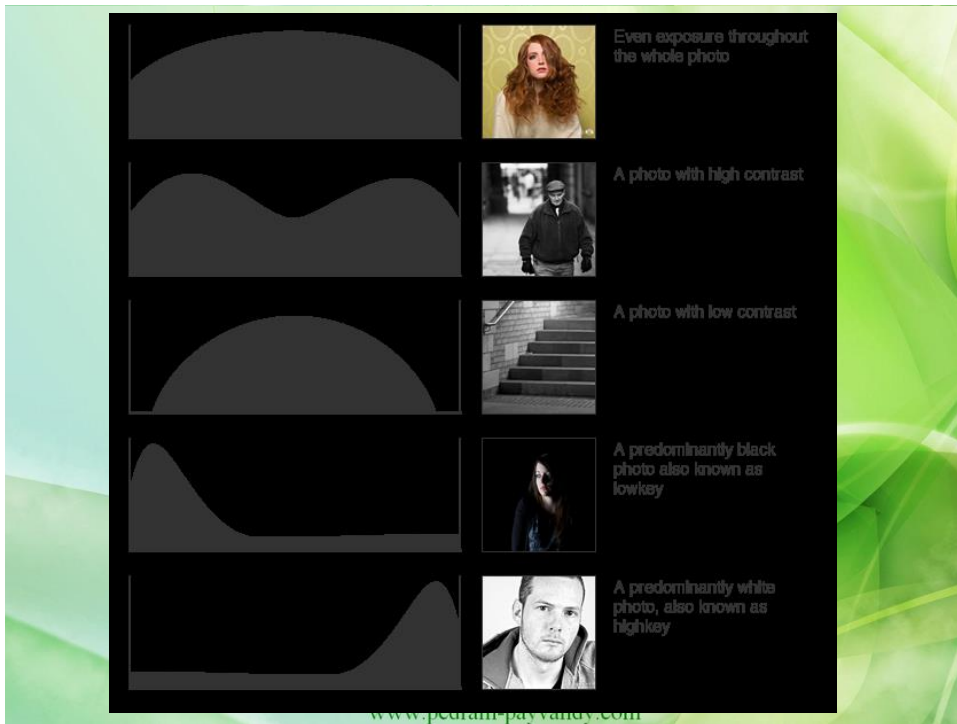


$$Y' = 0.299R' + 0.587G' + 0.114B'$$

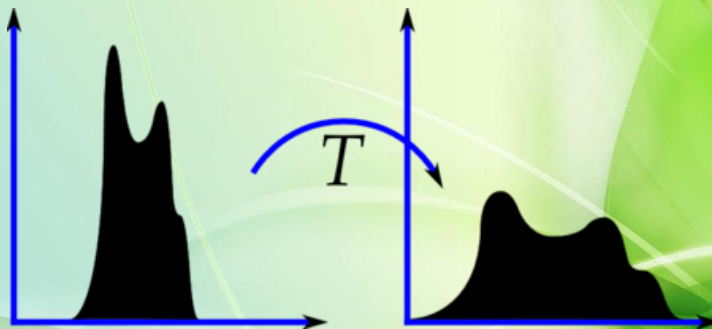
www.pedram-payvandy.com



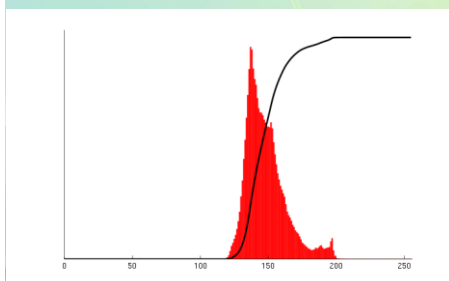




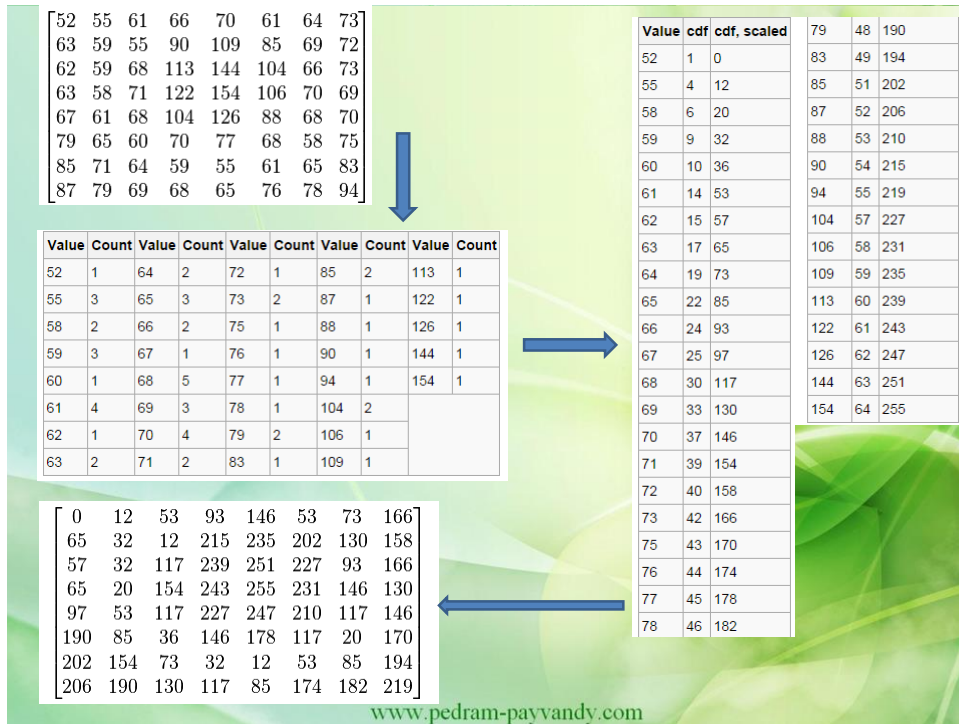
تنظیم هیستوگرام



www.pedram-payvandy.com



www.pedram-payvandy.com



This cdf shows that the minimum value in the subimage is 52 and the maximum value is 154. The cdf of 64 for value 154 coincides with the number of pixels in the image. 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} is the minimum non-zero value of the cumulative distribution function (in this case 1), $M \times N$ gives the image's number of pixels (for the example above 64, where M is width and N the height) and L is the number of grey levels used (in most cases, like this one, 256).

Note that to scale values in the original data that are above 0 to the range 1 to $L-1$, inclusive, the above equation would instead be:

$$h(v) = \text{round} \left(\frac{cdf(v) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L - 2) \right) + 1$$

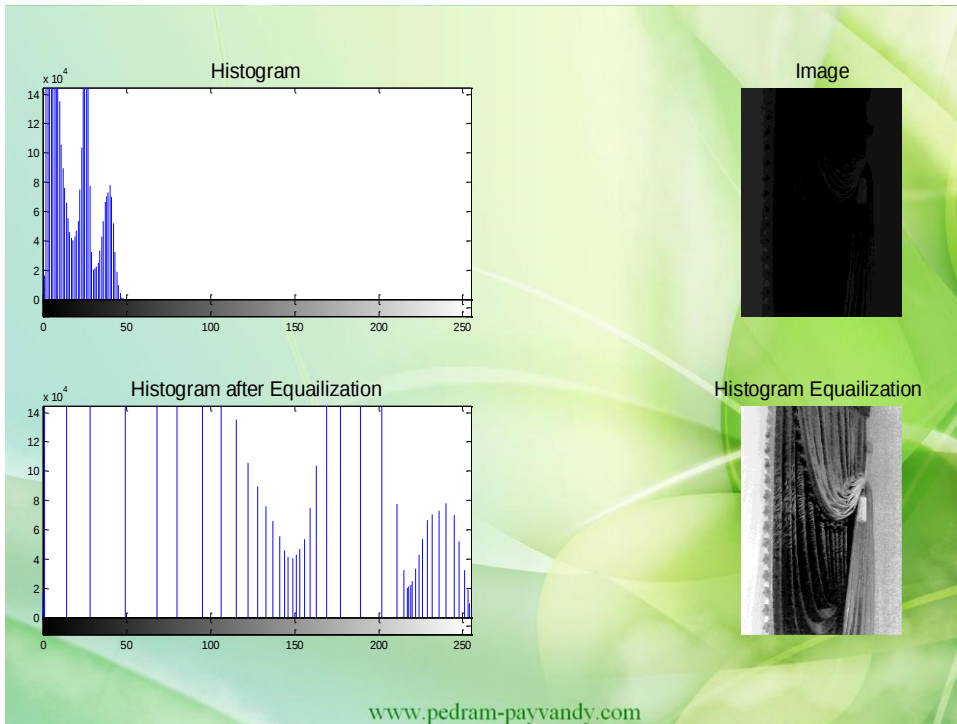
where $cdf(v) > 0$. Scaling from 1 to 255 preserves the non-zero-ness of the minimum value.

The equalization formula for the example scaling data from 0 to 255, inclusive, is:

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

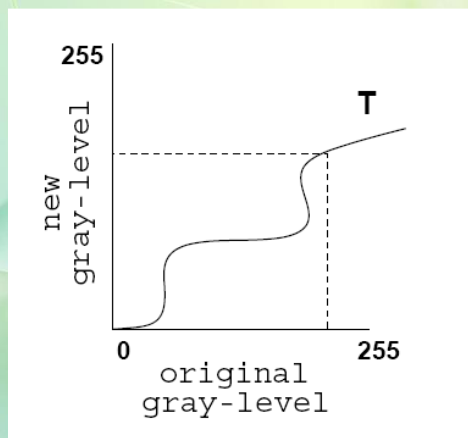
For example, the cdf of 78 is 46. (The value of 78 is used in the bottom row of the 7th column.) The normalized value becomes

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

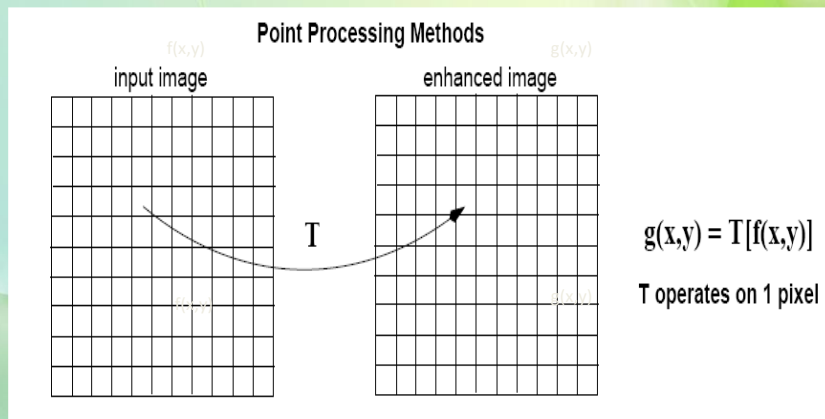


Point Processing Transformations

- Convert a given pixel value to a new pixel value based on some predefined function.

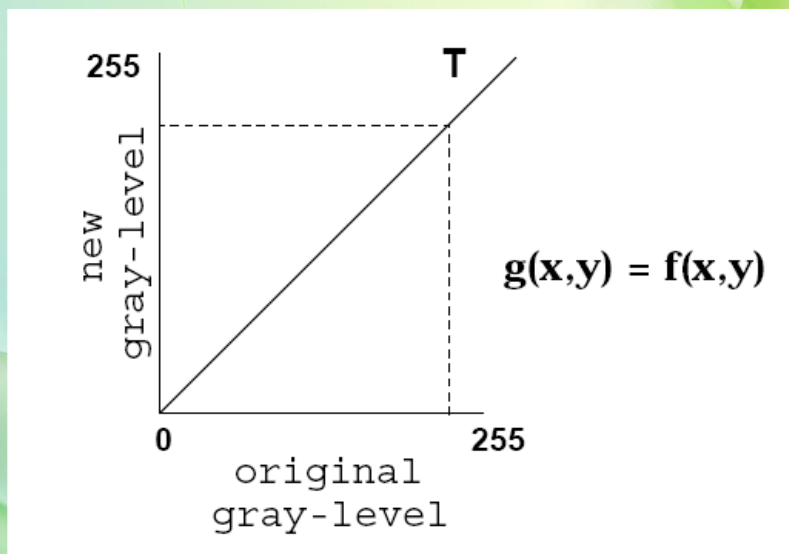


Spatial Domain Methods



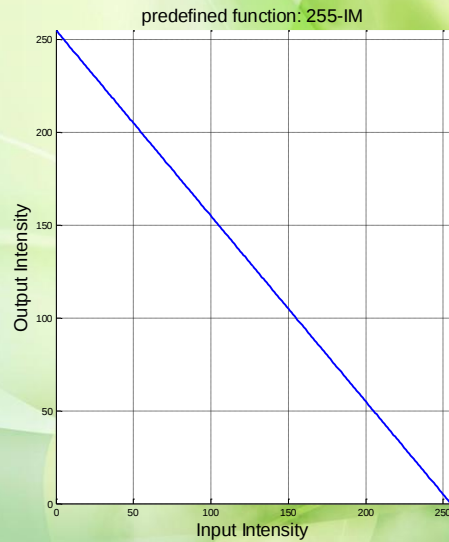
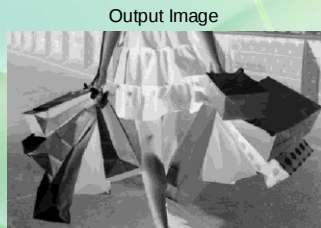
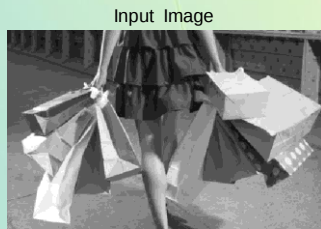
www.pedram-payvandy.com

Identity Transformation



www.pedram-payvandy.com

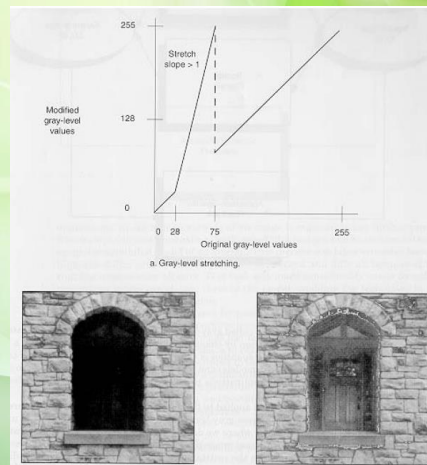
Negative Image



www.pedram-payvandy.com

Contrast Stretching/Compression

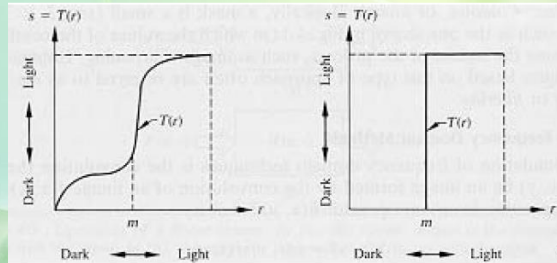
- Stretch gray-level ranges where we desire more information (slope > 1).
- Compress gray-level ranges that are of little interest ($0 < \text{slope} < 1$).



www.pedram-payvandy.com

Thresholding

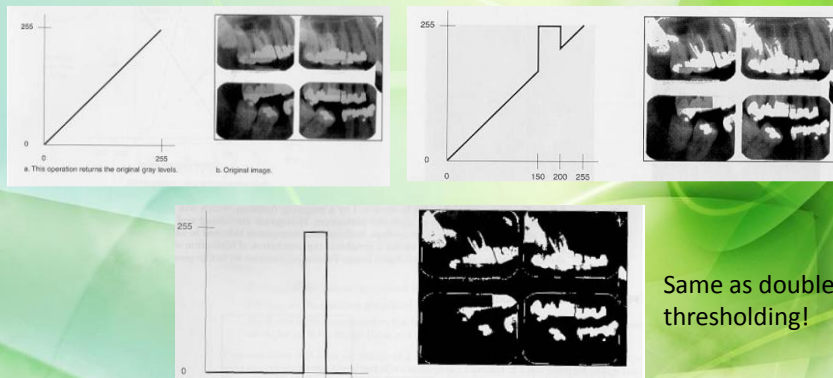
- Special case of contrast compression



www.pedram-payvandy.com

Intensity Level Slicing

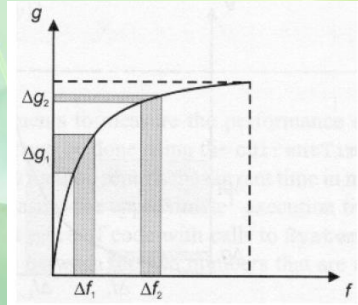
- Highlight specific ranges of gray-levels only.



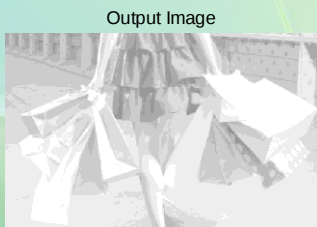
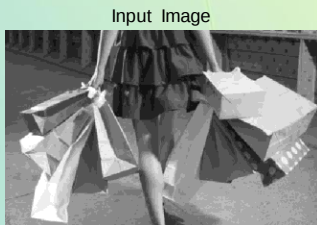
www.pedram-payvandy.com

Logarithmic transformation

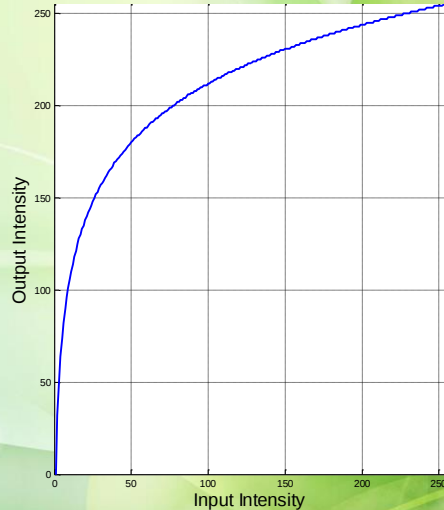
- Enhance details in the darker regions of an image at the expense of detail in brighter regions.



www.pedram-payvandy.com



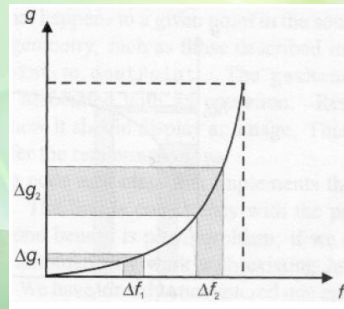
predefined function: $\text{round}((\log_{10}(\text{double}(x))/\log_{10}(255))*255)$



www.pedram-payvandy.com

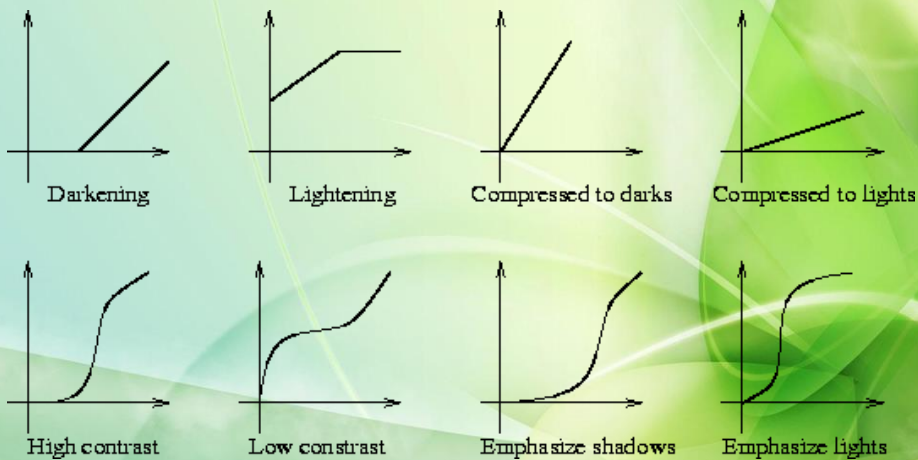
Exponential transformation

- Reverse effect of that obtained using logarithmic mapping.



www.pedram-payvandy.com

تنظیم هیستوگرام



www.pedram-payvandy.com

Addition

- Useful for combining information between two images

$$0 \leq \alpha \leq 1$$

$$O(r, c) = \alpha I_1(r, c) + (1 - \alpha) I_2(r, c)$$

www.pedram-payvandy.com

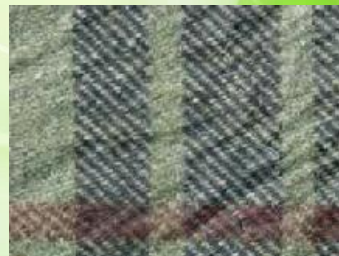
Image1



Image2



Image sumation $\alpha = 0.7$



www.pedram-payvandy.com

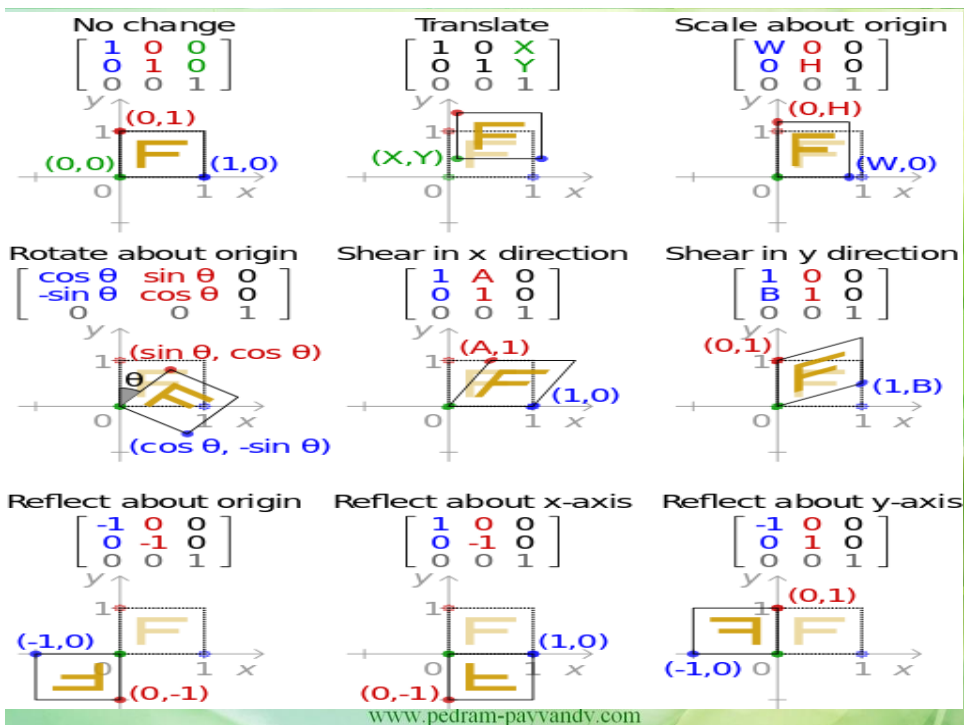
Geometric Transformations

Transformation Name	Affine Matrix, T	Coordinate Equations	Example
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = w$	
Scaling	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = c_x v$ $y = c_y w$	
Rotation	$\begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v \cos \theta - w \sin \theta$ $y = v \sin \theta + w \cos \theta$	
Translation	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$	$x = v + t_x$ $y = w + t_y$	
Shear (vertical)	$\begin{bmatrix} 1 & 0 & 0 \\ s_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v + s_v w$ $y = w$	
Shear (horizontal)	$\begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = s_h v + w$	

affine transformation

$$[x \ y \ 1] = [v \ w \ 1] \begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 1 \end{bmatrix}$$

www.pedram-payvandy.com



www.pedram-payvandy.com

Original Image



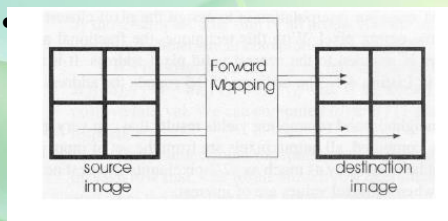
Wrapt Image



www.pedram-payvandy.com

Forward mapping

- Transformed pixel coordinates might not lie within the bounds of the image.
- Transformed pixel coordinates can be non-integer.

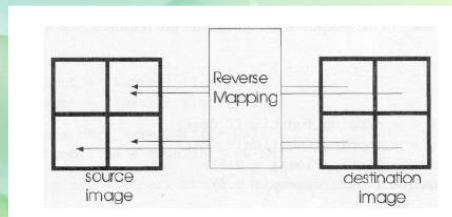


in the input image
locations in the

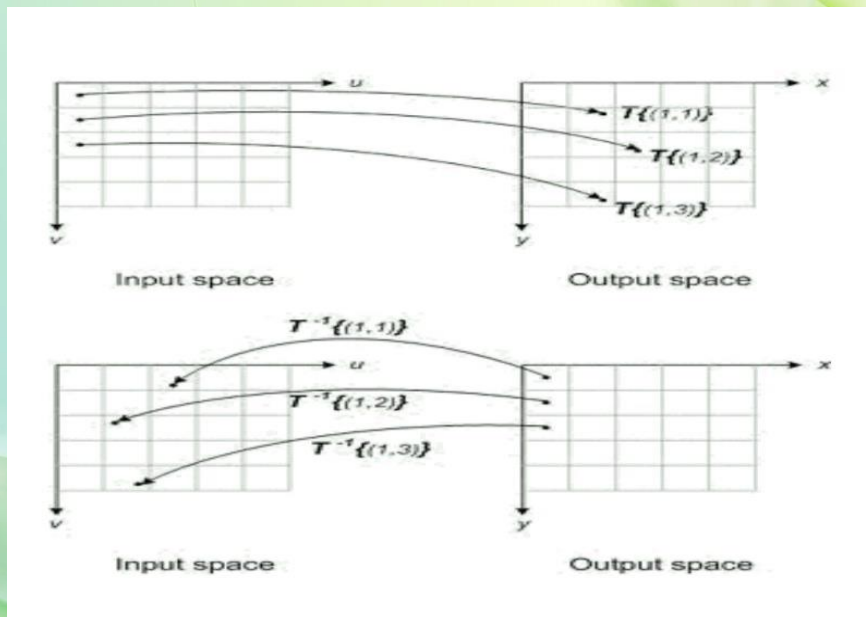
www.pedram-payvandy.com

Inverse Mapping

- To guarantee that a value is generated for every pixel in the output image, we must consider each output pixel in turn and use the *inverse* mapping to determine the position in the input image.
- To assign intensity values to these locations, we use intensity *interpolation*.



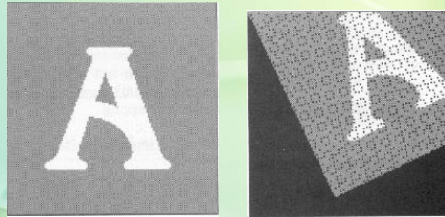
www.pedram-payvandy.com



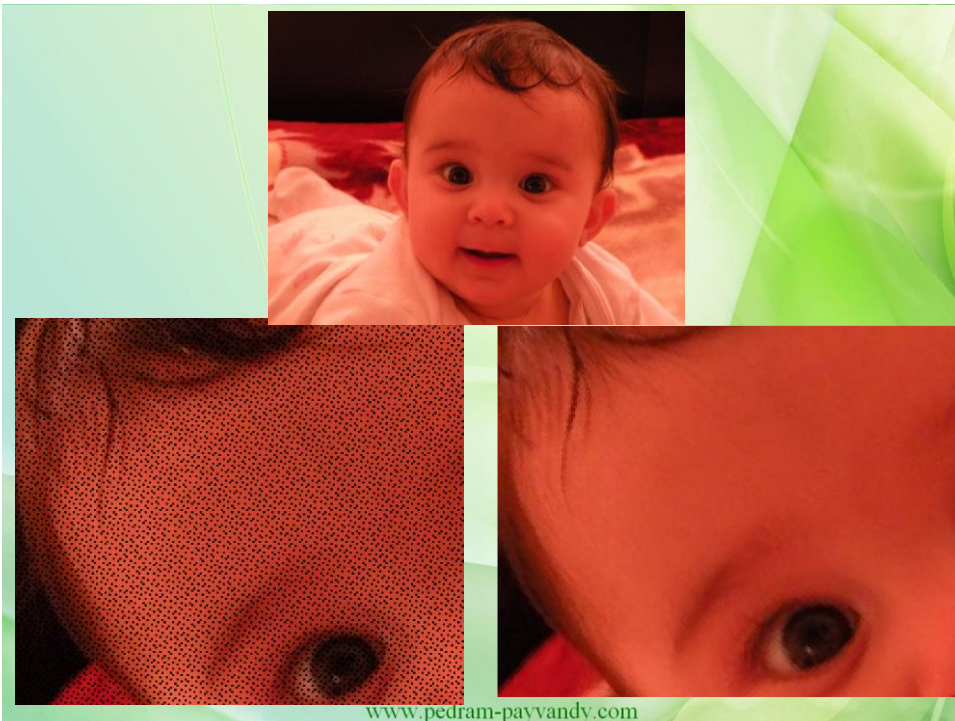
www.pedram-payvandy.com

Forward mapping (cont')

- An example of holes due to image rotation



www.pedram-payvandy.com

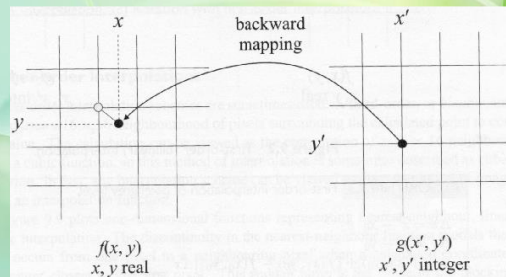


www.pedram-payvandy.com

Interpolation

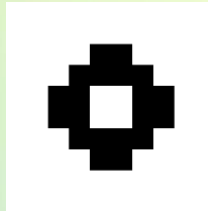
- Interpolation is the process of using known data to estimate values at unknown locations.

Zero-order interpolation (or nearest-neighbor)

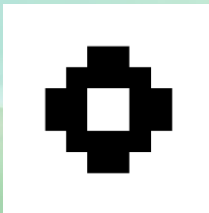


www.pedram-payvandy.com

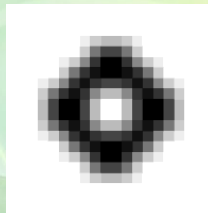
Image



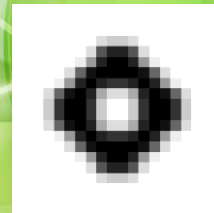
Nearest Method



bilinear Method



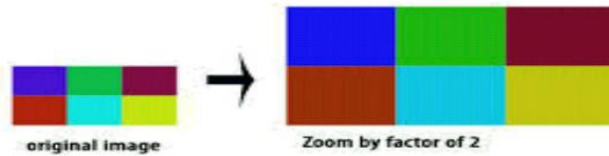
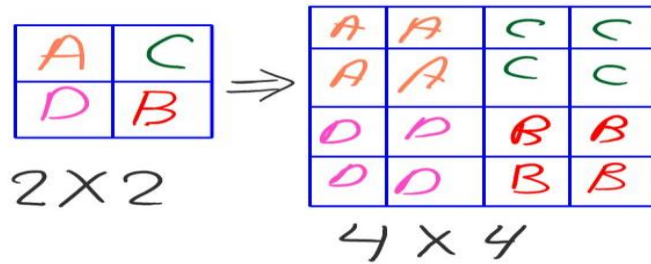
bicubic Method



www.pedram-payvandy.com

nearest neighbor interpolation

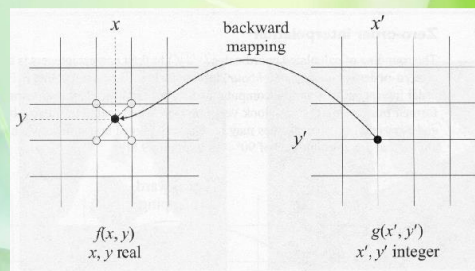
روش نزدیکترین همسایگی



Interpolation (cont'd)

First-order interpolation:

average

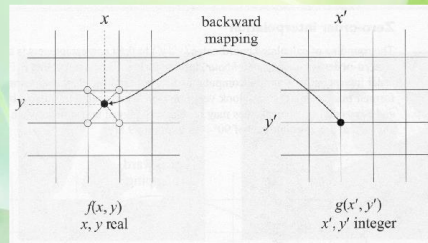


Interpolation (cont'd)

bilinear

First-order interpolation:

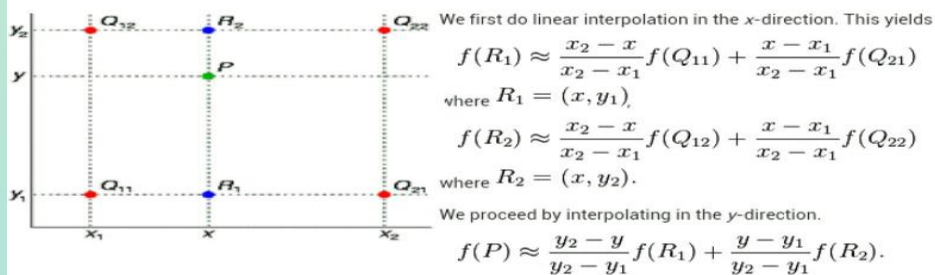
$$I(x,y) = ax + by + cxy + d$$



- The unknowns (a,b,c,d) are determined from four equations formed by the four nearest neighbors.

www.pedram-payvandy.com

bilinear Interpolation



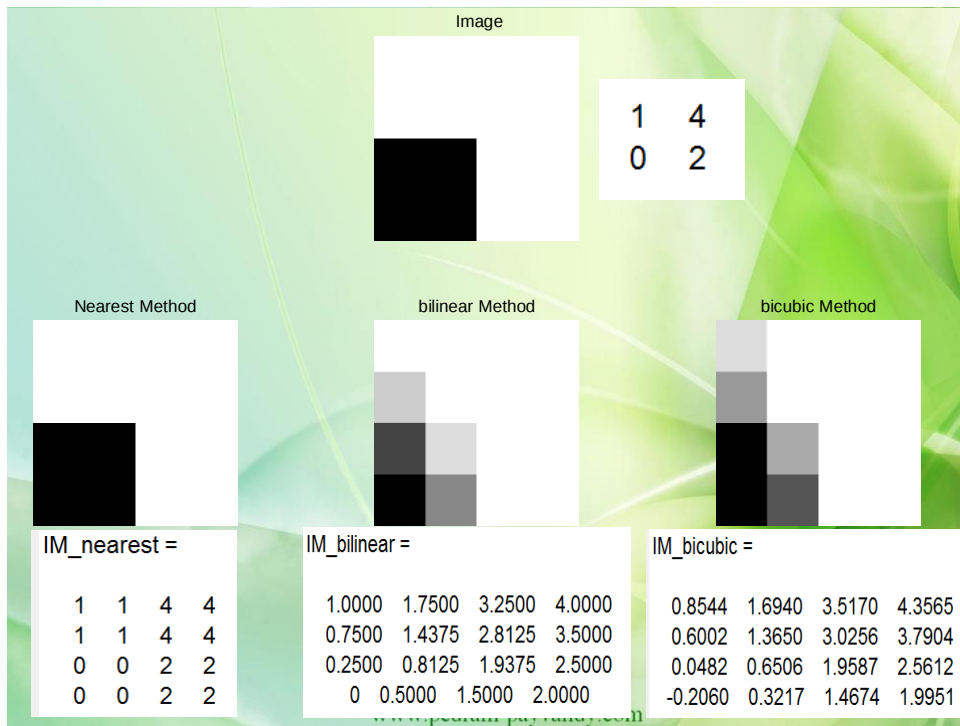
Unit Square

If we choose a coordinate system in which the four points where f is known are $(0, 0)$, $(0, 1)$, $(1, 0)$, and $(1, 1)$, then the interpolation formula simplifies to

$$f(x, y) \approx f(0, 0)(1-x)(1-y) + f(1, 0)x(1-y) + f(0, 1)(1-x)y + f(1, 1)xy.$$

Or equivalently, in matrix operations:

$$f(x, y) \approx \begin{bmatrix} 1-x & x \end{bmatrix} \begin{bmatrix} f(0, 0) & f(0, 1) \\ f(1, 0) & f(1, 1) \end{bmatrix} \begin{bmatrix} 1-y \\ y \end{bmatrix}.$$



Interpolation (cont'd)

Bicubic interpolation

- It involves the sixteen nearest neighbors of a point.
- The unknowns a_{ij} are determined from sixteen equations formed by the sixteen nearest neighbors.

$$I(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} x^i y^j$$

