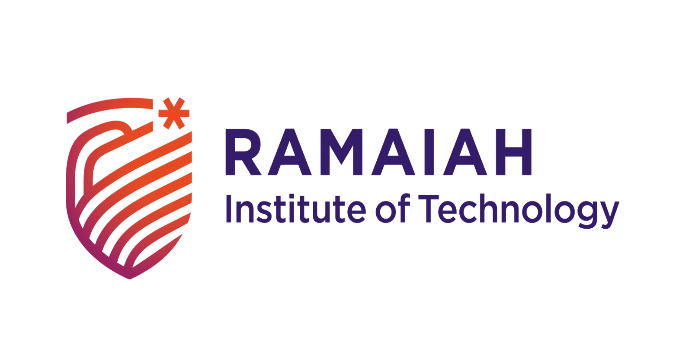
**M S RAMAIAH INSTITUTE OF TECHNOLOGY**

**(Autonomous Institute Affiliated to VTU)**

**Department of Information Science and Engineering**

****

**UNIT-4** Notes

**COMPUTER GRAPHICS**

Submitted by

Tejaswini K 1MS18IS415

Yaswanth V 1MS18IS422

Apurva Raj 1MS17IS158

Mahesh G 1MS17IS060

Mayur S.H. 1MS17IS063

**Definition of Solid Modeling:**

A solid model of an object is a more complete representation than its surface (wireframe) model. It provides more topological information in addition to the geometrical information which helps to represent the solid unambiguously.

**Representing Solids:**

The representation of solid models uses the fundamental idea that a physical object divides the 3-D Euclidean space into two regions, one exterior and one interior, separated by the boundary of the solid. Solid models are:

* bounded
* Homogeneously three dimensional
* Finite

There are six common representations in solid modeling:

* 1. **Spatial Enumeration:** In this simplest form of 3D volumetric raster model, a section of 3D space is described by a matrix of evenly spaced cubic volume elements called voxels.
  2. **Cell Decomposition:** This is a hierarchical adaptation of spatial enumeration. 3D space is sub-divided into cells. Cells could be of different sizes. These simple cells are glued together to describe a solid object.
  3. **Boundary Representation:** The solid is represented by its boundary which consists of a set of faces, a set of edges and a set of vertices as well as their topological relations.
  4. **Sweep Methods:** In this technique a planar shape is moved along a curve. Translational sweep can be used to create prismatic objects and rotational sweep could be used for axisymmetric components.
  5. **Primitive Instancing:** This modeling scheme provides a set of possible object shapes which are described by a set of parameters. Instances of object shape can be created by varying these parameters.
  6. **Constructive Solid Geometry (CSG):** Primitive instances are combined using Boolean set operations to create complex objects.

In most of the modeling packages, the approach used for modeling uses any one of the following three techniques:

1. Constructive solid geometry (CSG or C-Rep)
2. Boundary representation (B-Rep)
3. Hybrid method which is a combination of B-Rep and CSG

**Advantages of solid Modeling:**

1. Unlike wireframes and surface representations which contain only geometrical data
2. The solid model uses topological information in addition to the geometrical information to represent the object unambiguously and completely.

**iii.** Solid model results in accurate design, helps to further the goal of CAD/ CAM like CIM, Flexible manufacturing leading to better automation of the manufacturing process.

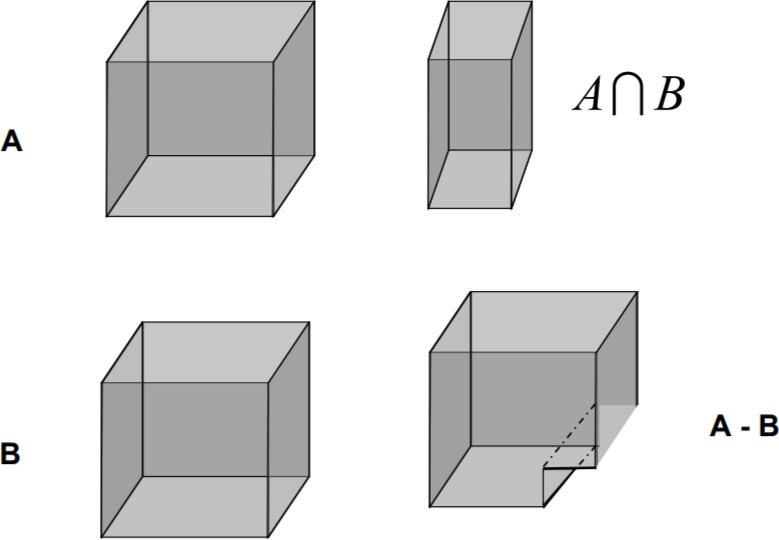
**Disadvantages of Solid Modeling:**

1. Ambiguity
2. Subjective human interpretation
3. Complex objects with many edges become confusing
4. Lengthy and verbose to define
5. Not possible to calculate Volume and Mass properties, NC tool path, cross sectioning etc.

**Regularized Boolean Set Operations:**

A Way to show physical objects via mathematical operation in solid modeling

Boolean intersections of, say cubes, may produce solids, planes, lines, points and null object. Two examples:



We certainly expect that the union, intersection and difference of two solids is a solid. Unfortunately, in many cases this is not always true. In the following figure, two cubes touch each other and their intersection is a rectangle shown on the right. A rectangle is not a three-dimensional object and hence not a solid

To eliminate these lower dimensional branches, the three set operations are regularized as follows. The idea of regulating is very simple.

* + Compute the result as usual and lower dimensional components many be generated.
  + Compute the interior of the result. This step removes all lower dimensional components. The result is a solid without its boundary.
  + Compute the closure of the result obtained in the above step. This adds the boundary back.

Let +, ^ and - be the regularized set union, intersection and difference operators. Let A and B be two solids. Then, A+B, A ^ B and A - B can be defined mathematically based on the above description:

A + B = closure(int(the set union of A and B)

A ^ B = closure(int(the set intersection of A and B) A - B = closure(int(the set difference of A and B)

Based on this definition, the intersection of the two cubes shown at the beginning of this page is empty. As mentioned earlier, the set intersection of these two cubes is a rectangle, which is a two dimensional object and has no interior. Hence, after taking interior (i.e., int()), we get an empty set, whose closure is also empty. Consequently, the intersection is empty.

**i.** Explain the role of Regularized Boolean set operations and Boolean intersections explain from the context of solid modeling.

Normal Boolean set operations are union , intersection and subtraction Applied to volume these operations not closed, because they can yield points, lines or planes Regularized Boolean operations always yields volumes , so the intersection of 2 cubes sharing an edge is null.it would be a line in a normal Boolean operation A regularized Boolean set-operation op∗ can be obtained by first taking the interior of the resultant point set of an *ordinary* Boolean set-operation (P op Q) and then by taking the closure . That is, P op∗ Q=closure(interior(P op Q)). Regularized Boolean set-operations appear in Constructive Solid Geometry (CSG), because regular sets are closed under regularized Boolean set-operations, and because regularization eliminates lower dimensional features, namely isolated vertices and antennas, thus simplifying and restricting the representation to physically meaningful solids. Our package provides regularized operations on polygons and general polygons, where the edges of a general polygon may be general x-monotone curves, rather than being simple line segments.

**Sweep Representation:**

* Sweep representations are useful for constructing 3-D objects that, rotational, possess translational, or other symmetries. We can represent a sweep that moves the shape through a region of space and such objects by specifying a 2-D shape.
* A set of 2-D primitives, such as circles and rectangles, can be provided for sweep representations as menu options. Other methods for obtaining 2-D figures include closed spline constructions and cross- sectional slices of solid objects.

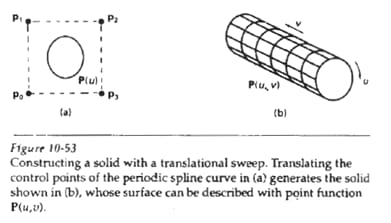
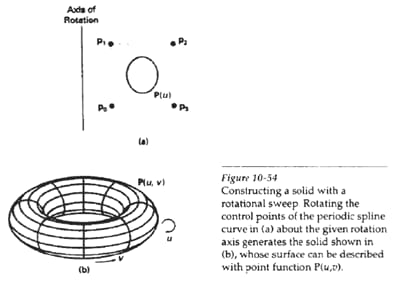


Figure 10-53 illustrates a translational sweep.

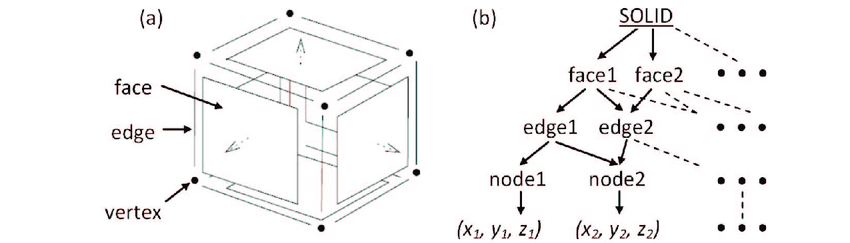
* We then perform a translational sweep by moving the point’s p through p3 a collection distance on a line path perpendicular to the plane of cross section...
* At intervals on this we have a tendency to replicate the cross-sectional shape and draw a collection of connecting lines within the direction of the sweep to get the wireframe in 10-53(b).
* An example of object design employing a rotational sweep is given in fig 10-54.This time; the periodic spline cross section is revolved concerning the axis of rotation laid out in the plane of the cross section to provide the wireframe illustration in fig 10-54(b).
* Any axis may be chosen for a move sweep. If we have a tendency to use a rotational axis perpendicular to the plane of the spline cross section in fig. 10-54, we have a tendency to generate a 2-D form. However if the cross section has depth, then we are using one 3-D object to generate another.



• In general, we are able to specify sweep constructions of any path. For motion sweeps, we are able to move on a circular path through any angle from zero to 360’.  
• For noncircular paths, we are able to specify the curve operate describing the trail and also the distance of the trail.  
• In addition, we are able to vary the form or size of the cross-section on the sweep path. Or we have a tendency to change the orientation of the cross section relative to the sweep path as we have a tendency to move a region of space.

**Boundary Representation:**

* Boundary representation is made on the idea that an object is boxed in by a collection of faces that are closed surfaces. Face is bounded by edges and edges are bounded by vertices.
* The entities which constitute a B-rep model are: Geometric entities, Topological entities Point Vertex Curve, line Edge Surface Face.
* A model is a 3-D illustration of an object. It is a correct description of an object which has the interior structure and also the external surfaces of the object. An observer will be able to identify the information like mass, internal cross sections, and interferences.
* Solid models disagree from wire frame and surface models within the reasonably geometric data they supply. Wire frame models show the edge geometry of an object and have no data regarding the interior of the structure. Surface models give surface data, however they need no data regarding the interior.
* Models give complete geometric descriptions of objects. Engineers use solid models in numerous ways for the design process. They are able to modify a design as they develop it. Since computer-based solid models are a lot a solid model is a 3-D illustration of an object. It is a correct geometric description which includes the interior structure and the exterior surfaces of the object.



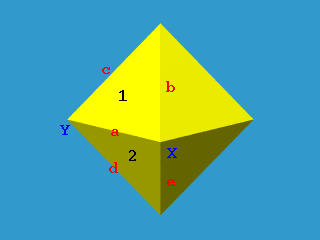
* By using solid modelling techniques a design engineer will be able to modify a design many times while optimizing geometry. This suggests that designers can be completing more designs in less time than by utilizing traditional design methods or 2-D CAD drafting tools. Solid models could be used for quick and reliable design analysis.
* Solid models leaving geometric information provide important data such as volume, mass, mass properties and centre of gravity. Finally designers will generate elaborated production drawings directly from the solid model. This capability will increases design productivity significantly.
* Another necessary feature of solid modelling is associatively.  Elaborated drawings are connected to solid model through the associatively feature. . This can be a robust operate - as an engineer modifies the design which gets updated automatically. In bidirectional associatively, any modifications made to geometry in the drawing are mirrored within the model.
* In additional advanced design and production environments, solid models are used for rapid prototyping and automated manufacturing applications. Like how they are used in Tesla Automated machines.

What is it used for?

* For one thing - collision detection. Two objects most definitely don’t collide if their bounding volumes don’t collide. you'll be able to prove this by thinking how, during this case, neither vertex of the boundary of object, it can be found in the bounding volume of object B, so no vertex of object A is be found within the bounding volume of object B, so no vertex of object A is found within the volume of object B, since bounding volumes strictly include entire volumes of objects, together withal vertices.
* Calculating collisions between 2 boxes is so simpler than calculating collisions between 2 random objects, defined by hundreds, or thousands, of vertices, thus this provides you a large improvement of not having to even have confidence trying to find precise collision points unless the bounding boxes intersect.

**Winged Edge Data Structure**

* It is quite totally different from that of a wireframe model, as a result the winged-edge data structure uses edges to keep track on everything. In what follows, we will consider that there are no holes in each face and later extend it to cope with holes. Moreover, we will consider edges and faces are line segments and polygons. Topologically, one can always stretch curvilinear edges and faces so that they become flat while not changing the relationship among them.



* The object on top shows a polyhedron with vertices, edges and faces indicated with upper cases, lower cases and digits, respectively. As we see at edge a = XY. This edge has 2 vertices X and Y, and 2 faces 1 and 2. A face is surrounded by edges. For instance, face 1 has its edges a, c and b, and face 2 has its edges a, e and d. Note that the ordering is clockwise viewed from outside of the solid. If the direction is from X to Y, face 1 and 2 are on the right and left side of edge a, respectively. To capture the ordering of edges correctly, we need 4 additional information. Since edge is traversed once when traversing face 1 and traversed a second time when traversing face 2, it is used twice in different directions. For example, when traversing the edges of face 1, the predecessor and successor of edge a are edge b and edge c, and when traversing the edges of face 2, the predecessor and successor of edge a are edge d and edge e. Note that although there are four edges incident to vertex X, only three of them are used when finding faces incident to edge a.
* Therefore, for every edge, the subsequent information are important:

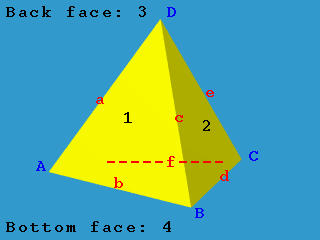
1. vertices of this edge,
2. its left and right faces,
3. the predecessor and successor of this edge when traversing its left face, and
4. The predecessor and successor of this edge when traversing its right face.

**The Edge Table**

* Each entry inside the edge table contains that data mentioned earlier: edge name, start vertex and end vertex, left face and right face, the predecessor and successor edges when traversing its left face, and the predecessor and successor edges when traversing its right face. Note that clockwise ordering (viewing from outside of the polyhedron) is for traverse. Note that the direction of edge A is from X to Y. If the direction is modified to from Y to X, all entries however the first one inside the following table should be changed consequently.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| https://pages.mtu.edu/~shene/COURSES/cs3621/NOTES/model/wing-0.jpg | |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Edge | Vertices | | Faces | | Left Traverse | | Right Traverse | | | Name | Start | End | Left | Right | Pred | Succ | Pred | Succ | | a | X | Y | 1 | 2 | b | d | e | c | |

The one in the above table shows the information for the entry of edge a. The 4 edges b, c, d and e are the wings of edge and hence edge is "winged."



The on top of could be a tetrahedron with 4 vertices A, B, C and D, 6 edges a, b, c, d, e and f, and 4 faces 1, 2, 3 (back) and 4 (bottom). Its edge table is the following. Using the above diagram we will verify this table.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Edge | Vertices | | Faces | | Left Traverse | | Right Traverse | |
| Name | Start | End | Left | Right | Pred | Succ | Pred | Succ |
| a | A | D | 3 | 1 | e | f | b | c |
| b | A | B | 1 | 4 | c | a | f | d |
| c | B | D | 1 | 2 | a | b | d | e |
| d | B | C | 2 | 4 | e | c | b | f |
| e | C | D | 2 | 3 | c | d | f | a |
| f | A | C | 4 | 3 | d | b | a | e |

Other Tables

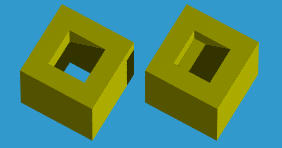
The winged-edge data structure requires 2 additional tables, the vertex table and the face table. These 2 are very simple. The vertex table has 1 entry for each vertex which contains an edge that is incident to this vertex. The face table has 1 entry for each face which contains an edge that is one of this face's boundary edges. Hence, we have the following table. Note that since there are multiple options of edges, you may have different tables:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Vertex Name | Incident Edge | | A | a | | B | b | | C | d | | D | e | | |  |  | | --- | --- | | Face Name | Incident Edge | | 1 | a | | 2 | c | | 3 | a | | 4 | b | |

With this data structure, one can simply answer the question: which vertices, edges, faces are adjacent to every face, edge, or vertex. The 9 of them adjacency relations. For example, is vertex X adjacent to face 5? Are faces 3 and 5 adjacent to each other? The winged-edge data structure can answer these queries very efficiently and some of them may even be answered in constant time. But, it should take longer time to answer other adjacency queries. Also that once the numbers of vertices, edges and faces are known, the size of all 3 tables are fixed and will not change.

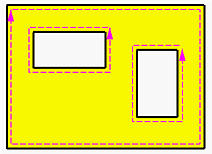
**What If Faces Have Holes?**

If some faces of a solid have holes, the above form of winged-edge data structure does not work. These holes may penetrate the solid (the left box below) or just like a pothole (the right box below).

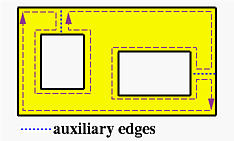


There are two ways for resolving this problem:

* For a face with inner loops, the outer boundary is ordered clockwise, while its inner loops, if any, are ordered counter clockwise.



* Another easy step is adding an *auxiliary* edge between each inner loop and the outer loop as shown below. Auxiliary edge will have the same face for its left and right faces. In this way, a face with holes becomes a single loop which can be represented with the winged-edge data structure. When traversing a loop, auxiliary edges can be identified easily since its left and right faces are the same.



SPACE PARTITIONING REPRESENTATION and OCTREES

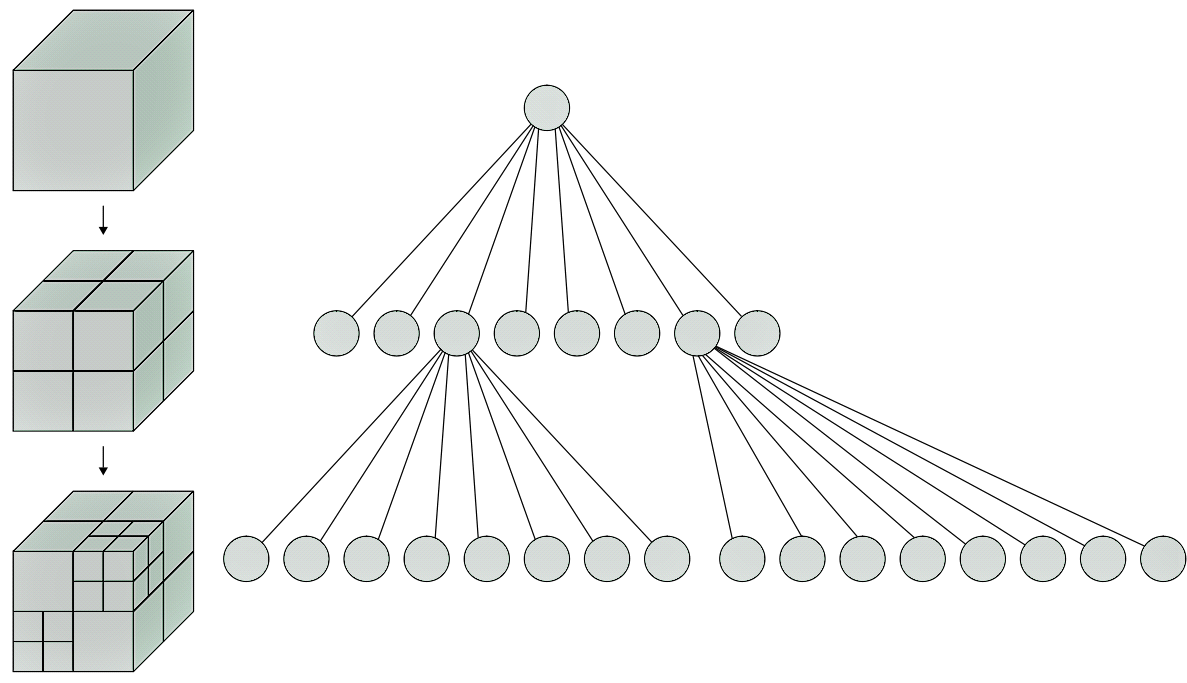
1. Short notes on spatial partitioning method.

* The process of dividing a space(usually Euclidean space) into two or more disjoint subsets.
* Divides a space into non-overlapping regions.
* A tree data structure is used to divide up the world in hierarchical manner.

The three main types are

1. Octrees:- An octrees is a tree in which each nodes has at most 8 children. Each node of the tree corresponds to a cubical region. If a node has children think of its region being chopped into 8 equal sub regions. A Child node corresponds to these smaller sub regions of their parent's region. Subdivide as little or as much is necessary. Each internal node has exactly 8 children.
2. BSP trees: - A BSP tree is a binary tree .nodes can have 0, 1 or 2 children .Tree gives a rendering order, correctly traversing this trees enumerates object from back to front. Tree split 3D world with planes.
3. Quadtees: - A quadtree is a tree in which each node has at most 4 children. If a node has children think of its region being chopped into 4 equal sub regions. Child nodes correspond to these smaller sub regions of their parent’s region. Each internal node has exactly 4 children.

2. Short notes on octrees.

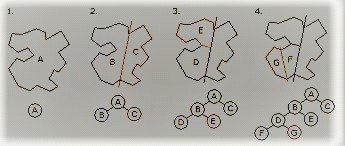


An octrees are often used in 3D graphics and 3D game engine. An octrees is a tree data structure in which each internal node has exactly eight children .Octrees are most often used to partition a three-dimensional space by recursively subdividing it into eight octant.

A single cube is placed around the entire world. If the cube contains too many triangles, then the cubes sub-divided into 8 smaller axes aligned cubes.

Recues until we hit the stopping condition.

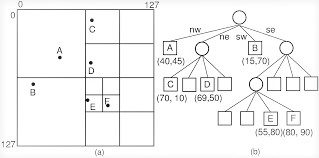
3. Short notes on Binary space-partitioning tree.



Binary tree structure that recursively partitions space into pair of subspaces with respect to dividing planes of arbitrary position and orientation. If the spaces being portioned is n-dimensional, the dividing planes are (n-1)-dimensional hyper planes.

The two partitions into which the space is divided into are referred to as the positive half spaces lies in front of the dividing hyper planes and the negative half spaces behind it.

4. Short notes on quadtrees.



It is a tree data structure in which each internal node has up to four children. Every node in the quadtree corresponds to a square. If a node V has a children, and then their corresponding squares are the four quadrants of the square of V. The leaves of quadtree form a quadtree subdivision o f square of root the children of node are labeled NE,NW,SW and SE to indicate to which quadrant they corresponds.

5. Difference between quadtrees and octrees.

The differences are:-

1. Quadtrees represents 2D planes where as octrees represents 3D cubical volumes.
2. Each internal node of quadtree has four children where as each internal node of octrees has eight children.
3. Each child in quadtree represents a quadrant of a sub-planes where as each child of octrees represents a 3D octant of a sub-volume.

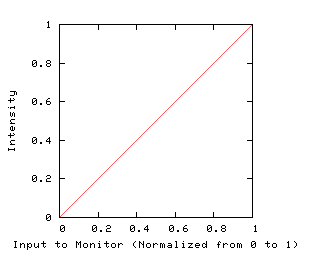
Achromatic & colored light: Gamma correction, Dither matrix, Chromatic Color, CIE Chromaticity.

Definition of Achromatic Light:

* Achromatic means that “light without color”, essentially Black-and-White.
* Black is the absence of incident light, not a color. Object that seem black (such as carbon soot) absorb incident light of all visible frequencies.
* Light combining intensity of all frequencies is termed white light. It preserves the color of objects.
* Examples of achromatic light are Radio waves, Micro waves, Infrared, Ultraviolet, X-rays, Gamma Rays.

Gamma Correction:

* Due to dis-continuity & color changing, we do not get an acceptable look of the initial image displayed on a monitor.
* Phosphors of monitors don’t react linearly with the intensity of the electron ray. If they did, a linear input ramp would lead to linear light intensity:



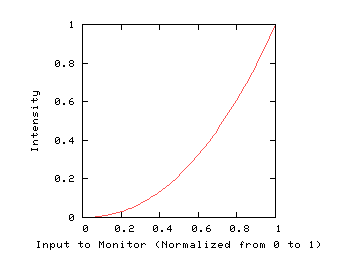
* Mapping the intensity levels in [0,1] linearly to 256 levels of brightness can turn out a ramp that seems non-linear, cause the eye is sensitive to ratios of intensities, not to absolute intensity values.
* Other drawbacks related to the show of calculated intensities are the nonlinearity of display devices. Illuminations models turn out a linear vary of intensities.
* So, Gamma correction is needed to get rid of the discontinuity.
* To correct for monitor nonlinearities, graphics systems use a video lookup table that adjusts the linear pixel values. The monitor response curve is delineated by the exponential function.

I=aVγ

* Parameter I is the displayed intensity, and parameter V is the input voltage. Values for parameters a and γ rely upon the characteristics of the monitor employed in the graphics system. Thus, if we want like to show a selected intensity value I, the proper voltage worth to supply this intensity is

V= (I/a) 1/γ

* A video lookup correction curve for mapping pixel intensities to electron-gun voltages using gamma correction with γ = 2.2. Values for both pixel intensity and monitor voltages are normalized on the interval zero to one.



* This calculation is said to be as gamma correction of intensity. Monitor gamma values are generally between 2.0 and 3.0. The National Television System Committee (NTSC) signal standard is γ= 2.2. Figure shows a gamma correction curve using the NTSC gamma value. Equation is employed to line up the video/picture lookup table that converts integer pixel values within the image file to values that controls the electron-gun voltages.

Dither Matrix:

**Dithering**

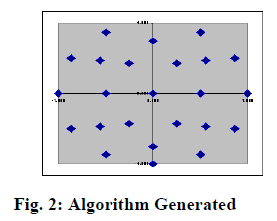
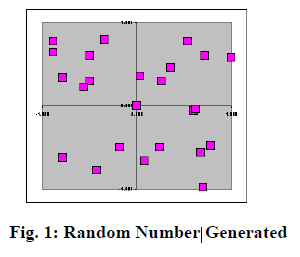
* A technique employed in quantization processes such as graphics and audio to cut back or take away the correlation between noise and signal. Dithering is employed in computer graphics to make additional colors and shades from an existing palette by interspersing pixels of various colors.
* On a monochrome show, areas of Grey are created by varying the proportion of black and white pixels. In color displays and printers, colors and textures are created by varying the proportions of existing colors. The various colors will either be distributed at random or regular manner.
* The higher the resolution of the display, the smoother the dithered color can seem to appear to the eye. Dithering doesn't cut down resolution.
* There are three types:

1. regular dithering which uses a very regular predefined pattern;
2. random dither where the pattern is a random noise;
3. And pseudo random dither that uses a very large, very regular, predefined pattern.

* Dithering is employed to make patterns to be used as backgrounds, fills and shading, further as for making halftones for printing. When used for printing is it very sensitive to paper properties. Dithering can be combined with rasterising. It is not associated with anti-aliasing.

**Dithering technique:**

* Dither algorithm performs an optimum adding of a sequence of pictures as far as resolution is concerned about. The principle is that, at sub-pixel level, shifts between individual input images are nearly at random distributed. For instance, a star within the 1st image may be centred perfectly in the middle of a pixel, whereas it’ll be across 2 pixels within the other, and so on.
* Since it’s simple to understand the precise shift between the pictures, it’s possible to make an output image with a finer sampling, within which resolution may be increased with respected to every input image. In fact, energy from every input pixel is dropped within the output image.



**Figure 1** shows a dither pattern generated by a random number generator, even with a randomized seed. Note that the distribution of points isn’t very uniform. Indeed, there are some points that would result in sensor noise not being eliminated however actual fact being reinforced due to insufficient difference in positions.

**Figure 2** shows the dither pattern those results from the employment of this algorithm. It achieves maximize separation from one frame to a different with a minimum overall movement of the guide star.

**Usage:**

* Dither should be added to any low-amplitude or highly-periodic signal before any quantization or re-quantization method, so as to De-correlate the quantization noise with the input signal and to stop non-linear behaviour (distortion); the lesser the bit depth, the greater the dither must be.
* The results of the method still yield distortion, however the distortion is of a random nature therefore its result’s effectively noise.
* Any bit-reduction method should add dither to the wave form before the reduction is performed.

**Half toning**

* Many displays and hardcopy devices are bi-level
* They will solely produce two intensity levels.
* In such displays or hardcopy devices we can produce a noticeable increase within the variety of available intensity value.
* 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.
* The phenomenon of apparent increase within the number of available intensities by considering combine intensity of multiple pixels is known as half toning.
* The half toning is commonly used in printing black and white photographs in newspapers, magazines and books.
* The pictures produced by half toning process are referred as halftones.
* In computer graphics, halftone reproductions are approximated using rectangular pixel regions, say 22 pixels or 33 pixels.
* These regions are called halftone patterns or pixel patterns.
* Figure shows the halftone pattern to make a number of intensity levels.

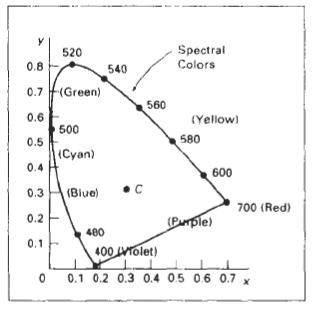


Chromatic Colour:

* Light- It’s a physical phenomenon. Light is an electromagnetic radiation within the [400 nm-700nm] wavelength range.
* Colour- It’s a psychological phenomenon. Interaction of the light of various wavelengths with our visual system causes the colour.
* Chromatic are the visible light, visual range: 400nm-700nm. The visible region of the spectrum is the chromatic colour region.
* Properties of Chromatic colours are:
* Hue: This helps in distinguish colour like red, green, purple etc.
* Saturation: This shows how far the colour from a gray intensity is.
* Lightness: Perceived intensity of the reflecting object and brightness is used when the object is an emitter.

CIE Chromaticity diagram:

* In 1931, the CIE (Commission Internationale de l'Éclairage) defined three standard primaries, called X, Y and Z, that can be added to form all visible colours.
* The primary Y was chosen so its colour matching function exactly matches the luminous-efficiency function for the human eye, given by the sum of the three curves.



This is the CIE Chromaticity Diagram showing all visible colours. x and y are the normalised amounts of the X and Y primaries present, and hence z = 1 - x - y gives the amount of the Z primary required.

* The CIE Chromaticity Diagram shows all visible colours. The x and y axis offer the normalised amounts of the X and Y primaries for a particular colour, and hence z = 1 - x - y gives the amount of the Z primary required.
* Chromaticity depends on dominant wavelength and saturation, and is independent of luminous energy. Colours with the identical chromaticity, but different luminance all map to the same point within this region.
* The line joining the red and violet spectral points, called the purple line***,*** is not part of the spectrum. Interior points represent all possible visible colour combinations. Point *C* in the diagram corresponds to the white-light position.
* Actually, this point is plotted for a white-light source known as illuminant C, which is used as a standard approximation for "average" daylight.
* Luminance values are not available in the chromaticity diagram because of normalization. Colours with different luminance but the same chromaticity map to the same point.
* The chromaticity diagram is used for the following:

1. Comparing colour gamuts for various sets of primaries.
2. Identifying complementary colours.
3. Determining dominant wavelength and purity of a given colour.

* The pure colours of the spectrum lie on the curved part of the boundary, and a regular white light has colour defined to be near (but not at) the point of equal energy x = y = z = 1/3.
* Complementary colours, i.e. colours that add to give white, lie on the endpoints of a line through this point. All the colours along any line in the chromaticity diagram may be obtained by mixing the colours on the end points of the line.
* Furthermore, all colours within a triangle may be formed by mixing the colours at the vertices. This property illustrates graphically the fact that all visible colours cannot be obtained by a mix of R, G and B (or any other 3 visible) primaries alone, since the diagram isn’t triangular.

RGB,CMY,YIQ,HSV,HLS Color Models.

**RGB Color Model:-**

The RGB color model is an additive color model where, green, and blue light are added together in various ways to reproduce a broad array of colors. The name of the model comes from the initials of the 3 additive primary colours, red, green, and blue.

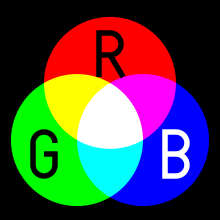
The main purpose of the RGB color model is for the sensing, representation, and display of images in electronic systems, such as televisions and computers, though it has also been used in conventional photography. Before the electronic age, the RGB color model already had a solid theory behind it, based in human perception of colors.

**Additive Colours**

RGB Color Model is referred to as (Additive Colors) the colors area unit called as Primary color

To form a color with RGB, 3 light beams (one red, one green, and one blue) must be superimposed (for example by emission from a black screen or by reflection from a white screen). Every of the 3 beams is termed a *component* of that color, and every of them will have an arbitrary intensity, from fully off to fully on, within the mixture.

The RGB color model is *additive* within the sense that the 3 light beams are added together, and their light spectra add, wavelength for wavelength, to create the ultimate color's spectrum. This is essentially opposite to the subtractive color model, particularly the CMY color model, that applies to paints, inks, dyes, and different substances whose color depends on *reflecting* the light under which we see them. Because of properties, these 3colorsproducewhite, this is often in stark contrast to physical colors, like dyes that produce black once mixed.



Each color is drawn in its primary color elements Red, Green and Blue.

This model is based on Cartesian Coordinate System. 

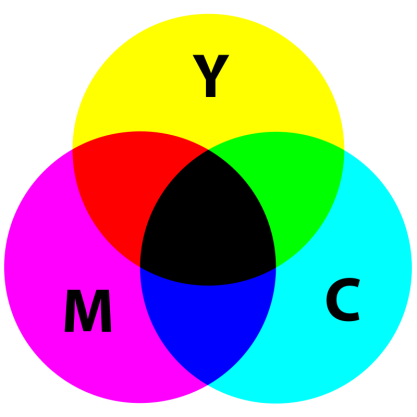
**RGB Color System**

* Additive color model.
* For computer Screens (Displays).
* Uses light to display color.
* Colors result from transmitted light.
* Red + Green + Blue = White.
* The combination of Red, Green and Blue in fully intensity makes white.
* White light is created when all color of the EM spectrum (electromagnetic spectrum) converge in fully intensity

**Importance of RGB color Model**

* The color model RGB is used in hardware applications like Computer monitors, cameras and scanners.
* It is used for Web graphics; however it cannot be used for print production.
* It directly reflects the physical properties of “True color” displays.
* It is used—
* For sensory illustration.
* Show of text pictures in electronic system.
* For example- Laptop, TV, camera.

**CMY Color Model:-**



The **CMY color model** is a subtractive color model in which cyan, magenta and yellow pigments or dyes are added together in various ways to reproduce a broad array of colors. The name of the model comes from the initials of the 3 subtractive primary colors, cyan, magenta, and yellow.

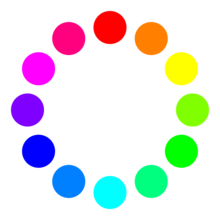
The CMY color model itself doesn’t outline what’s meant by *cyan*, *magenta* and *yellow* calorimetrically, and so the results of blending them aren’t given as absolute, however relative to the primary colors. When the exact chromatic ties of the cyan, magenta, and yellow primaries are defined, the color model then becomes an absolute color space.

$\displaystyle \begin{pmatrix}
C \\ M \\ Y
\end{pmatrix}=
\begin{pmatrix}
1 \\ 1 \\ 1
\end{pmatrix}-
\begin{pmatrix}
R \\ G \\ B
\end{pmatrix}$

**Subtractive Colors**

The CMY color model is *subtractive* in the sense that mixtures of dye (typically sitting on a white substrate like paper) subtract specific wavelengths from the spectral power distribution of the illuminating light that is scattered back into the viewer's eye and is perceived as coloured.

Layering dyes on top of each is used to reproduce a gamut of colors, the resultant color from this layer is expected by multiplying (not subtracting)the absorbance profiles of the dyes. This is often basically opposite to the additive color model, significantly the RGB color model, that applies to lights whose color depends directly on the light.



Clockwise from the top: red, orange, yellow, chartreuse, green, spring, cyan, azure, blue, violet, magenta, and rose.

**CMYK color model** is a subtractive color model, based on the CMY color model, used in color printing, and is additional wants to describe the printing process itself. CMYK refers to the four inks used in some color printing: cyan, magenta, yellow, and key. It uses K, black ink, since C, M, and Y inks area unit are clear and can solely turn out a grey color once mixed.

**YIQ Color Model**

This is used for color TV. Here $ Y$ is the luminance (the only component necessary for B&W-TV). The conversion from RGB to YIQ is given by

$\displaystyle \begin{pmatrix}
Y \\ I \\ Q
\end{pmatrix}=
\begin{pmatrix}
0.30 &...
...& -0.52 & 0.31 \\
\end{pmatrix}\cdot
\begin{pmatrix}
R \\ G \\ B
\end{pmatrix}$

for standard NTSC RGB phosphor with chromaticity values

|  |  |  |  |
| --- | --- | --- | --- |
|  | R | G | B |
| x | 0.67 | 0.21 | 0.14 |
| y | 0.33 | 0.71 | 0.08 |

The advantage of this model is that more bandwidth can be assigned to the Y-component (luminance) to which the human eye is more sensible than to color information. So for NTSC TV there are 4MHz assigned to$ Y$, 1.5MHz to $ I$ and $ 0.6$MHz to Q.

**HSV Color Model:-**

All color models treated so far are hardware oriented. The Hue-Saturation-Value model is oriented towards the user/artist. The allowed coordinates fill a six sided pyramid the 3 top faces of the color cube as base. Note that at the same height colors of different perceived brightness are positioned. Value is given by the height; saturation is coded in the distance from the axes and hue by the position on the boundary.

|  |
| --- |
| \begin{figure}\begin{picture}(5,6)(1,0) \put(2,4){\makebox(0,0){$\bullet$}\makeb... ...(0,-0.3){Black}} \put(1.7,2.2){\makebox(0,0.3){White}} \end{picture}\end{figure} |
| The HSV-model versus the RGB-model |

Note that conversion from RGB to HSV is given by affine coordinate changes on each of the 3 four-sided sub-pyramids corresponding each to 1/3 of the color cube.

**HLS Color Model**

Here the RGB-cube is deformed in such a way that a six sided double pyramid results with the same base as in the HSV-model, but with two tips at black and at white.

|  |
| --- |
| \begin{figure}\begin{picture}(6,6) \put(2,2){\makebox(0,0){$\bullet$}\makebox(-0... ...put(3,4.5){\line(4,-3){2}} \put(3,4.5){\line(2,-1){1}} \end{picture}\end{figure} |
| The HLS-model |