SCHOOL OF COMPUTER TECHNOLOGY

AASD 4001
Mathematical Concepts for Machine Learning
Lecture 6

Reza Moslemi, Ph.D., P.Eng.

Session 5

- Image processing techniques (Image transformations)
  - Convolution
  - Scaling
  - Rotation
  - Translation
  - Smoothing

Convolution

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

6×6 Matrix

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

-1	1	-1
-1	1	-1
-1	1	-1

Filter 2

Each filter detects a small pattern.

Convolution

Stride =1 pixel

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

6×6 image

Dot product

3 -1

1	-1	-1
-1	1	-1
-1	-1	1

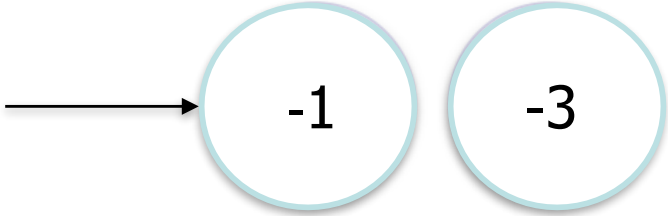
Filter 1

Convolution

Stride=2

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

6×6 image



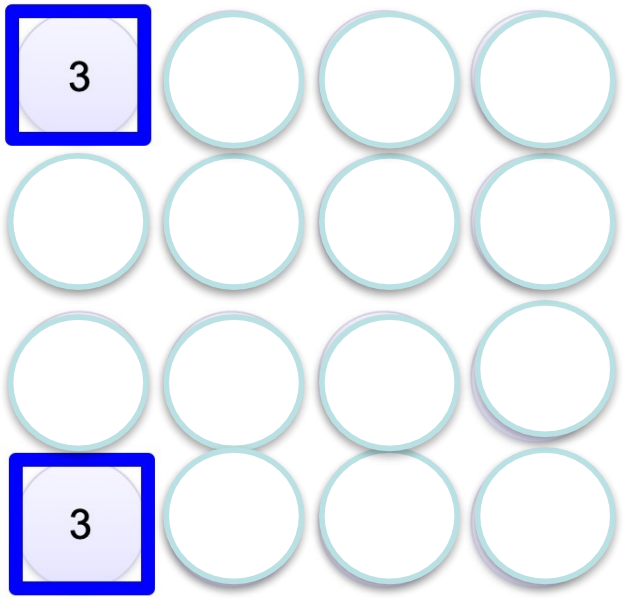
1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

Convolution

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

1	-1	-1
-1	1	-1
-1	-1	1



Exercise: Compute missing values
in the resulting matrix (avoid border pixels!)

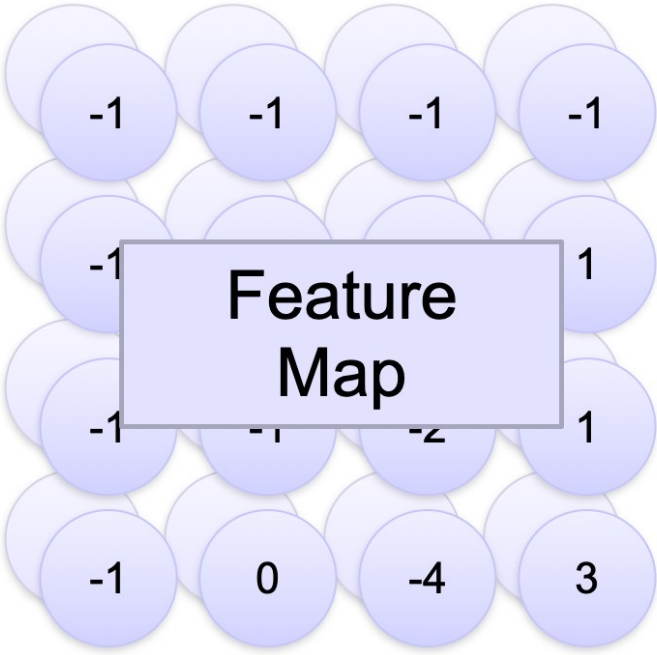
Convolution

6×6 image

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

-1	1	-1
-1	1	-1
-1	1	-1

Filter 2



Two images

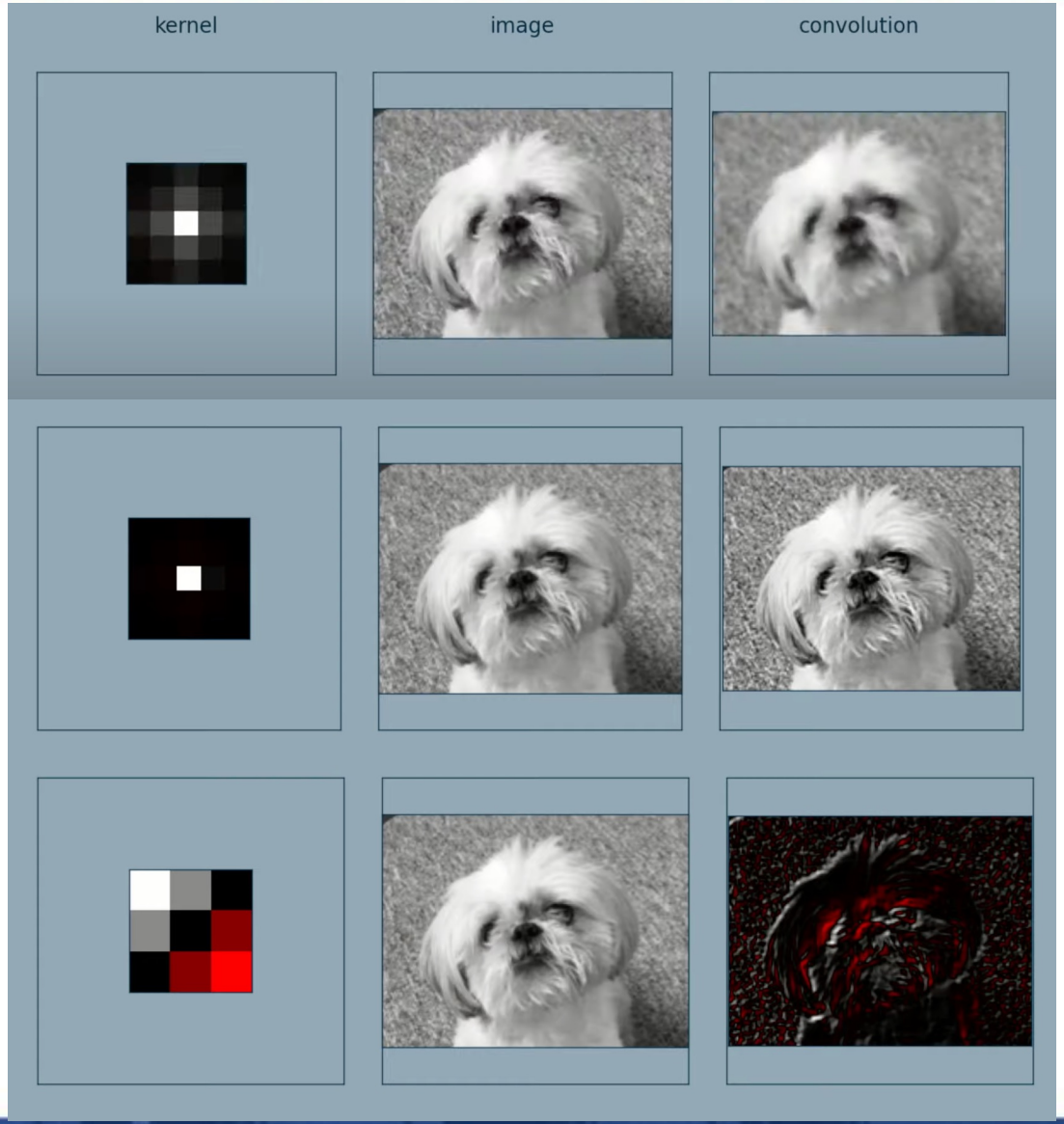
When you repeat convolution operation for each filter, you will extract features from the original image that will be then used in training Convolutional Neural Networks to detect objects.

Convolution

Blurring/smoothing kernel

Sharpening kernel

Feature detector kernel



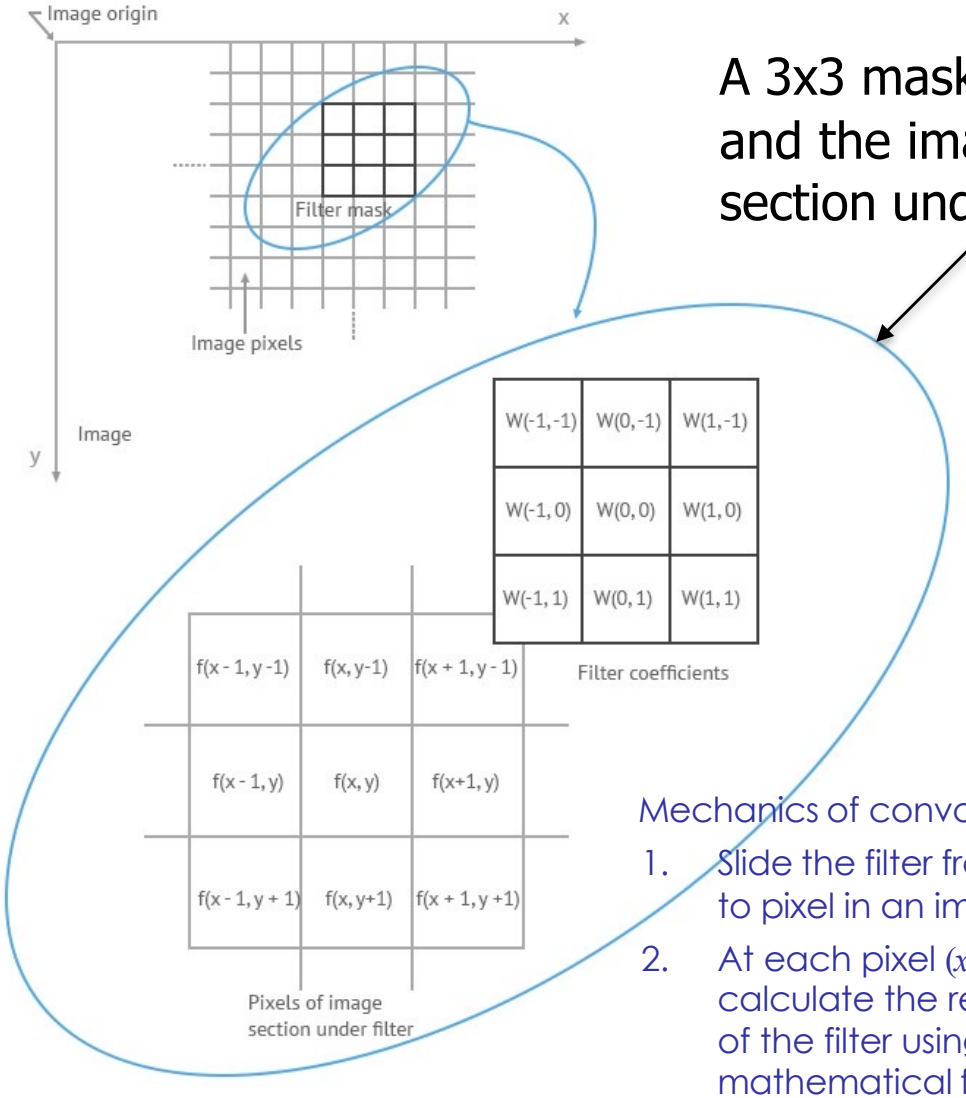
Convolution/spatial filtering

In image processing, a kernel, convolution matrix, filter, or mask is a small matrix used for blurring, sharpening, embossing, edge detection, and more. This is accomplished by doing a convolution between the kernel and an image.

Convolving a filter with an image results in extraction of features that help the computer detect an object in an image. Otherwise, by simply collapsing an image into a vector of its pixel intensities, these features would not be captured.

Convolution operations on images are widely used for image enhancement. Before we state the mathematical definition of convolution in the context of images, let's form its conceptual understanding.

Convolution is a misnomer, it actually is cross-correlation since the kernel is not flipped (reversed).



A 3x3 mask/filter and the image section under it

Mechanics of convolution:

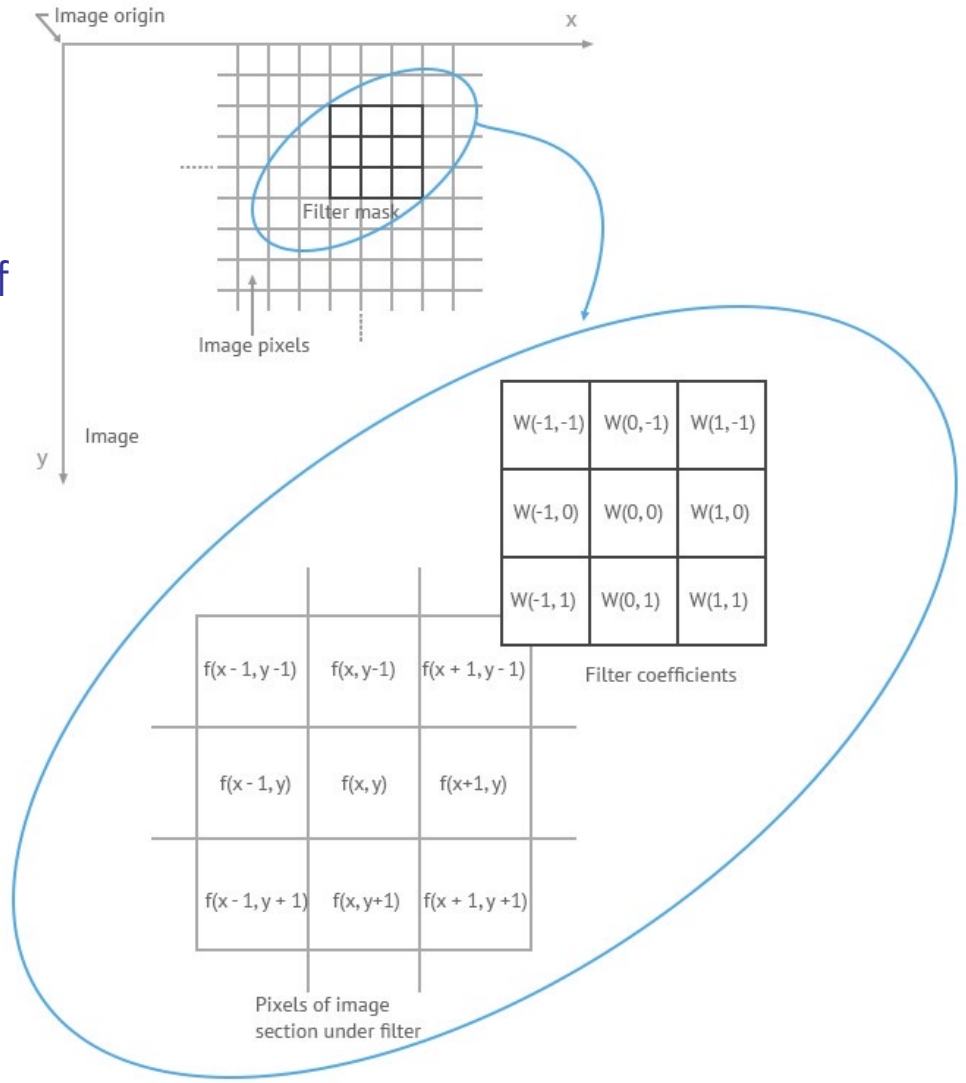
1. Slide the filter from pixel to pixel in an image,
2. At each pixel (x, y), calculate the response of the filter using a mathematical formula.

Convolution/spatial filtering

The response of the filter at pixel (x, y) is given by the sum of products of the filter coefficients and the corresponding image pixel values in the area covered by the filter mask. For the 3×3 matrix shown on the right panel, the response of filtering with with filter at a point (x, y) in the image is

$$R = w(-1, -1) f(x-1, y-1) + w(-1, 0) f(x-1, y) + \ldots$$
$$+ w(0, 0) f(x, y) + \ldots + w(1,0) f(x+1, y) + w(1, 1) f(x+1, y+1)$$

For a mask of size $m \times n$, we assume that $m=2a+1$ and $n=2b+1$, $a>0$, $b>0$. In practice, we use filters of odd sizes, with the smallest meaningful size being 3×3.



Convolution/spatial filtering

In general, filtering of an image $f(x, y)$ of size $M \times N$ with a filter mask w of size $m \times n$ is given by:

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t) \qquad (*)$$

$$\left\{ \text{Convolution:} \qquad g(x,y) = w * f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x-s, y-t) \right\}$$

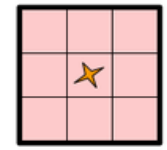
where $a = (m-1)/2$ and $b = (n-1)/2$. How do we generate a complete image? We apply this equation for $x = 0, 1, ..., M\text{-}1$ and $y = 0, 1, 2, ..., N\text{-}1$

In this way, the mask processes all pixels in the image. Such filter masks are sometimes called *convolution masks*. Equation (*) defines a convolution operation (truly a cross-correlation, convolution rotates the filter by 180° before performing multiplication) on the image $f(x, y)$. It is also referred to as *spatial filtering* as opposed to *frequency domain filtering* (Fourier transform).

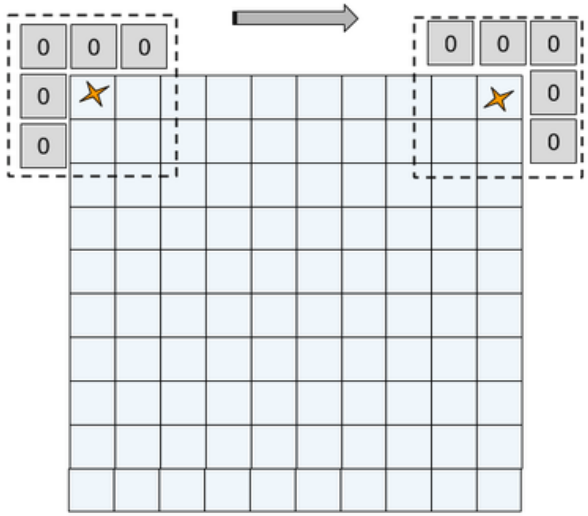
What happens when the centre of the filter approaches the border of the image? If the centre of the mask moves any closer to the border, one or more rows or columns of the mask will be located outside of the plane.

Convolution/spatial filtering

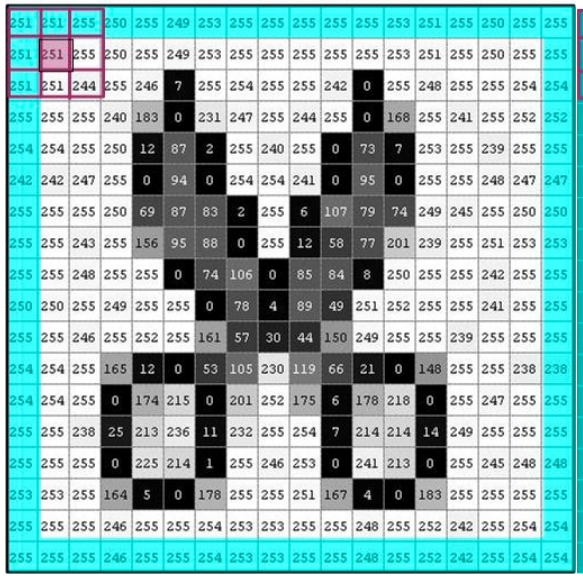
A common approach is to do image "padding" that is, to add rows and columns of 0s (or other constant pixel value) or padding by replicating rows and columns. The padding is then stripped off at the end of the process. This keeps the size of the filtered image the same as original. But the values of the padding will have an effect near the edges especially if the size of the mask is large.

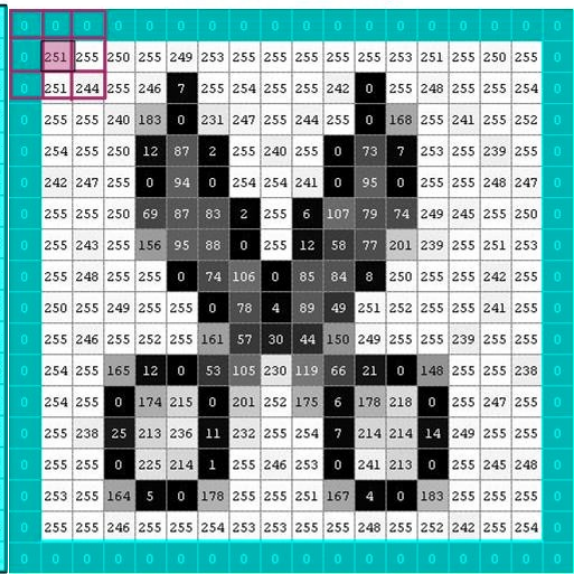


Kernel Size:
3×3



Padding





Geometric transformations of images

Geometric transformations are widely used for the removal of image artifacts and for image registration (geographical mapping).

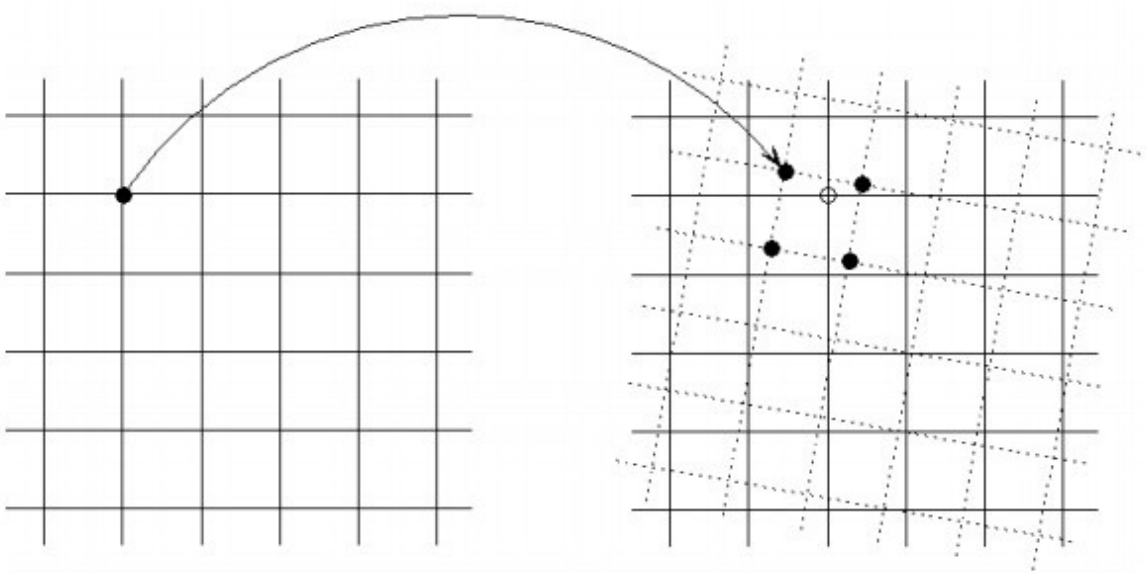
It is often necessary to perform a geometric transformation of the image coordinate system in order to:

- Align images that were taken at different times or with different sensors
- Correct images for lens distortion
- Correct effects of camera orientation
- Create special effects by morphing images

Geometric transformations of images

In a geometric/spatial transformation each point (x, y) of image $f(x, y)$ is mapped to a point (u, v) in a new coordinate system.

$$u = f_1(x, y)$$
$$v = f_2(x, y)$$



A digital image array has an implicit grid that is mapped to discrete points in the new domain. These points may not fall on grid point in the new domain.

Affine Transformation

An affine transformation is any transformation that preserves collinearity (all points lying on a line before the transformation will lie on a line after it) and ratios of distances (the midpoint of a line segment remains the midpoint after transformation).

In general, an affine transformation is a composition of rotations, translations, magnifications, and shears.

$$u = c_{11}x + c_{12}y + c_{13}$$
$$v = c_{21}x + c_{22}y + c_{23}$$

Magnifications

Translations (change in location)

The combination of all these non-zero parameters creates rotations and shears.

What is the difference between linear transformation and affine transformation?

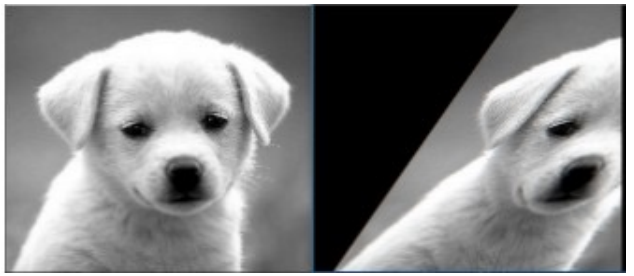
A linear transformation fixes the origin, whereas an affine transformation need not do so. An affine transformation is the composition of a linear transformation with a translation, so while the linear part fixes the origin, the translation can map it somewhere else.

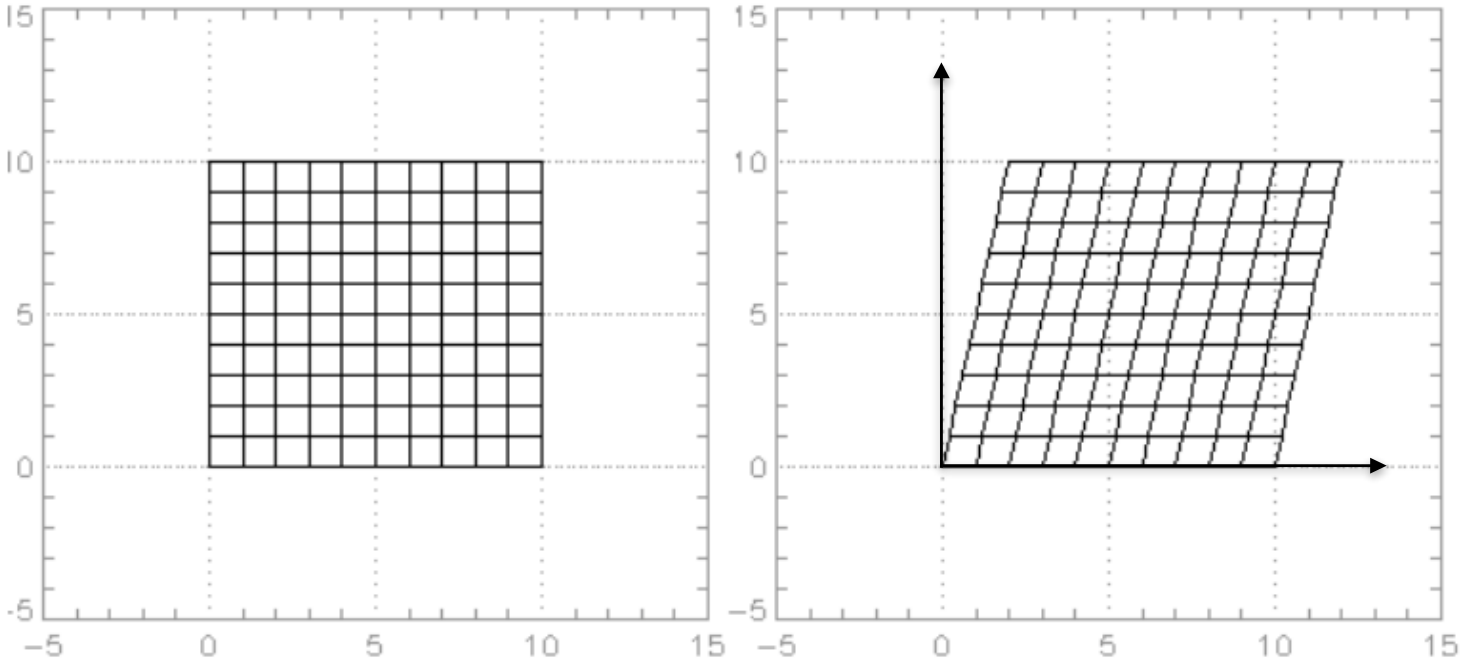
Affine Transformation

A shear in the x direction shown in the below graph is produced by

$$u = x + 0.2y$$

$$v = y$$



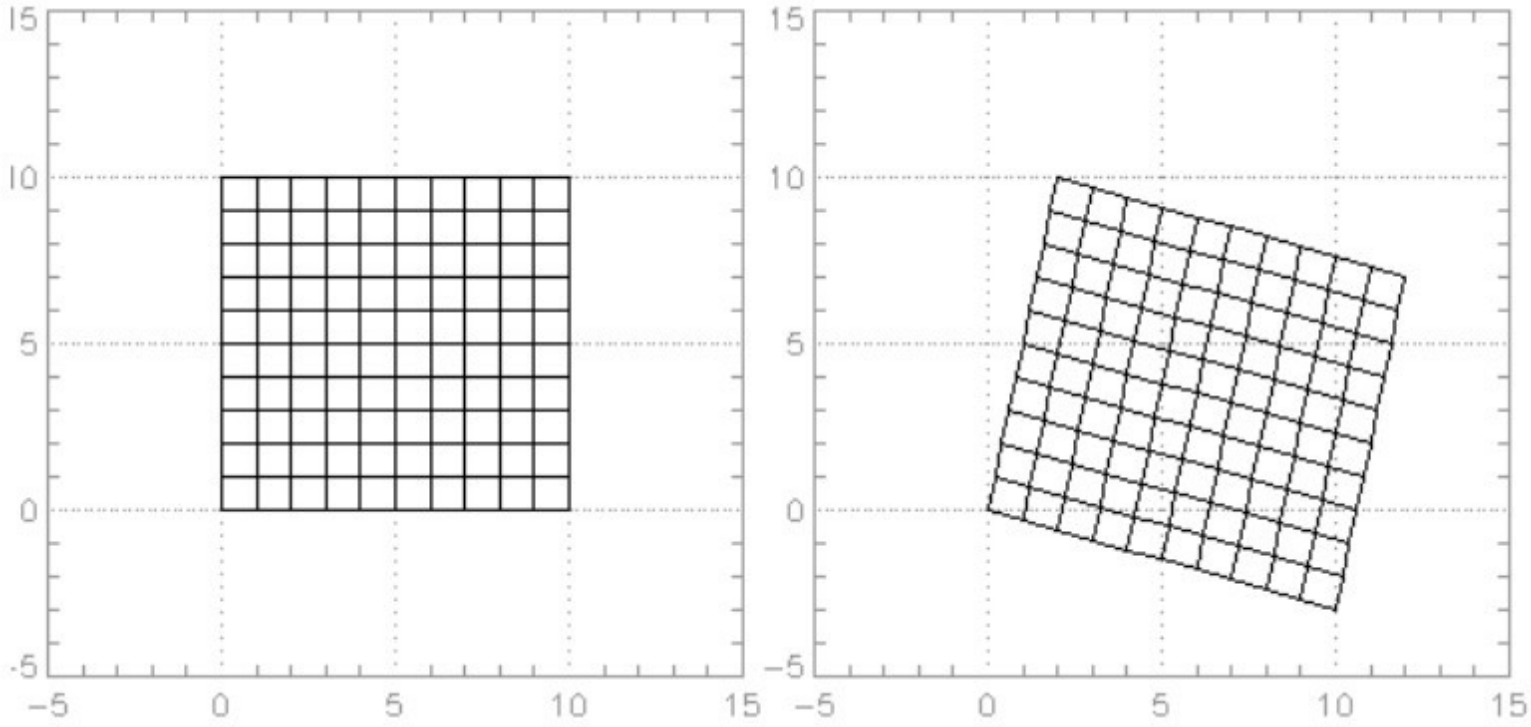


Affine transformation

This transformation produces both, a shear and a rotation.

$$u = x + 0.2y$$

$$v = -0.3x + y$$

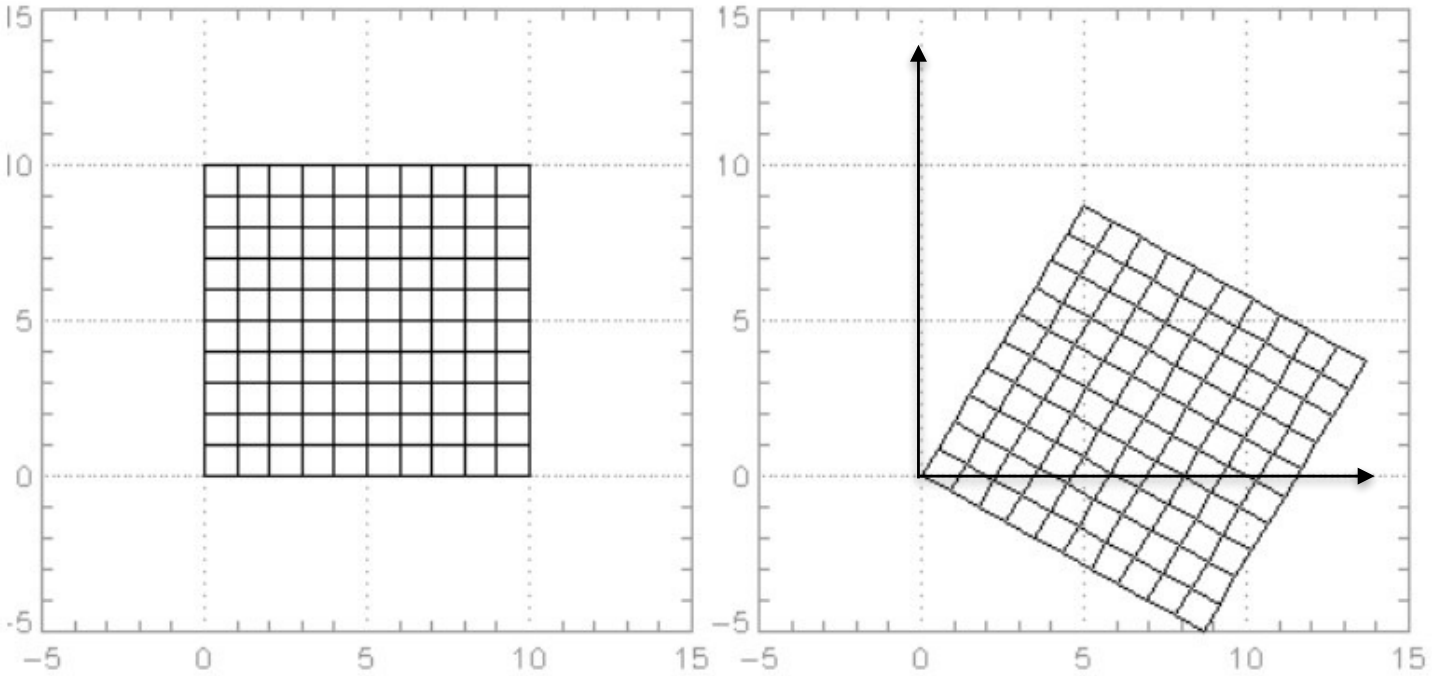


Affine Transformation

A rotation by θ is produced by

$$u = x\cos\theta + y\sin\theta$$

$$v = -x\sin\theta + y\cos\theta$$



Affine transformation

We can evolve a sequence of basic affine transformations into a complex affine transform.

Combinations of transformations are most easily described in terms of matrix operations. To use matrix operations we introduce homogeneous coordinates. These enable all affine operations to be expressed as a matrix multiplication. Otherwise, translation is an exception.

The rotation of a point, straight line or an entire image on the screen, about a point other than origin, is achieved by first moving the image until the point of rotation occupies the origin, then performing rotation, then finally moving the image to its original position.

Translation of point by the change of coordinate cannot be combined with other transformation by using simple matrix application. Such a combination is essential if we wish to rotate an image about a point other than origin by translation, rotation again translation.

To combine these three transformations into a single transformation, homogeneous coordinates are used. In homogeneous coordinate system, two-dimensional Cartesian coordinate positions (x, y) are represented by triple-coordinates $(h.x, h.y, h)$, $h \neq 0$ in homogeneous coordinates.

Affine transformation

As such, the affine equations are expressed as:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

An equivalent expression using matrix notation is

$$\mathbf{q} = \mathbf{T}\mathbf{p}$$

Where,

$$\mathbf{q} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \quad \mathbf{T} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{p} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Affine transformation

The transformation matrices can be used as building blocks.

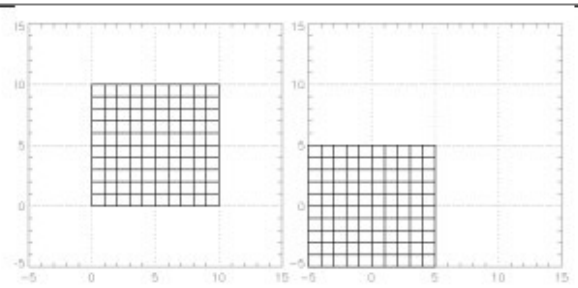
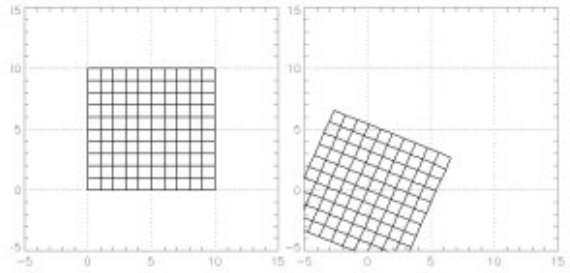
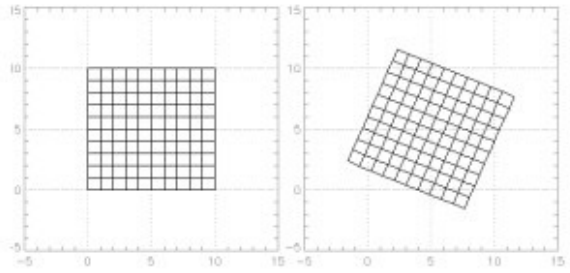
How do transformation matrices for translation and scaling look like?

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{T} = \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{T} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Translation by (x_0, y_0) Scaling by s_1 and s_2 Rotating by θ

You will usually want to translate the center of the image to the origin of the coordinate system, do any rotations and scalings, and then translate it back.

Affine transformations

Operation	Expression	Result
Translate to Origin	$\mathbf{T}_1 = \begin{bmatrix} 1.00 & 0.00 & -5.00 \\ 0.00 & 1.00 & -5.00 \\ 0.00 & 0.00 & 1.00 \end{bmatrix}$	
Rotate by 23 degrees	$\mathbf{T}_2 = \begin{bmatrix} 0.92 & 0.39 & 0.00 \\ -0.39 & 0.92 & 0.00 \\ 0.00 & 0.00 & 1.00 \end{bmatrix}$	
Translate to original location	$\mathbf{T}_3 = \begin{bmatrix} 1.00 & 0.00 & 5.00 \\ 0.00 & 1.00 & 5.00 \\ 0.00 & 0.00 & 1.00 \end{bmatrix}$	

Affine Transformations

The transformation matrix of a sequence of affine transformations, T_1, T_2, T_3 is:

$$T = T_3 T_2 T_1$$

The composite transformation for the example above is

$$T = T_3 T_2 T_1 = \begin{pmatrix} 0.92 & 0.39 & -1.56 \\ -0.39 & 0.92 & 2.35 \\ 0.0 & 0.0 & 1.0 \end{pmatrix}$$

Any combination of affine transformations in this way is an affine transformation.

The inverse transform is:

$$T^{-1} = T_1^{-1} T_2^{-1} T_3^{-1}$$

Composite Affine Transformation

Given scaling and rotation matrices, R, S, T for rotation, scaling, and translation, obtain formula for their product H=RST.

$$R = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad S = \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad T = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & x_1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$H = \begin{bmatrix} s_0\cos\theta & s_1\sin\theta & s_0x_0\cos\theta + s_1x_1\sin\theta \\ -s_0\sin\theta & s_1\cos\theta & s_1x_1\cos\theta - s_0x_0\sin\theta \end{bmatrix}$$

How to Find Transformation

Suppose that you are given a pair of images to align. You want to try an affine transform to register one to the coordinate system of the other. How do you find the transform parameters?

Image A

Image B

How to Find Transformation

Find a number of points $\{p_0, p_1, \ldots, p_{n-1}\}$ in image A that match points $\{q_0, q_1, \ldots, q_{n-1}\}$ in image B. Use the homogeneous coordinate representation of each point as a column in matrices P and Q:

$$\mathbf{P} = \begin{bmatrix} x_0 & x_1 & \cdots & x_{n-1} \\ y_0 & y_1 & \cdots & y_{n-1} \\ 1 & 1 & \cdots & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{p}_0 & \mathbf{p}_1 & \cdots & \mathbf{p}_{n-1} \end{bmatrix}$$

$$\mathbf{Q} = \begin{bmatrix} u_0 & u_1 & \cdots & u_{n-1} \\ v_0 & v_1 & \cdots & v_{n-1} \\ 1 & 1 & \cdots & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{q}_0 & \mathbf{q}_1 & \cdots & \mathbf{q}_{n-1} \end{bmatrix}$$

then we can write:

$$Q=HP$$

We need to solve for H in order to find the appropriate transformation.

Affine transformation

Matrices for two-dimensional transformation in homogeneous coordinate:

1. Translation $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$ or $\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$

2. Scaling $\begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$

3. Rotation (counter-clockwise) $\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$

4. Rotation (clockwise) $\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$

5. Reflection against X axis $\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

6. Reflection against Y axis $\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

7. Reflection against origin $\begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

8. Reflection against line Y=X $\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

9. Reflection against Y= –X $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

10. Shearing in X direction $\begin{bmatrix} 1 & 0 & 0 \\ Sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

11. Shearing in Y direction $\begin{bmatrix} 1 & Sh_y & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

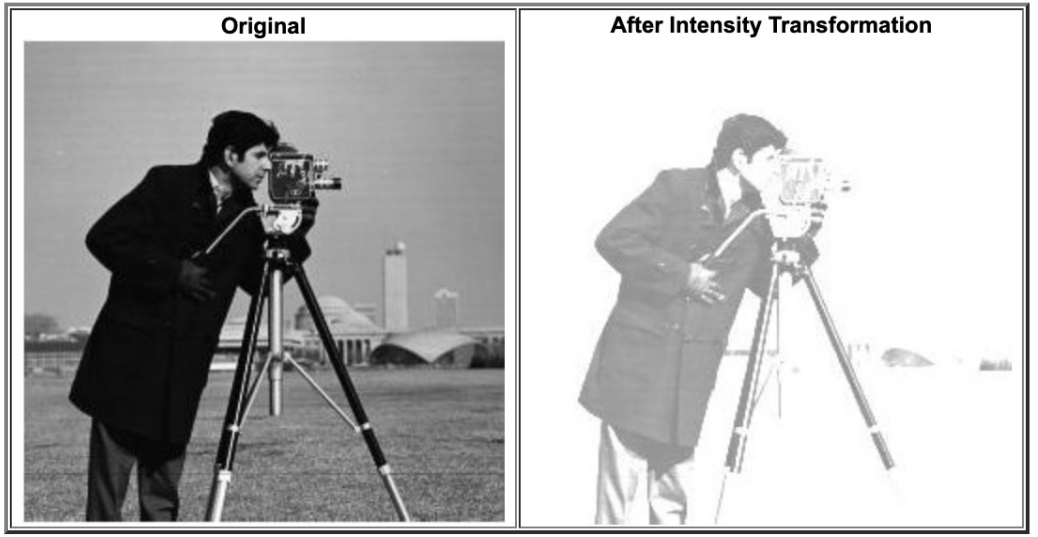
12. Shearing in both x and y direction $\begin{bmatrix} 1 & Sh_y & 0 \\ Sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Intensity transformations

Intensity transformations are used to increase contrast between certain intensity values that emphasize certain image features. For instance, you may want to reverse black and the white intensities or you may want to make the darks darker and the lights lighter. An application of intensity transformations is to increase the contrast between certain intensity values so that you can pick out things in an image.

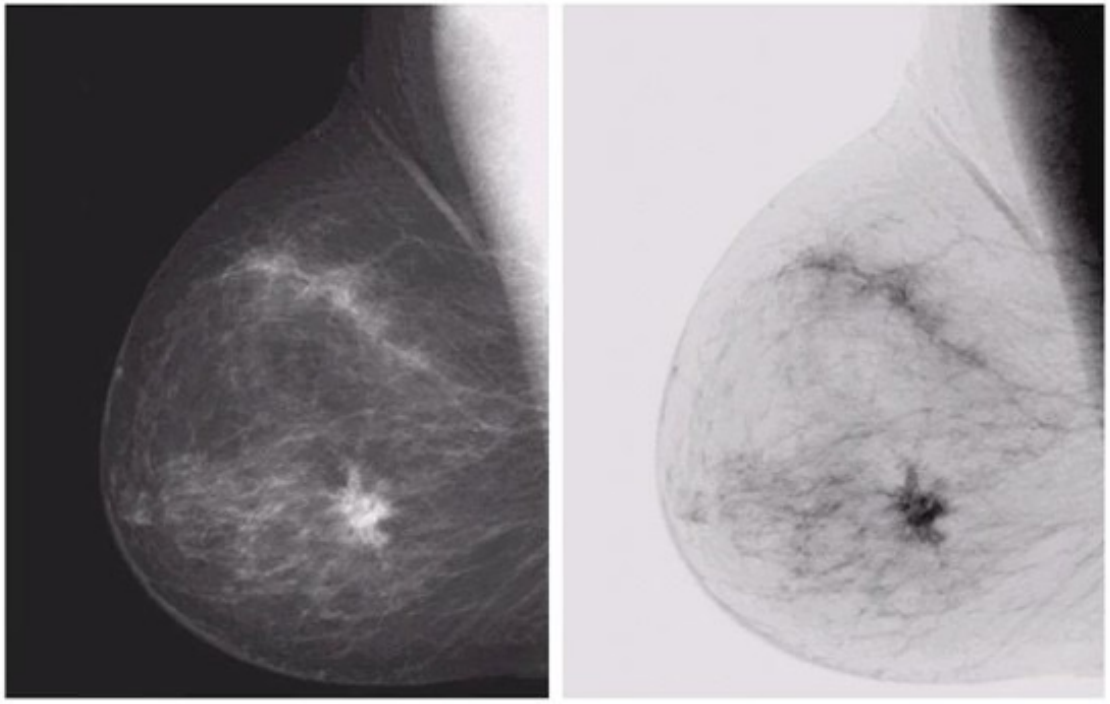
Typical intensity transformation functions include:

1. Photographic negative
2. Gamma transformation
3. Logarithmic transformations
4. Contrast-stretching transformation



Photographic negative

Assume input pixel intensities are in range [0, L], where L=255.



(a) (b)

(a) Original digital mammogram $f(x, y)$,
(b) Negative image obtained using the negative transformation

$$g(x, y) = 255 - f(x, y)$$

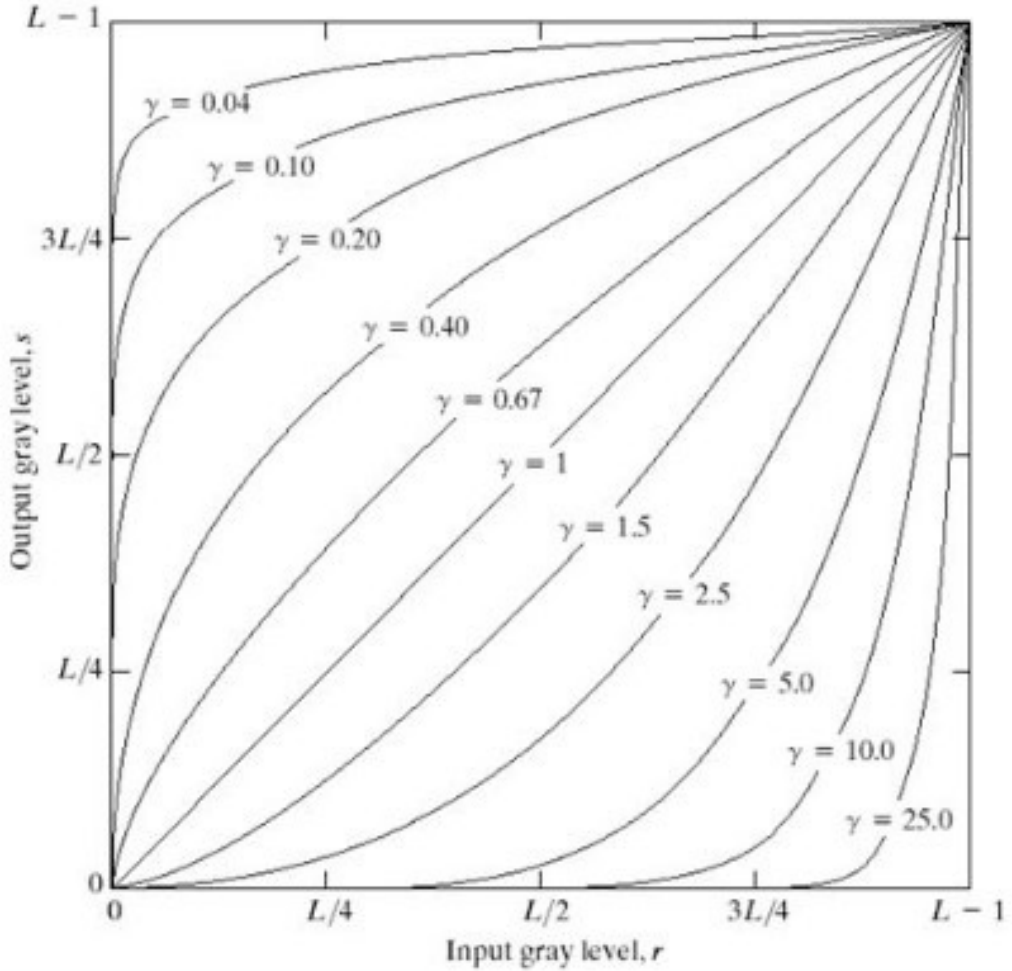
Gamma transformation/ correction

With Gamma correction

$g(x, y) = cf(x, y)^{\gamma}$, where c, γ are positive constants, we can adjust grey level components to either brighten up or darken their intensities.

γ here controls the curve. Notice that the full range of the input is mapped to the full range of the output.

What condition should γ satisfy for brightening up the image?



Gamma correction

Original

γ-corrected, $\gamma = 2.0$

Logarithmic transformations

Logarithmic transformation can be used to brighten the intensities in an image. It is used to increase the detail(contrast) of lower intensity values. They are especially useful for bringing out detail in Fourier transforms. The logarithmic transform of image $f(x, y)$ is:

$$g(x, y) = c \log(1 + f(x, y))$$

The constant c is typically used to scale the range of the log function to match the intensity range of the original image.

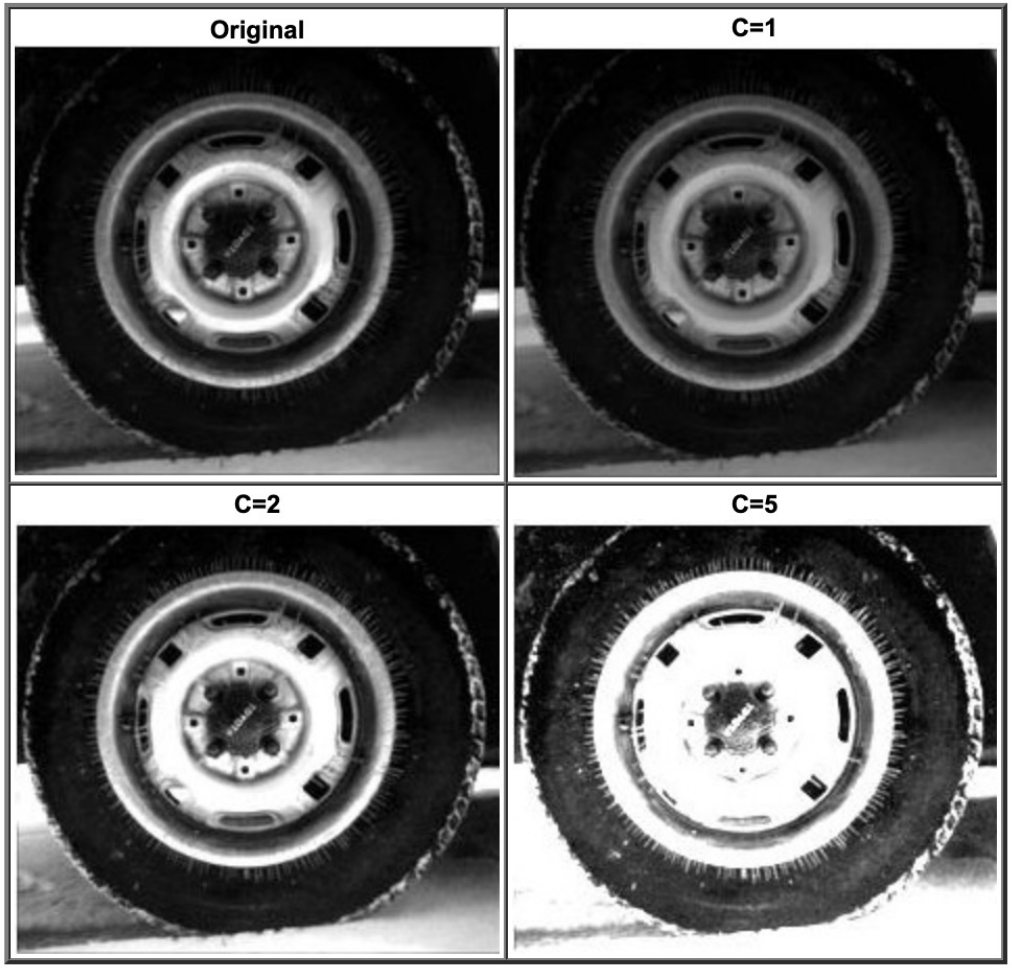
$$c = 255/\log(1 + 255)$$

It can also be used to further increase contrast. The higher the c, the brighter the image will appear.

Log transformation compresses the dynamic range of images with large variations in pixel values.

Logarithimic transformations

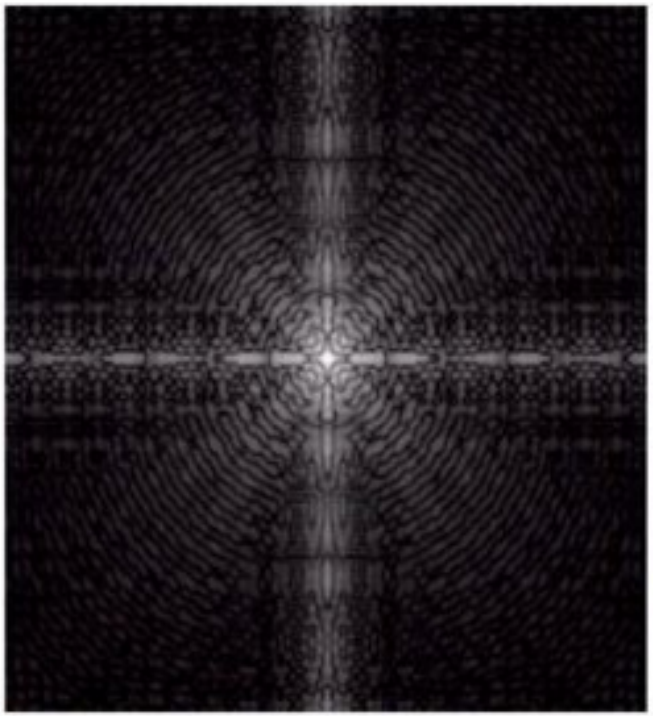
Image enhancement in spatial domain



Logarithimic transformations

Image enhancement in frequency domain

(a) Fourier spectrum



(b) Log-transformed

Contrast-stretching transformations

Contrast-stretching transformations increase the contrast between the darks and the lights. Sometimes, we want to increase the intensity around a certain grey level. As a result, dark colours become a lot darker and light colours become a lot lighter with only a few levels of grey around the level of interest.

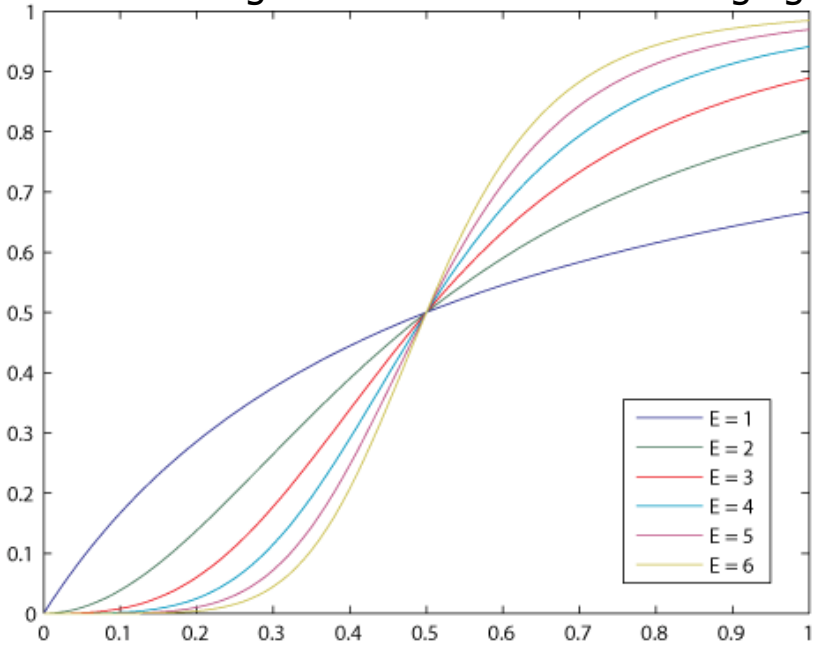
Contrast-stretching transformation can be created with the following function:

$$g(x,y) = \frac{1}{(1 + m/(f(x,y) + \epsilon))^E}$$

E controls the slope of the function and m is the mid-line where we want to switch from dark to light values. ϵ is a small constant used to prevent division by 0.

Grey levels

Contrast-stretching transformations with changing E

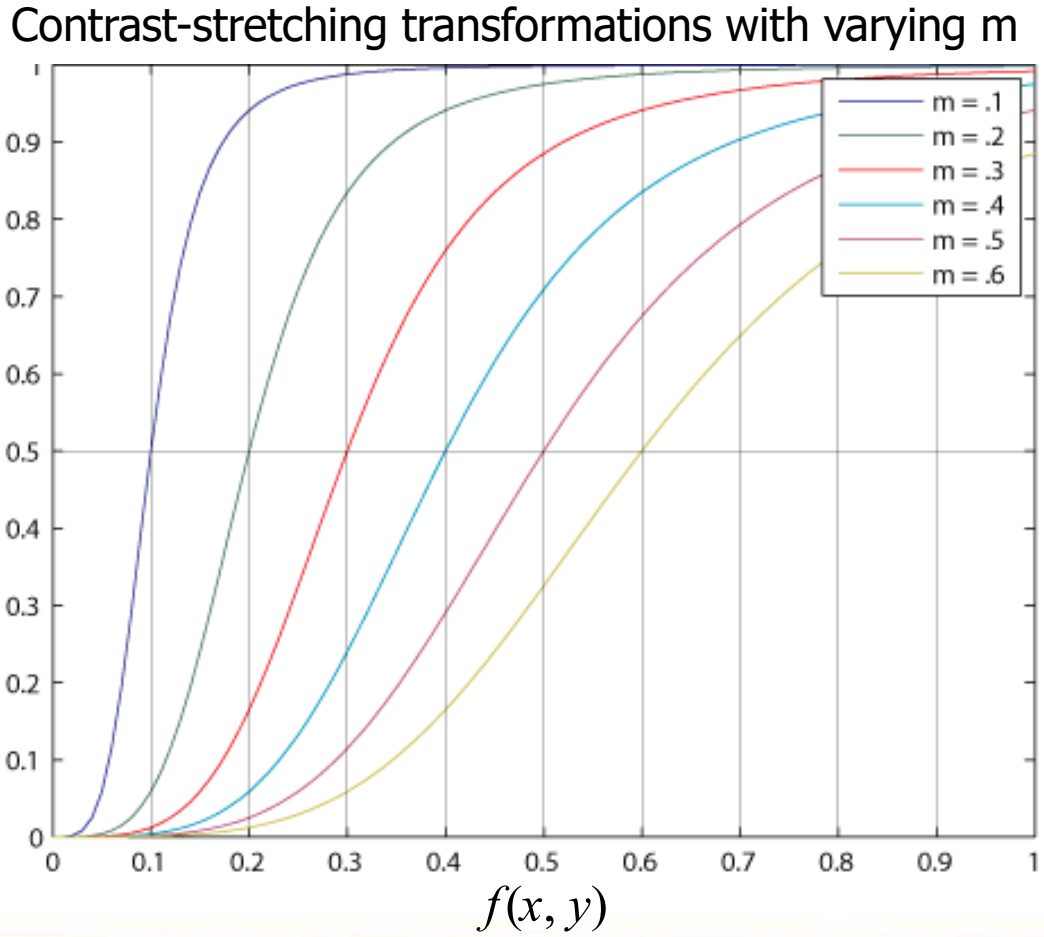


What is the midline point m in this plot?

Contrast-stretching transformations

We can also change the midline that affects the contrast curve. Fixing E=4 and varying m = 0.1,0.2,...,0.6, we have the following plot.

$$g(x,y) = \frac{1}{(1 + m/(f(x,y) + \epsilon))^E}$$



Smoothing filters

Smoothing filters are used for blurring and noise reduction. Blurring is used in preprocessing steps, specifically, for removal of small details from an image prior to object extraction and bridging of small gaps in lines and curves.

Noise reduction can be accomplished by blurring with an *average filter*. By replacing the value of every pixel in an image by the average of the grey levels in the neighbourhood defined by the filter mask, this process results in an image with reduced abrupt transitions in pixel intensities. A major use of the averaging filter is in reduction of "irrelevant" detail in an image (pixel regions that are small with respect to the size of the filter mask).

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

The diagram on the left shows an example of A 3×3 averaging filter. It yields the standard average of pixels under the mask. The constant, is a normalizing constant. For a mask of size m × n, what is the normalizing constant equal to?

The response of this mask at any point of an image (x, y) is given by:

$$R = \frac{1}{9} \sum_{i=-1}^{1} \sum_{j=-1}^{1} f(x+i, y+j)$$

Smoothing filters

Given below is another example of the smoothing filter. What is the constant multiplier in front of the mask equal to?

This mask is called *a weighted average*. The image pixels are multiplied by different coefficients, thus giving more importance to some pixels at the expense of others. The pixel at the centre of the mask is multiplied by a higher value than any other, thus giving this pixel more importance in the calculation fo the average. The other pixels are inversely weighted as a function of distance from the centre of the mask.

$$\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

The diagonal terms are further away from the centre than the orthogonal neighbours by a factor of $\sqrt{2}$ and thus are weighted less. The basic strategy behind weighing the centre point the highest and then reducing the value of the coefficients as a function of increasing distance from the origin is simply an attempt to reduce blurring in the smoothing process.

Smoothing filters

Given below is a general form of a 3×3 smoothing filter.

$$W = \begin{bmatrix} w(-1,-1) & w(-1,0) & w(-1,1) \\ w(0,-1) & w(0,0) & w(0,1) \\ w(1,-1) & w(1,0) & w(1,1) \end{bmatrix}$$

In general, a smoothing filter is a weighted averaging filter of size $m \times n$ (m and n are odd). The formula for filtering an $M \times N$ image with the weighted averaging filter is given by the expression

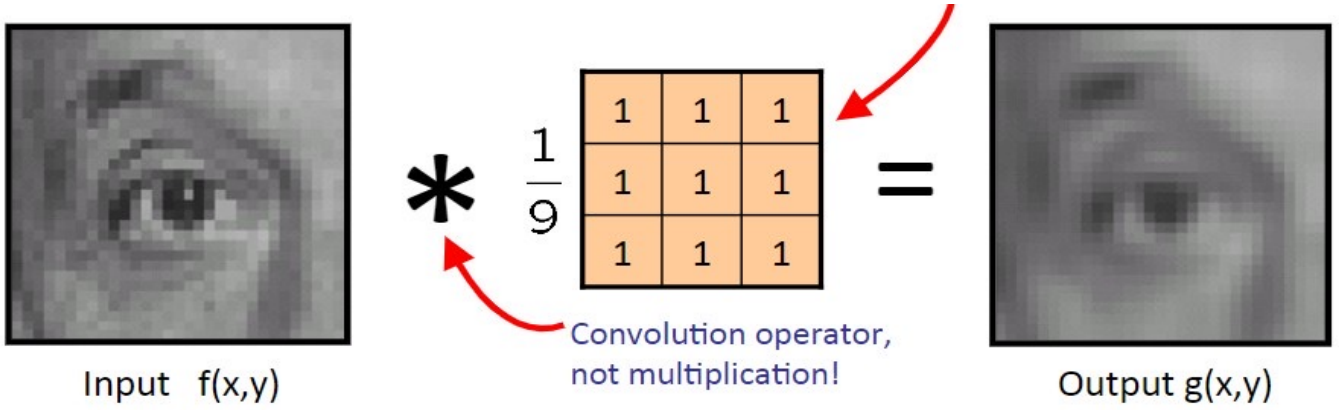
$$g(x,y) = \frac{\sum_{s=-a}^{s=a} \sum_{t=-b}^{t=b} f(x+s, y+t) w(s,t)}{\sum_{s=-a}^{s=a} \sum_{t=-b}^{t=b} w(s,t)} = f(x,y) * W. \qquad (1)$$

The complete filtered image is obtained by applying equation (1) for $x = 0,1,2,...,M\text{-}1$ and $y = 0,1,2,...,N\text{-}1$. * denotes the operation of convolution.

What is the denominator equal to? Is it a constant?

Smoothing filters

Here is the result of smoothing an image using averaging filter. Notice how filtering smoothes local sharp changes in pixel intensities of the original image.



Final Exam Review

Final exam (30 pts) will focus on the material that was covered during the course sessions:

- You need to have knowledge of the underlying concepts and mathematics of the topics covered in the classroom.
  - The final exam will NOT have any (python) coding questions
  - You CAN use your course material during the exam
- This is NOT a group exam. Each student shall only use his/her knowledge to answer the questions.
- You cannot communicate (in any form) with your classmates or other individuals to answer the questions.
- Failure to comply with GBC exam policies results in academic consequences.