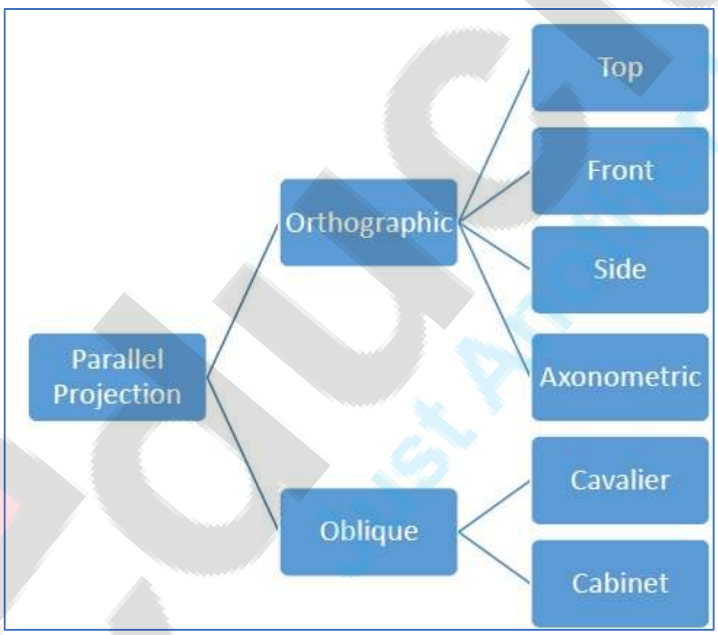
Parallel and perspective projections and their types

Parallel Projection

Parallel projection discards z-coordinate and parallel lines from each vertex on the object are extended until they intersect the view plane. In parallel projection, we specify a direction of projection instead of center of projection.

In parallel projection, the distance from the center of projection to project plane is infinite. In this type of projection, we connect the projected vertices by line segments which correspond to connections on the original object.

Parallel projections are less realistic, but they are good for exact measurements. In this type of projections, parallel lines remain parallel and angles are not preserved. Various types of parallel projections are shown in the following hierarchy.

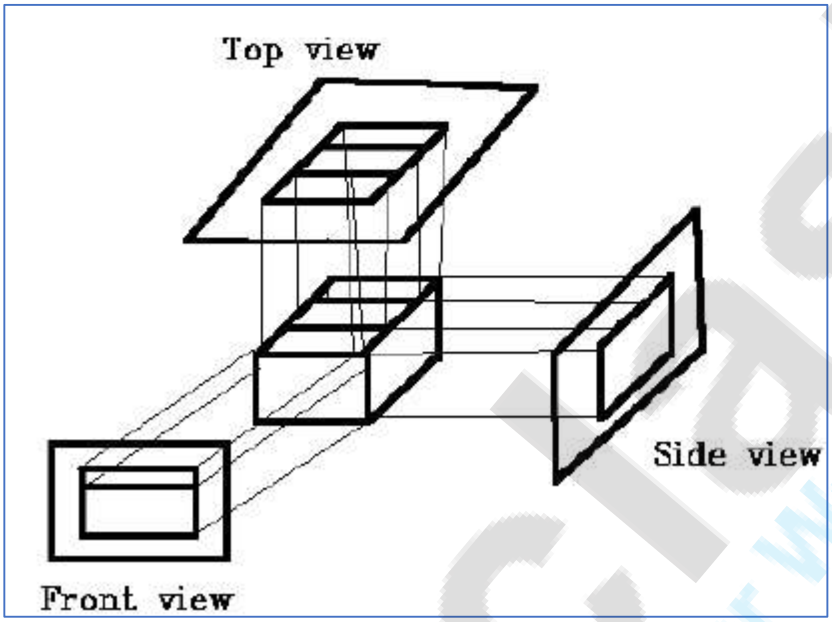


Orthographic Projection

In orthographic projection the direction of projection is normal to the projection of the plane. There are three types of orthographic projections −

- Front Projection

- Top Projection
- Side Projection

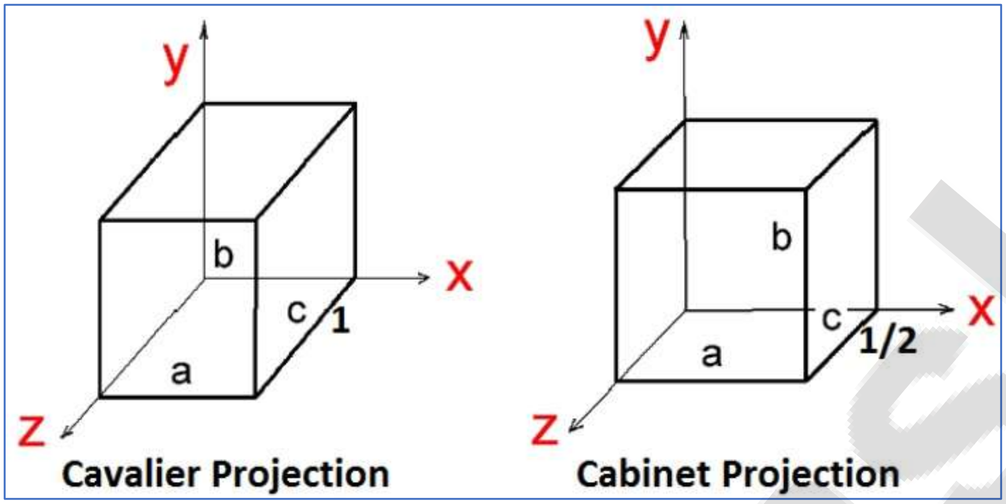


Oblique Projection

In orthographic projection, the direction of projection is not normal to the projection of plane. In oblique projection, we can view the object better than orthographic projection.

There are two types of oblique projections − **Cavalier** and **Cabinet**. The Cavalier projection makes 45° angle with the projection plane. The projection of a line perpendicular to the view plane has the same length as the line itself in Cavalier projection. In a cavalier projection, the foreshortening factors for all three principal directions are equal.

The Cabinet projection makes 63.4° angle with the projection plane. In Cabinet projection, lines perpendicular to the viewing surface are projected at ½ their actual length. Both the projections are shown in the following figure −

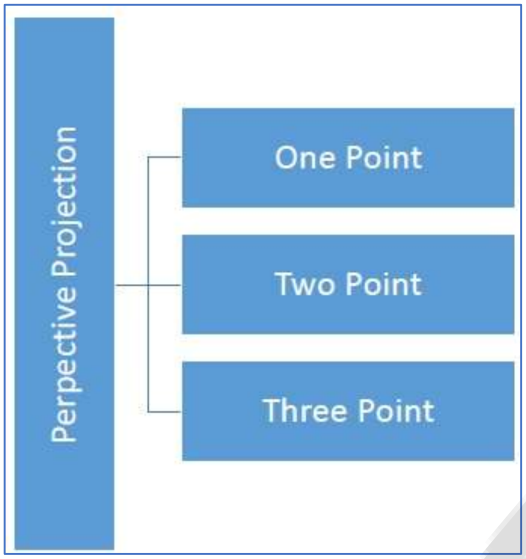


Perspective Projection

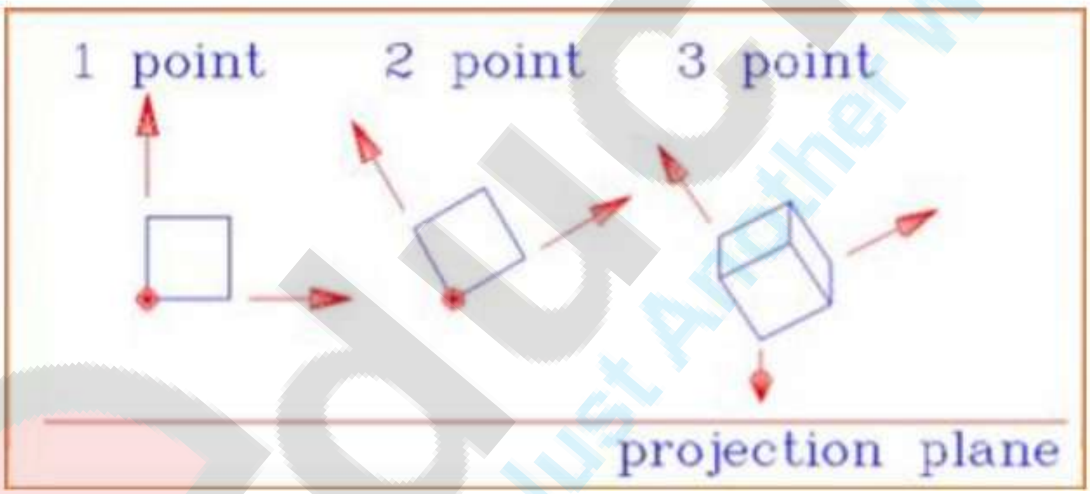
In perspective projection, the distance from the center of projection to project plane is finite and the size of the object varies inversely with distance which looks more realistic.

The distance and angles are not preserved and parallel lines do not remain parallel. Instead, they all converge at a single point called **center of projection** or **projection reference point**. There are 3 types of perspective projections which are shown in the following chart.

- **One point** perspective projection is simple to draw.
- **Two point** perspective projection gives better impression of depth.
- **Three point** perspective projection is most difficult to draw.



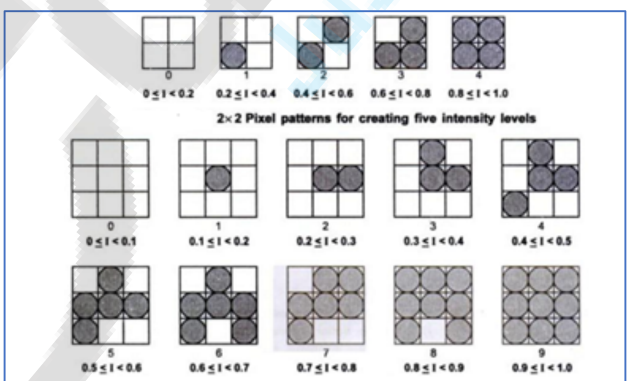
The following figure shows all the three types of perspective projection:



Halftoning and Dithering

Half toning

1. Many displays and hardcopy devices are bi-level
2. They can only produce two intensity levels.
3. In such displays or hardcopy devices we can create an apparent increase in the number of available intensity value.
4. When we view a very small area from a sufficient large viewing distance, our eyes average fine details within the small area and record only the overall intensity of the area.
5. The phenomenon of apparent increase in the number of available intensities by considering combine intensity of multiple pixels is known as half toning.
6. The half toning is commonly used in printing black and white photographs in newspapers, magazines and books.
7. The pictures produced by half toning process are called halftones.
8. In computer graphics, halftone reproductions are approximated using rectangular pixel regions, say 2$2 \; pixels \; or \; 3$3 pixels.
9. These regions are called halftone patterns or pixel patterns.
10. Figure shows the halftone pattern to create number of intensity levels.



3×3 Pixel patterns for creating ten intensity levels

Dithering

- Dithering refers to techniques for approximating halftones without reducing resolution, as pixel grid patterns do.
- The term dithering is also applied to halftone approximation method using pixel grid, and something it is used to refer to color halftone approximations only.
- Random values added to pixel intensities to break up contours are often referred as dither noise.
- Number of methods is used to generate intensity variations.
- Ordered dither methods generate intensity variations with a one-to-one mapping of points in a scene to the display pixel.
- To obtain n2n2 intensity levels, it is necessary to set up an n*n dither matrix DnDn whose elements are distinct positive integers in the range of 0 to n2−1n2−1.
- For example it is possible to generate four intensity levels with

$$D_2 = \begin{matrix} 3 & 1 \\ 0 & 2 \end{matrix}$$ and it is possible to generate nine intensity levels with

$$D_3 = \begin{matrix} 7 & 2 & 6 \\ 4 & 0 & 1 \\ 3 & 8 & 5 \end{matrix}$$

- The matrix element for D_2 and D_3 are in the same order as the pixel mask for setting up 22 *and* 33 pixel grid respectively.
- For bi-level system, we have to determine display intensity values by comparing input intensities to the matrix elements.
- Each input intensity is first scaled to the range 0< I < n2n2.
- If the intensity I is to be applied to screen position (x, y), we have to calculate numbers for the either matrix as

i=(xmodn)+1 j=(ymodn)+1i=(xmodn)+1 j=(ymodn)+1

- If I>Dn(i,j)I>Dn(i,j) the pixel at position (x, y) is turned on; otherwise the pixel is not turned on.
- Typically the number of intensity levels is taken to be a multiple of 2.

- High order dither matrices can be obtained from lower order matrices with the recurrence relation.

$$D_n = \begin{matrix} 4D_{n/2} + D_2(1,1)u_{n/2} & 4D_{n/2} + D_2(1,2)u_{n/2} \\ 4D_{n/2} + D_2(2,1)u_{n/2} & 4D_{n/2} + D_2(2,\ 2)u_{n/2} \end{matrix}$$

- Another method for mapping a picture with m*n points to a display area with m*n pixels is error diffusion.
- Here, the error between an input intensity value and the displayed pixel intensity level at a given position is dispersed, or diffused to pixel position to the right and below the current pixel position.

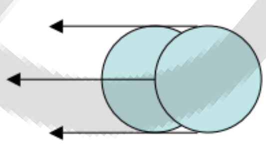
octrees, sweep and csg representation

Octrees

- Are another hierarchal structure sometimes used to represent 3D-objects.
- Often used when want to be able to draw cross sections of a figure as in medical imaging.
- Octree are essentially the extension to 3D of quadtrees which are used in 2D (and fit on Powerpoint slides better…)

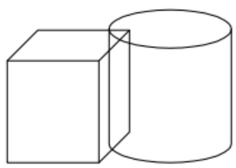
Sweep Representations

- A useful way to specify a 3D object.
- Start with a 2D object, say a closed Bezier Curve…Then translate or rotate curve slightly and create a surface



Constructive Solid Geometry Methods

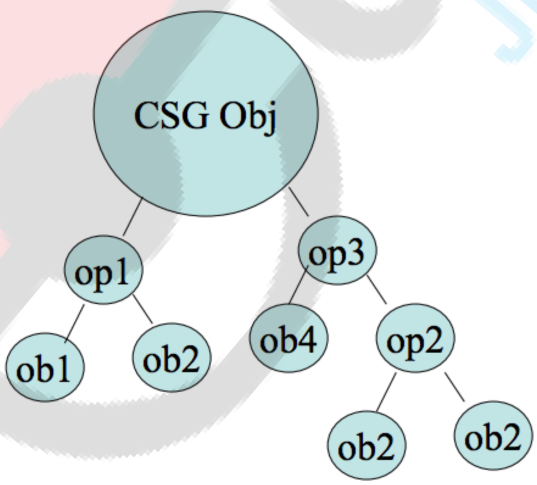
- Another way to generate 3D Objects is to have a set of primitive objects and support operations like union, intersection, set difference on them. These kind of techniques are called **constructive solid geometry methods**.



Might compute intersection of these figures

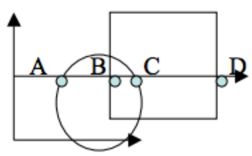
More CSG

- Objects can be viewed as binary trees:



Still More CSG

- **Ray casting** is typically used to implement CSG when objects given by boundary representations.
- We imagine shooting a rays perpendicular to the x-y plane along the z-direction. Surface intersections along each ray path are calculated and sorted by depth. Using thing and the set operations of the CSG we determine the surface limits of the object.



Operation	Surface Limit
Union	A,D
Intersection	B, C
Diff	C, D

Explain koch curve

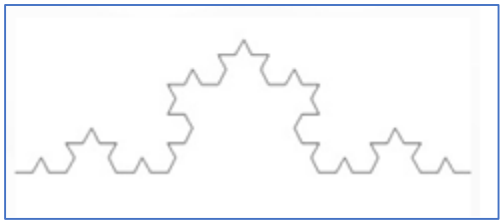
1. The Koch snowflake (also known as the Koch curve, star, or island) is a mathematical curve and one of the earliest fractal curves to have been described.
2. A Koch curve is a fractal generated by a replacement rule. This rule is, at each step, to replace the middle 131/3 of each line segment with two sides of a right triangle having sides of length equal to the replaced segment.
3. This quantity increases without bound; hen
4. ce the Koch curve has infinite length.
5. However, the curve still bounds a finite area.
6. We can prove this by noting that in each step, we add an amount of area equal to the area of all the equilateral triangles created.

Construction:

1. The Koch snowflake can be constructed by starting with an equilateral triangle, then recursively altering each line segment as follows:
   - Divide the line segment into three segments of equal length.
   - Draw an equilateral triangle that has the middle segment from step 1 as its base and points outward.

- Remove the line segment that is the base of the triangle from step 2.

- After one iteration of this process, the resulting shape is the outline of a hexagram.



1. The Koch snowflake is the limit approached as the above steps are followed over and over again.
2. The Koch curve originally described by Koch is constructed with only one of the three sides of the original triangle.
3. In other words, three Koch curves make a Koch snowflake.

Fractal and its types & how it is measured

Fractals are those concepts in mathematics which are quite interesting because it uses the concept of algebra and the concept of complex numbers. Fractals are always made up by applying some functions. Thus, Fractals are those set of numbers which represent some self-repeating properties.

There are two important types of fractal equation which are described by the two most widely used fractal sets which are:

1) The Julia Set
2) The Mandelbrot Set:

We know that in the field of complex numbers, there are real numbers along with some imaginary numbers. Any complex number is made up of a real number along with some imaginary number.
For example: (4 + i5) and (6 + i7) are some examples of complex numbers.

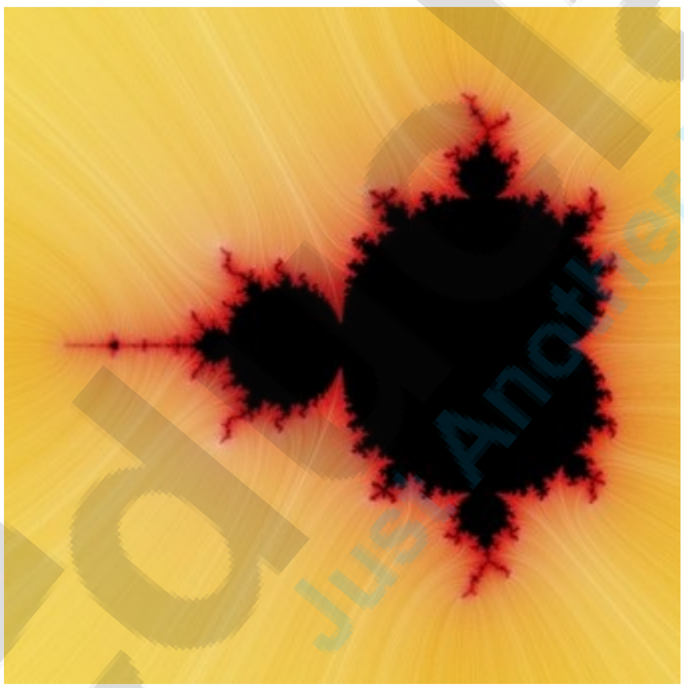
In the field of fractals, the real numbers are used for representing the x coordinate systems and the complex numbers are used for representing the y coordinate system. Therefore, if we take a pair of numbers as (5, 7), then in the field of fractals we will consider this number as (5 + i7).

For Example: consider cc = (1 + i1) and xx = (3 + i 1), then the function f(x)f(x)
= ff (3 + i1)
= $(3 + i1)^2 + (1 + i1)$
= 3 ×× 3 + 3 ×× i1 + 3 ×× i1 + i1 ×× i1 + 1 + i1
= 9 + i3 + i3 + 1(-1) + 1 + i1
= 10 + i7 - 1
= 9 + i7.
Thus, in the first loop we get the number as (9 + i7) and the same step can be repeated again to get the next second loops answer.
Now for the second loop we will consider cc = (1 + i1) and xx = (9 + i 7), then the function f(x)f(x)
= ff (9 + i7)
= $(9 + i7)^2 + (1 + i1)$
= 9 ×× 9 + 9 ×× i7 + 9 ×× i7 + i7 ×× i7 + 1 + i1
= 81 + i63 + i63 + 1(-49) + 1 + i1

= 82 + i127 - 49
= 33 + i127
The repeated answers of the above loops would show us the self similarity patterns of the fractals.

The Julia Sets:

These are those sets of fractals which are created by using the exact same formula as in case of the Mandelbrot set, except that the initial or the starting points are different every time. This implies, c is any constant with z1 as the initial or starting point on a plane. This definition implies that there can be infinite number of Julia Sets, having infinite number of values of c. In general, each and every point on a complex plane gives result to its corresponding Julia set. The following is an example of the fractals created by the Julia Set:



The Mandelbrot Set:

This set is a subset of a complex plane which always consist of those variables or parameters from which any Julia set is joined with. In simpler words, Any Mandelbrot set is a set of those values for which the set has always one and only one finite upper bound.
The relationship between the Julia set and the Mandelbrot set is that the Mandelbrot set acts as an index set for the Julia sets. All the values of c, which are inside any Mandelbrot set, we will always be joined with the Julia sets, which will

be connected. And conversely, the values of c that are outside the Mandelbrot set, we will always get unconnected sets.

Image sampling and Quantization

The output of most of the image sensors is an analog signal, and we can not apply digital processing on it because we can not store it. We can not store it because it requires infinite memory to store a signal that can have infinite values.

So we have to convert an analog signal into a digital signal.

To create an image which is digital, we need to covert continuous data into digital form. There are two steps in which it is done.

- Sampling
- Quantization

Sampling.

Sampling has already been introduced in our tutorial of introduction to signals and system. But we are going to discuss here more.

Here what we have discussed of the sampling.

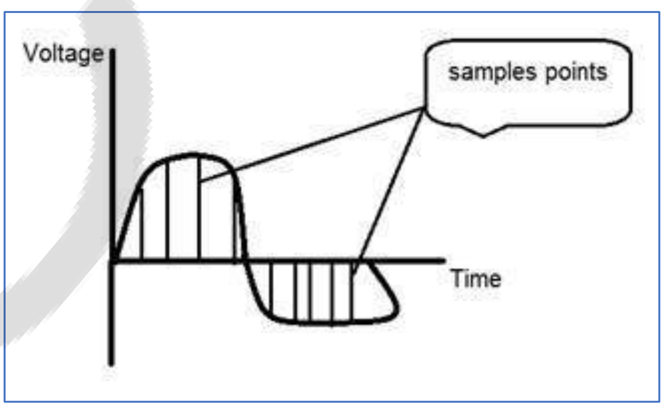
The term sampling refers to take samples

We digitize x axis in sampling

It is done on independent variable

In case of equation y = sin(x), it is done on x variable

It is further divided into two parts , up sampling and down sampling



If you will look at the above figure, you will see that there are some random variations in the signal. These variations are due to noise. In sampling we reduce this noise by taking samples. It is obvious that more samples we take, the quality of the image would be more better, the noise would be more removed and same happens vice versa.

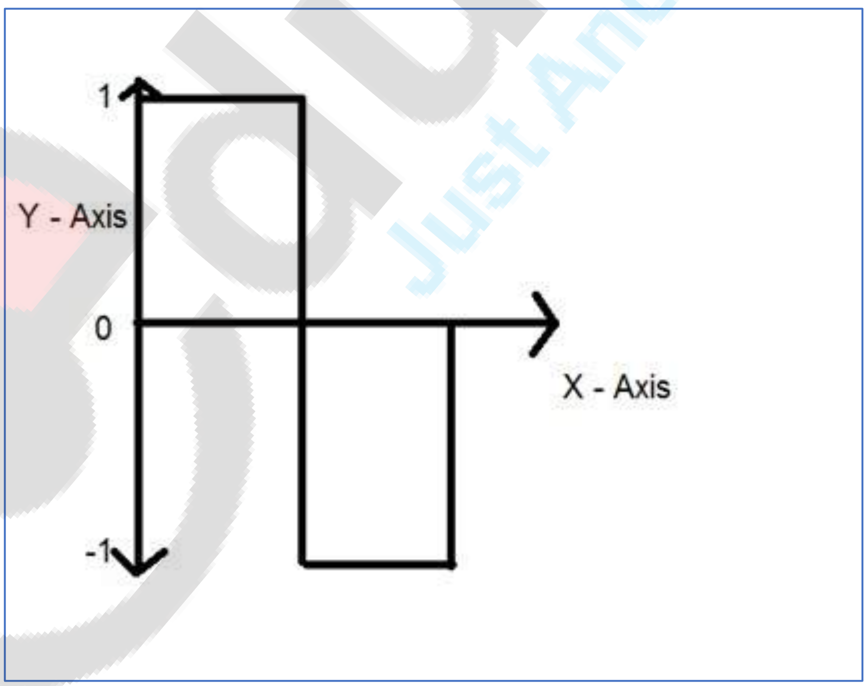
However, if you take sampling on the x axis, the signal is not converted to digital format, unless you take sampling of the y-axis too which is known as quantization. The more samples eventually means you are collecting more data, and in case of image, it means more pixels.

Quantization

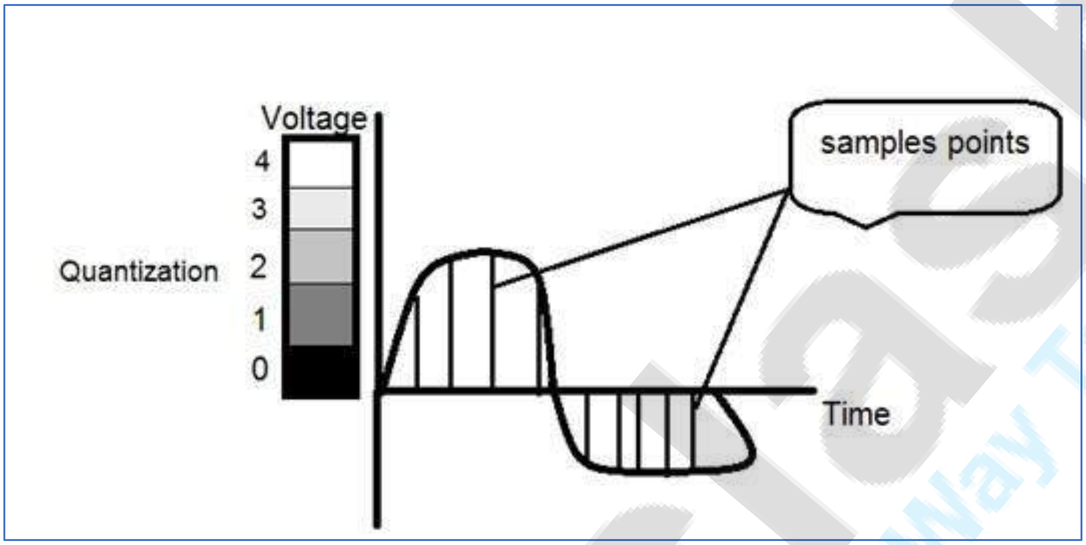
Quantization is opposite to sampling. It is done on y axis. When you are quantizing an image, you are actually dividing a signal into quanta(partitions).

On the x axis of the signal, are the co-ordinate values, and on the y axis, we have amplitudes. So digitizing the amplitudes is known as Quantization.

Here how it is done



You can see in this image, that the signal has been quantified into three different levels. That means that when we sample an image, we actually gather a lot of values, and in quantization, we set levels to these values. This can be more clear in the image below.



In the figure shown in sampling, although the samples has been taken, but they were still spanning vertically to a continuous range of gray level values. In the figure shown above, these vertically ranging values have been quantized into 5 different levels or partitions. Ranging from 0 black to 4 white. This level could vary according to the type of image you want.

Bit-plane slicing, image subtraction, image averaging, log transformation and power law transformation

BIT-PLANE SLICING

-Pixels are digital numbers, each one composed of 8 bits.
-The image is composed of 8 1-bit planes.
-Plane 0 contains the least significant bit and plane 7 contains the most significant bit.
-In terms of 8-bits bytes, plane 0 contains all lowest order bits in the bytes comprising the pixels in the image and plane 7 contains all high order bits.
-Instead of highlighting gray-level range, we could highlight the contribution made by each bit
-Only the higher order bits (top four) contain visually significant data.
-The other bit planes contribute the more subtle details.
-This method is useful and used in image compression.

Image Subtraction

¬ The difference between two images f(x,y) and h(x,y) is expressed as g(x,y)= f(x,y)-h(x,y)
¬ It is obtained by computing the difference between all pairs of corresponding pixels from f and h.
¬ The key usefulness of subtraction is the enhancement of difference between images.
¬ This concept is used in another gray scale transformation for enhancement known as bit plane slicing
¬ The higher order bit planes of an image carry a significant amount of visually relevant detail while the lower planes contribute to fine details.
¬ It we subtract the four least significant bit planes from the image
¬ The result will be nearly identical but there will be a slight drop in the overall contrast due to less variability in the graylevel values of image .
¬ The use of image subtraction is seen in medical imaging area named as mask mode radiography.
¬ The mask h (x,y) is an X-ray image of a region of a patient's body this image is captured by using as intensified TV camera located opposite to the x-ray machine then a consistent

Image Averaging

¬ Consider a noisy image g(x,y) formed by the addition of noise n(x,y) to the original image f(x,y) g(x,y) = f(x,y) + n(x,y)
¬ Assuming that at every point of coordinate (x,y) the noise is uncorrelated and has zero average value
¬ The objective of image averaging is to reduce the noise content by adding a set of noise images, {gi(x,y)}
¬ If in image formed by image averaging K different noisy images ( ) ( ) { } ( ) ( ) 1 1 , , , , K i i g x y g x y K E g x y f x y = = = ∑
¬ As k increases the variability (noise) of the pixel value at each location (x,y) decreases.
¬ E{g(x,y)} = f(x,y) means that g(x,y) approaches f(x,y) as the number of noisy image used in the averaging processes increases.
¬ Image averaging is important in various applications such as in the field of astronomy where the images are low light levels.

Log transformation

The log transformations can be defined by this formula

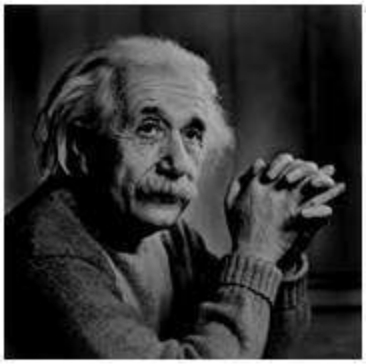
s = c log(r + 1).

Where s and r are the pixel values of the output and the input image and c is a constant. The value 1 is added to each of the pixel value of the input image because if there is a pixel intensity of 0 in the image, then log (0) is equal to infinity. So 1 is added, to make the minimum value at least 1.

During log transformation, the dark pixels in an image are expanded as compare to the higher pixel values. The higher pixel values are kind of compressed in log transformation. This result in following image enhancement.

The value of c in the log transform adjust the kind of enhancement you are looking for.

Input Image



Log Tranform Image

The inverse log transform is opposite to log transform.

Power – Law transformations

There are further two transformation is power law transformations, that include nth power and nth root transformation. These transformations can be given by the expression:

$s=cr^\gamma$

This symbol γ is called gamma, due to which this transformation is also known as gamma transformation.

Variation in the value of γ varies the enhancement of the images. Different display devices / monitors have their own gamma correction, that's why they display their image at different intensity.

This type of transformation is used for enhancing images for different type of display devices. The gamma of different display devices is different. For example Gamma of CRT lies in between of 1.8 to 2.5, that means the image displayed on CRT is dark.

Correcting gamma.

s=cr^γ

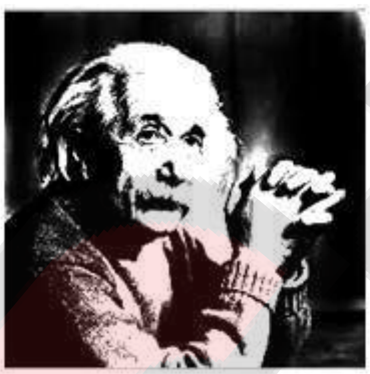
s=cr^(1/2.5)

The same image but with different gamma values has been shown here.

For example

Gamma = 10

Gamma = 8



Gamma = 6

Components of IP system

Components of the System

- **Image Sensor:** With reference to sensing, two elements are required to acquire digital images: **a sensor and a digitizer**. The sensor that is sensitive to the energy radiated by the object we wish to image. The second, called a digitizer, is a device for converting the output of the physical sensing device into digital form. For example, in a digital video camera, the sensors produce an electrical output proportional to light intensity. The digitizer converts these outputs to digital data.
- **Specialized image processing hardware:** usually consists of the digitizer just mentioned, plus hardware that performs other primitive operations, such as an arithmetic logic unit (ALU). ALU performs arithmetic and logical operations in parallel on entire images. ALU is used is in averaging images as quickly as they are digitized, for the purpose of noise reduction. This type of hardware sometimes is called a front-end subsystem, and its most distinguishing characteristic is speed in which they process image.

Components of the System

- **Computer:** in an image processing system is a general-purpose computer and can range from a PC to a supercomputer. In dedicated applications, some times specially designed computers are used to achieve a required level of performance.
- **Software:** for image processing consists of specialized modules that perform specific tasks
- **Mass storage:** capability is a must in image processing applications. Digital storage for image processing applications falls into three principal categories: (1) short-term storage for use during processing, (2) on-line storage for relatively fast recall, and (3) archival storage, characterized by infrequent access.
- **Image display:** it displays images.
- **Hardcopy devices:** used for recording images include laser printers, film cameras, heat-sensitive devices, inkjet units, and digital units, such as optical and CD-ROM disks.

29