LOVELY PROFESSIONAL UNIVERSITY

COMPUTER GRAPHICS

Edited By

Mandeep Kaur

ISBN: 978-93-87034-65-5

Printed by
EXCEL BOOKS PRIVATE LIMITED
Regd. Office: E-77, South Ext. Part-I, Delhi-110049
Corporate Office: 1E/14, Jhandewalan Extension, New Delhi-110055
+91-8800697053, +91-011-47520129
info@excelbooks.com/projects@excelbooks.com
internationalalliance@excelbooks.com
www.excelbooks.com

for
Lovely Professional University
Phagwara

CONTENTS

Unit 1:	Fundamentals of Computer Graphics *Sarabjit Kumar, Lovely Professional University*	1
Unit 2:	Colors in Computer Graphics *Avinash Bhagat, Lovely Professional University*	11
Unit 3:	Overview of Graphics I/O Devices *Anil Sharma, Lovely Professional University*	21
Unit 4:	Scan Conversion I *Yadwinder Singh, Lovely Professional University*	39
Unit 5:	Scan Conversion II *Mithilesh Kumar Dubey, Lovely Professional University*	53
Unit 6:	2-D Transformation *Kumar Vishal, Lovely Professional University*	63
Unit 7:	2-D Viewing *Anuj Sharma, Lovely Professional University*	89
Unit 8:	3-D in Computer Graphics *Balraj Kumar, Lovely Professional University*	105
Unit 9:	Clipping I *Pawan Kumar, Lovely Professional University*	129
Unit 10:	Clipping II *Pawan Kumar, Lovely Professional University*	143
Unit 11:	Hidden Surfaces *Kamlesh Lakhwani, Lovely Professional University*	155
Unit 12:	Color and Shading Model *Balraj Kumar, Lovely Professional University*	171
Unit 13:	Advanced Computer Graphics *Yadwinder Singh, Lovely Professional University*	191
Unit 14:	Animation in Computer Graphics *Ajay Kumar Bansal, Lovely Professional University*	213

SYLLABUS

Computer Graphics

Objectives: This course provides an introduction to the principles of computer graphics. In particular, the course will consider methods for modelling 3-dimensional objects and efficiently generating photorealistic renderings on colour raster graphics devices. The emphasis of the course will be placed on understanding how the various elements that underlie computer graphics (algebra, geometry, algorithms and data structures, optics, and photometry) interact in the design of graphics software systems.

S. No.	Description
1.	Fundamentals of Computer Graphics: Image Files, Image Processing and User Interface, Overview of Image Representation, Color Models, Direct Coding, Lookup Table, Setting the Color Attributes of Pixels, Mandelbrot Set, Julia Set
2.	Overview of Graphics I/O Devices: Input Devices, Output Devices, Printers and Projectors, Image Persistence, Resolution, Frame Buffer, Logical Functioning of I/O Devices
3.	Scan Conversion: Mechanisms and Methods, Line, Circles, Ellipses, Polygons, Region Filling, Glyph, Aliasing and Antialiasing, Arcs and Sectors
4.	2-D Transformation: Transformation Matrix, Homogenous Co-ordinate System, Composing and Inverting Transformation, Affine Transformation
5.	2-D Viewing: Concept of Window and Viewport, Window-to-Viewport Mapping, Graphic Pipeline, Panning and Zooming
6.	3-D in Computer Graphics: 3-D Representation, 3-D Transformation, 3-D Viewing
7.	Clipping: Overview of Clipping, Point Clipping, Line Clipping, Polygon Clipping, Projection
8.	Hidden Surfaces: Z-Buffer, Binary Space Partitioning, Painter's Algorithm, Warnock Algorithm, Ray tracing Vs. Rasterization, Determination, Removal
9.	Color and Shading Model: Light and Color, Phong Model, Interpolative Shading Methods
10.	Advanced Computer Graphics: Texturing, Ray Tracing, Morphing, Animation, Additional Visual Effects

Unit 1: Fundamentals of Computer Graphics

CONTENTS

Objectives

Introduction

1.1. Image Files

1.2. Image Processing and User Interface

1.2.1 Image Processing

1.2.2 User Interface

1.3. Overview of Image Representation

1.4. Direct Coding

1.5. Lookup Table

1.6. Summary

1.7. Keywords

1.8. Self Assessment

1.9. Review Questions

1.10 Further Readings

Objectives

After studying this unit, you will be able to:

- Define image files
- Comprehend image processing and user interface
- Analyze overview of image representation
- Describe direct coding

Introduction

Computer graphics is a field of computer science, which deals with creation, representation and management of images on the computer screen. Computer graphics deals with the technological and theoretical aspects of computerized image synthesis. An image created by a computer can illustrate a simple scene as well as complex scenes.

Example: A simple scene would be the outline of a triangle on an even background, and a complex scene would be a stunning dinosaur in a forest.

It can be said that computer graphics is an immense field that includes almost any graphical aspect, such as generation of images.

Did you know? A Massachusetts Institute of Technology (MIT) student, Ivan Sutherland was the first to create a computer graphic. Sutherland produced a computer drawing program called Sketchpad in 1961. With the help of a light pen, the Sketchpad allowed one to draw simple shapes on the computer screen, save them, and even retrieve them later.

Computer graphics covers many important elements in the field of computer science, such as the processing of images, interaction with images, representation of images, and display of images. Due to these important elements, today, computer graphics have become an important aspect of our daily lives. Computer graphics is used in almost all the fields today, such as games, weather reports, or all kinds of medical investigation and surgical procedures.

1.1 Image Files

An image is made up of a rectangular grid of pixels. It has a definite height and a definite width counted in pixels.

An image file refers to any pictorial representation that is already stored in the computer memory. An image file format refers to the particular format in which an image file is stored. Generally, a file format stores the number of rows and columns of image pixels, in the header information of the image file.

You can save an image in different file formats. The most commonly used image file formats are TIFF, JPG, and GIF, which are very important in the process of printing, process of scanning and Internet use.

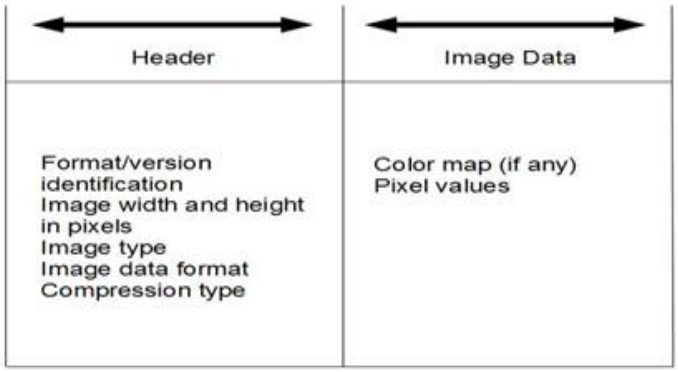
Notes In Internet browsers, TIF cannot be used.

It is very important to know that a digital image is normally encoded in the form of a binary file for the function of transmission and storage.

Did you know? The pixels that represent an image are structured as a grid (columns and rows), where every single pixel comprises numbers that depict the degree of color and brightness.

Figure 1.1 depicts a typical organization of image files with reference to information encoded in an image file.

Figure 1.1: Typical Image File Organization



As shown in the figure 1.1, a typical image file mainly consists of two parts, namely, header and image data. At the beginning of the header, a binary code or American Standard Code for Information Interchange (ASCII) string identifies the format being used along with the version number. The height and width of the image are specified in the number of pixels. The most common image type includes black and white (1 bit/pixel), 8 bit gray scale (256 levels with the gray axis), 8 bit color (lookup table), and 24 bit color.

The main function of an image data format is that it specifies the order, in which, the pixel values are stored in the image data section. Generally, the left to right and top to bottom orders are used, and

another possible order is the vice versa of the normal order. The other function of the image data format is that it specifies whether the RGB values are interlaced in the color map or in the image. When the values are in an interlaced fashion, the RGB components for an exact pixel stay together consecutively. This is followed by the three color components for the next entry or pixel. Therefore, the color values in the image data section are a sequence of red, green, blue…, red, green, blue, and so on.

When the values are in non-interlaced fashion, the values of one primary color for all pixels or table entries come first, and then the values of another primary color appear. This is followed by the values of third primary color. Therefore, the image data is in the form of red, red…, green, green…, blue, blue.

The image data's values may be compressed with the help of a compression algorithm, such as Run Length Encoding (RLE) algorithm. The main idea of RLE can be demonstrated with a character string "xxxxxxyyzzzz", which takes 12 bytes of storage. At this instant, if the string is scanned from left to right for segments of repeating characters, and each segment is substituted by 1 byte, the count is repeated when the character being followed is also repeated. The compression or conversion of the given string to "6x2y4z", which takes only 6 bytes will be easy to do. This converted or compressed version can be expanded or decompressed, by repeating the character following each repeat count to recover the actual string.

The file header's length is often fixed. If it is not fixed, then it is important to include the length information in the header, to indicate where exactly the image data starts.

Notes

Some of the image data file formats include header length. The length of every component in the image data section is dependent on certain factors, such as compression type and image type. In addition, format specific information can also be found in the header.

1.2 Image Processing and User Interface

Computer graphics consists of many important elements, which make it appropriate for every field. Two such elements are image processing and user interface. These two concepts must be understood for the clear understanding of different concepts related to images, such as image representation, image display, and so on.

1.2.1 Image Processing

Image processing is a method used to enhance or develop raw images. Raw images refer to the images that are taken by cameras, satellites, space probes, and aircrafts or even pictures that are taken in our normal day-to-day lives.

During the last four to five decades, many techniques have evolved in image processing. Most of the various techniques are created to enhance images obtained from space probes, unmanned spacecrafts, and military investigation flights. Image processing systems are becoming prominent. The prominence is due to accessibility of large size memory devices, powerful personal computers, graphics software, and so on.

Today, image processing is applied in many fields, such as, medical imaging, remote sensing, forensic studies, textiles, military, film industry, document processing, graphic arts, printing industry, and so on.

There are four steps in image processing. These steps are:

1. ***Image Scanning*:** It is the action or process of converting the images to digital images.
2. ***Storing*:** It is nothing, but the way the image is stored.

Example: Drawing applications store pictures, spreadsheet applications store charts, Computer Aided Design (CAD) applications store drawings.

3. ***Enhancing*:** It is a process aimed at enhancing the visual appearance of an image.

4. ***Interpretation*:** It is the process of investigating the images, recognizing and judging its significance by considering their location and extent.

There are two methods available in image processing:

1. ***Analog Image Processing*:** Analog image processing refers to the modification of image using electrical means.

Example: A familiar example is the image you see on a television screen. A television signal is a voltage level which ranges in amplitude to signify brightness through the image. By varying the signal electrically, you can alter the appearance of the displayed image. The contrast and brightness controls on a television regulate the amplitude and reference of the video signal. This results in brightening, darkening, and alternating of the brightness range of the displayed image.

2. ***Digital Image Processing*:** In digital image processing, an image is processed with the help of digital computers. The image is first transformed to digital form, using a scanner and then it is processed. The digital image processing starts with a single image and produces a transformed version of the same image.

   Digital image processing usually refers to a two-dimensional image processing by a digital computer. A digital image is an array of real numbers characterized by a finite number of bits. The main benefit of digital image processing method is its repeated usability, adaptability, and the protection of the original data.

Now, it can be said that image processing is nothing, but a process, which is used to enhance an image

1.2.2 User Interface

As the name suggests, user interface is the intermediary between the user and the computer. User interface design refers to the designs of computers, machines gadgets, appliances, software applications, websites, and mobile communication devices, where the major emphasis is on the interaction and experience of the user.

Unlike the traditional design, whose objective was to make the application or object eye catching, the objective of user interface design is to create the user's interaction experience as simple and intuitive as possible. This process of achieving the user interface design objective is frequently called as user-centered design. A good user interface design is normally restrained and invisible, where a good graphic or a design is bold and eye catching.

By now it is clear that the user friendliness is one of the major essential factors for the success and popularity of any interface. The graphical interfaces provide an attractive and easy interaction between the computers and the humans.

Did you know? Xerox Corporation's Palo Alto Research Center designed the first graphical user interface in 1970s. However, it was in 1980s the graphical user interfaces became popular after the emergence of the Apple Macintosh.

A Graphical User Interface (GUI) is nothing, but a program interface that makes the program easier to use with the help of the computer graphics' abilities. Some basic components of a GUI are:

1. ***Pointer:*** Pointer is a symbol appearing on the screen, which is used to select objects and commands. Normally, a pointer looks like a small angled arrow.

2. ***Pointing Device*:** Pointing device is a device which enables the user to choose objects on the display screen.

Example: A mouse is an example of a pointing device.

3. ***Icons*:** Icons are the small images or pictures that signify commands, files, or windows.

4. ***Menus*:** Menu helps the user to select a command from a list of options to execute a function.
5. ***Scroll Bars*:** Scroll bars help the user to scroll sideways, up and down on a page or application.

A GUI offers a simple and easy modification means and data entry. GUI even helps a computer illiterate to use the system easily. With the help of GUIs, one can use a mouse to run programs, edit, copy, and delete files by selecting appropriate options from the menus. GUI also partitions the display screen into different sections or regions, wherein the outputs of different applications or programs are shown. Thus, GUIs offer a better means of communication with computer as compared to unwieldy text based interfaces.

Adoption of a standard GUI proves to be very helpful, since a standard GUI maintains the consistency of use across all the platforms and applications. All the Microsoft Windows and Apple Macintosh applications have similar look and feel. Therefore, the learning time for a new application does not take much time. Today, it is possible to run multiple applications simultaneously on most of the machines. To maintain uniformity and consistency in the standard GUI there are design rules to be followed such as:

1. Interface should be kept as simple as possible and consistent.
2. Provide easy access to common applications by keeping the user in mind always.
3. Provide customization control as per user preferences.
4. Communicate the application actions to the user clearly.

Thus, it can be concluded that GUI is a very important component of computer graphics.

1.3 Overview of Image Representation

Every computer has images. Most of them are clear, like icons on buttons and photos on web pages and some are more restrained. In particular, the great quantity of data in an image means that compression needs to be used for storing and transmitting that data efficiently.

Images must be represented in an appropriate manner, so that they can be clearly seen and understood. There are two main types of image representation:

1. ***Raster Image Representation*:** A digital image or a normal image is composed of picture elements or discrete pixels. These pixels are prearranged in a row and column style, to outline a rectangular picture area. It is sometimes referred to as a Raster. Many images are normally represented as sets of pixels encoding brightness/color information in the form of a matrix. This type of image representation is related to the matrix image representation, where every element of the matrix (pixel) possesses its own value, which corresponds to the brightness of that element. In this matrix form of image representation, every pixel has a value that is equal to 0 (background), or 1 (object), which needs one bit per one pixel. Generally, this type of image representation is not used in a direct manner in the information systems of computer graphics, except for activities, such as compression, storage, and transmission of information.
2. ***Vector Image Representation*:** In this type of image representation, curves or points with vectors that connect these points are used to represent every graphical object or image.

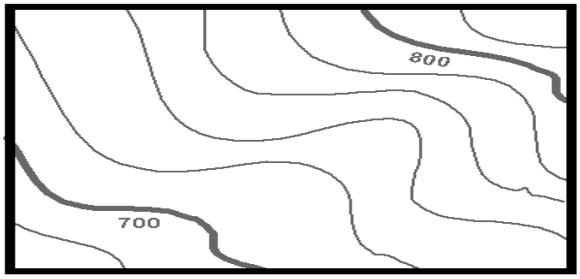
Apart from raster and vector image representations, images can also be represented using a model that is based on contour lines.

Notes

A contour line is a curve beside which the function has a constant value.

A typical contour image representation is clearly depicted in the figure 1.2.

Figure 1.2: Contour Image Representation



A contour representation helps in easy recovery of the full image in bitmap form. It has been used mainly for data compression of an image. The main idea is to program each level with the boundaries of connected regions of pixels at levels superior than or equal to each level. It is easy to recreate an original image from these boundaries.

One of the main issues in using contour image representations is how to save such representations in a compact manner. In practice, the complete contour representation is hardly ever required. Its distinctive use is in the form of a query, where it asks for the contours matching for a given gray level. To provide efficient solutions, the current data structuring techniques should be used. However, they have not yet been used, but attempts are being made in order to solve this.

Example: Consider a typical scenario encountered with this kind of representation. Presume that you want to erase wrinkles around the eyes of a person in a digitized picture given in contour representation. As the image of the wrinkles may intersect contour lines, these may appear to be detached after removal of the wrinkles. It is not easy to reconnect these contour lines. Dynamic programming might be a normal approach to solve this problem.
Note: The dynamic programming has been tried in numerous experiments effectively. The only drawback of dynamic programming is that it is very costly.

In fact, the total number of pixels in an image is a function of the size of the image and the number of pixels per unit length, such as inch in the horizontal as well as the vertical direction. This number of pixels per unit length is the resolution of the image.

Example: A 3×2 inch image at a resolution of 300 pixels per inch would have a total of 5, 40,000 pixels.

In general, the image size is given as the total number of pixels in the horizontal direction multiplied by the total number of pixels in the vertical direction, such as 512×512, 640×480, 800×600, or 1024×768. Even though, this convention makes it comparatively easy to measure the total number of pixels in an image, it does not specify the size of the image or its resolution.

Example: A 640×480 image would measure 62/31 inches by 5 inches when displayed or printed at 96 pixels per inch. Alternatively, the same image would measure 1.6 inches × 1.2 inches at 400 pixels per inch.

The ratio or the proportion of an image's width to its height, measured in unit length or number of pixels, is referred to as **aspect ratio**.

Example: A 6×41/2 inch image and a 1024×768 inch image have an aspect ratio of 4/3, whereas 2×2 inch image and 512×512 inch image have the aspect ratio of 1/1.

An image's individual pixels can be indicated by their co-ordinates. Classically, the pixel at the lower left corner of the image is considered to be at the origin (0, 0) of a pixel co-ordinate system.

Example: The pixel at the lower right corner of a 640×480 image would have (639, 0) co-ordinates, the pixel at the upper left corner would have (0, 479) co-ordinates and the pixel at the upper right corner would have (639, 479) co-ordinates.

1.4 Direct Coding

The representation of image is fundamentally the representation of pixel colors. Direct coding helps to allocate a specific amount of storage space for each pixel, in order to code its color.

Example: Consider a scenario, where 3 bits are allocated for each pixel, where each primary color will have one bit. This representation of 3 bit permits each primary color to vary independently between two intensity levels like 0 being the off and 1 being the on. Therefore, each pixel is able to take any of the eight colors which correspond to the RGB color cube corners.

The table 1.1 depicts the direct coding of colors using the 3 bits given in the above example.

Table 1.1: Direct Coding of Colors with 3 bits

Red Bit 1	Green Bit 2	Blue Bit 3	Color Names
0	0	0	Black
0	0	1	Blue
0	1	0	Green
0	1	1	Cyan
1	0	0	Red
1	0	1	Magenta
1	1	0	Yellow
1	1	1	White

A widely accepted industry standard uses 3 bits or 24 bits per pixel, with one byte for each primary color. With the help of this, all the primary colors are allowed to have 256 different intensity levels equivalent to binary values from 00000000 to 11111111. Therefore, a pixel can take on a color from 256×256×256 or 16.7 million possible choices. A 24-bit format is commonly referred to as the true color representation. The reason is the difference between two colors, which differ by one intensity level in one or more of the primary colors. This difference is virtually unnoticeable under normal viewing conditions. Therefore, a precise representation involving more bits is of modest use in terms of supposed color accuracy.

The most notable special aspect of direct coding is the black, white and gray scale image representations.

A black and white image needs only one bit per pixel, where the bit value 0 represents black and bit value 1 represents white. The gray scaled image is typically coded with 8 bits per pixel, in order to allow a total of 256 intensity levels or gray levels.

Although this direct coding method has simplified and supported a variety of applications, there is a relatively high demand for storage space with reference to the 24-bit standard can be seen.

Did you know? A 1000×1000 true color image can take up three million bytes. In addition, even if every pixel in that image has a different color, there would almost be a million colors in the image.

In most of the applications, the total number of colors that appear in any one particular image is much less. Therefore, the ability of 24-bit representation to have 16.7 million different colors appear all together in a single image seems to be overloaded.

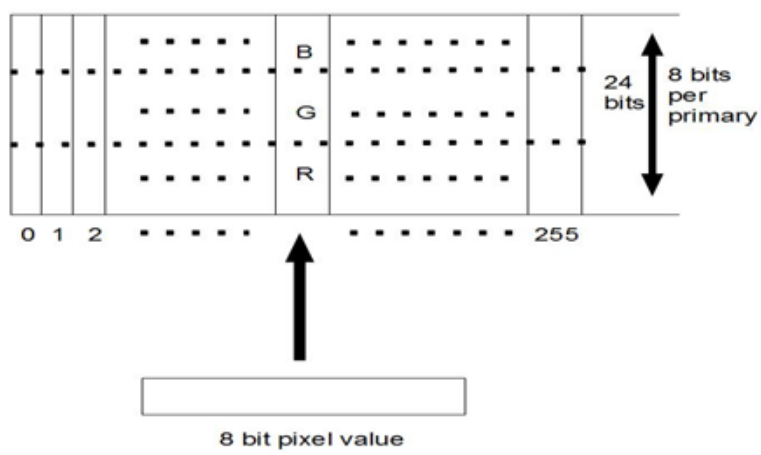
1.5 Lookup Table

In computer graphics, an important mechanism is used for converting a variety of colors into another variety of colors. This mechanism is referred to as a lookup table, also called as a Color Look Up Table (CLUT). Lookup table is generally used for partitioning the pixel values into the colors that are displayed, and the colors that are not displayed. Generally, the operating system initializes this table to particular default values. The lookup table size relies on the number of colors that the graphics system can display simultaneously.

The representation of image with the help of a lookup table can be viewed as a negotiation between the desire to have a lower storage necessity and the requirement to support a reasonably sufficient number of simultaneous colors. The main aspect of this approach is that there will be no direct coding of colors with pixel values. Instead, the pixel values are addresses of color values in a table. The color of a certain pixel is concluded by the color value in the table entry, which the pixel value refers to.

The figure 1.3 depicts the lookup table with 256 entries.

Figure 1.3: 24-Bit Lookup Table with 256 Entries



In the figure 1.3, the entries have addresses ranging from 0 to 255. Each entry has a 24 bit RGB color value. Pixel values are now 8 bit or 1 byte quantity. The color of a pixel, whose value is i, where $0 < i < 255$ is determined by the color value in the table entry, whose address is i. The above lookup table representation is often referred as the 8 bit format. It decreases the storage requirement of a 1000×1000 image to one million bytes plus 768 bytes for the color values in the lookup table. It selects 256 simultaneous colors from 16.7 million possible colors.

One important application of lookup table is in animation.

Example: Lookup tables are used in animation displaying short sequence of images that are repetitive.

1.6 Summary

- Creation, representation, and management of images on the computer screen come under the computer graphics which is a field of computer science.
- An image is a minute rectangular box or a grid of pixels.
- An image file refers to any pictorial representation which is stored in the computer memory.
- Image processing is a method used to develop or enhance raw images.
- Designs with reference to appliances, computers, machines, and gadgets are called as User interface design.
- Image representation is nothing, but the depiction of images in clear and comprehensible manner.
- Direct coding helps us to allocate a specific amount of storage space for each pixel, in order to code its color.
- Lookup table is the mechanism used for converting a variety of colors into another variety of colors.

1.7 Keywords

GIF: Graphic Interchange Format.

JPG/JPEG: Joint Photographic Experts Group.

TIFF: Tagged Image File Format.

Unwieldy: Difficult to carry or manage because of size, shape, weight, or complexity.

1.8 Self Assessment

1. State whether the following statements are true or false:
   (a) The main function of an image data format is that it specifies the order, in which, the pixel values are stored in the image data section.
   (b) Image enhancing is the action or process of converting the images to digital images.
   (c) A pointer is a device which enables you to select objects on the display screen.
   (d) A contour representation helps in easy recovery of the full image in bitmap form.
   (e) The ratio or the proportion of an image's width to its height, measured in unit length or number of pixels, is referred to as aspect ratio.
2. Fill in the blanks:
   (a) The file header's length is often ……..
   (b) …….. is a process intended to enhance the visual appearance of an image.
   (c) …… offer an easy means of modification and data entry.
   (d) An image's individual pixels can be indicated by their……...
   (e) The lookup table size relies on the number of colors that the …… system can display simultaneously.

3. Select the suitable choice for every question:

   (a) One of the important elements of computer science which is covered by computer graphics is

   (i) Interaction with images (ii) Image transition

   (iii) Image storage (iv) Image buffer

   (b) A 6×41/2 inch image and a 1024×768 inch image have an aspect ratio of

   (i) 3/4 (ii) 4/3

   (iii) 1/1 (iv) 2/3

   (c) Classically, the pixel at the lower left corner of the image is considered to be at the origin

   (i) (1, 0) (ii) (0, 1)

   (iii) (0, 0) (iv) (1, 1)

1.9 Review Questions

1. "One of the main issues in using contour image representations is how to save such representations in a compact manner". Explain.
2. "A typical image file mainly consists of header and image data". Describe.
3. "The image data's values may be compressed with the help of a compression algorithm". Which algorithm?
4. "During the last four to five decades, various techniques have been developed in image processing". Explain.
5. "Analog image processing refers to the modification of image with the help of electrical means". Explain with an example.
6. "Curves or points with vectors that connect these points are used to represent every graphical object or image". Explain under which image representation this is observed.
7. "The most notable special aspect of direct coding is the black, white and gray scale image representations". Explain.
8. "The representation of image with the help of a lookup table can be viewed as a negotiation between the necessity to have a lower storage necessity and the requirement to support a reasonably sufficient number of simultaneous colors". Discuss.

Answers: Self Assessment

1. (a) True (b) False (c) False (d) True (e) True
2. (a) Fixed (b) Enhancing (c) GUIs

   (d) Co-ordinates (e) Graphics
3. (a) Interaction with images (b) 4/3 (c) (0, 0)

1.10 Further Readings

Shirley P, Marschner S., Fundamentals of Computer Graphics. Library of Congress Catalogue-in-Publication Data.

http://diwww.epfl.ch/lami/detec/perrigproj96/node10.html

Unit 2: Colors in Computer Graphics

CONTENTS

Objectives

Introduction

2.1. Color Models

2.1.1 RGB Color Model

2.1.2 CMYK Color Model

2.2. Setting the Color Attributes of Pixels

2.3. Mandelbrot Set

2.4. Julia Set

2.5. Summary

2.6. Keywords

2.7. Self Assessment

2.8. Review Questions

2.9. Further Readings

Objectives

After studying this unit, you will be able to:

- Explain red green blue (RGB) color model
- Define setting the color attributes of pixels
- Explain Mandelbrot set
- Define Julia set

Introduction

Color plays an important role in images. The light enters our eyes in the form of a spectrum with variation in the wavelengths. Human beings have three kinds of cones in their eyes that respond to different colors with different wavelengths. The basic colors red, green and blue represents long, medium and short wavelengths respectively.

Did you know? The human vision is three-dimensional known as principle of trichromacy. According to this, the spectrum of light is encoded into three values that correspond to the amount of light absorbed by each type of cone.

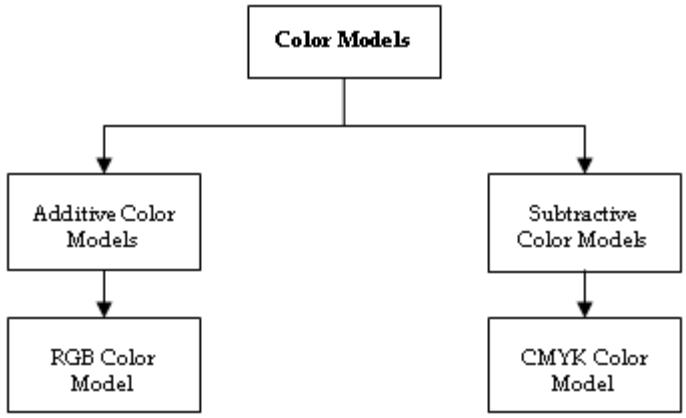
This unit gives an overview of the RGB color model, CMYK color model, setting the color attributes of pixels, Mandelbrot set and Julia set.

2.1 Color Models

Color is a complex, interdisciplinary phenomenon spanning from psychology to physics. As the name suggests, color model is a model of colors. It can be defined as an organized system, which is used for the creation of a wide variety of colors starting from a small collection of primary colors.

Color models are of two types, as depicted in the figure 2.1.

Figure 2.1: Color Models



1. ***Additive Color Models*:** Additive color models are those color models, which use light for displaying colors. In these color models, colors are seen as an aftereffect of transmitted light.

Example: The RGB color model is an example of an additive color model.

2. ***Subtractive Color Models*:** Subtractive color models are those color models, which use printing inks for displaying colors. In these color models, colors are seen as an aftereffect of reflected light.

Example: The CMYK color model is an example of a subtractive color model.

2.1.1 RGB Color Model

The RGB color model is an additive color model, which uses three primary colors, namely, red (R), green (G), and blue (B). These primary colors can further be used to create secondary colors, such as yellow, white, and magenta. Computers can display approximately 16.7 million colors using this color model.

In RGB model, every color's measurement is done with a value that ranges from 0 to 255, where **0** means no light, and value **255** means maximum intensity. This (0 to 255) is the limit of storing information in 1 byte of computer memory. In RGB model, 3 bytes (24 bits) of information are needed for defining the three primary colors RGB.

A large percentage of the visible band of colors can be represented by mixing red, green, and blue (RGB) colored light in various intensities and proportions.

RGB colors are also called as **additive colors**, as they combine to create white color. In other words, adding all colors creates white, that is, all visible wavelengths are transmitted back to the eye. Additive colors are used for monitors, video, and lighting.

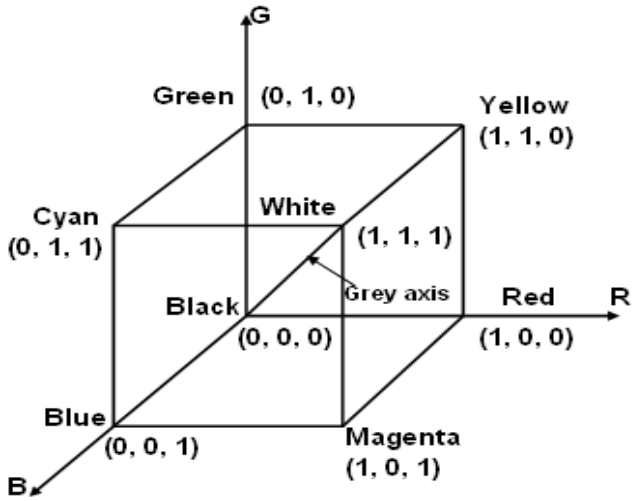
Did you know? Your computer or television monitor, creates color by emitting light through red, green, and blue phosphors.

Notes

The computer monitor's screen is black, that is, it emits no light. The color display is created, when the three primary colors RGB are generated. Intermediate colors are produced, when there is a variation of the intensities of the red, green, and blue colors.

A unit cube is used to represent the color space of RGB colors. This unit cube is placed at origin in its standard position. The figure 2.2 depicts the color co-ordinate system with the three primary colors of RGB model, that is, RGB.

Figure 2.2: Color Co-ordinate System with RGB Colors



Source: Schaum's outline of theory and problems of computer graphics

Each of the RGB (primary) colors can take up the intensity value ranging from 0 being the lowest to 1 being the highest. When these primary colors are mixed at different intensity levels, they produce a variety of colors including cyan, magenta, yellow, and white. The collection of all the colors available by such a linear combination of red, green, and blue forms the cube-shaped RGB color space as shown in figure 2.2. The corner of the RGB color cube that is at the origin of the co-ordinate system corresponds to black. The corner of the cube that is diagonally or obliquely opposite to the origin represents white. The oblique line connecting white and black corresponds to all the gray colors between white and black and it is called as the gray axis.

With the help of this RGB color model, a random color within the cubic color space can be specified by its color co-ordinates, that is red, green, and blue.

Notes

(0,0,0) is for black, (1,1,1) is for white, (1,1,0) for yellow.

The color specification using the RGB color model is an additive process. First, black color is considered and the suitable primary component is added on to yield a preferred color.

Task

Research and find the resultant color, when the color intensity of RGB is (1, 0.7, 0.7).

2.1.2 CMYK Color Model

The colors cyan (C), magenta (M), yellow (Y) and black (K) (CMYK) is a subtractive color model, which is mainly used in color print production. This color model is also called as a complementary RGB model. In this color model, inks of four colors, cyan (C), magenta (M), yellow (Y) and black (K) are used for defining colors. Every color has some ink amount, whose measurement is done by using percent starting from 0 to 100, where a value of 100 signifies the application of inks at full intensity. This CMY color model describes colors using a subtractive process, which closely matches the working principles of the printer.

The color space of CMYK color model is called subtractive. The application of inks of cyan, magenta, yellow, and black color to a surface that is white in color, is done for subtracting some color from that white surface. This method of subtraction finally leads to the creation of the final color. This model is generally called as CMY model.

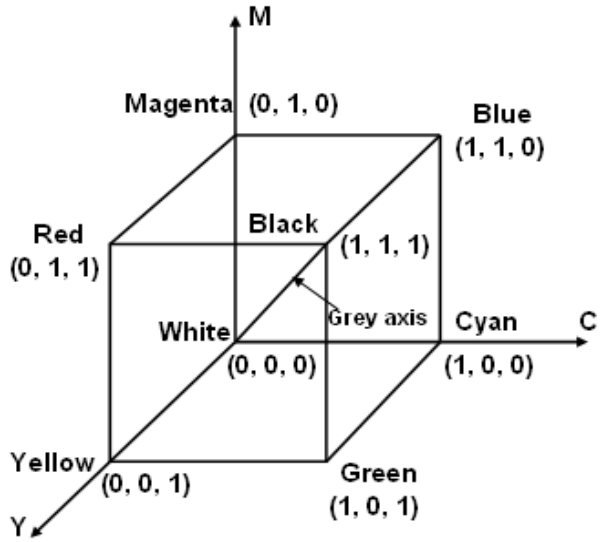
The subtraction of all colors by the CMY combination at full intensity renders black color. Nevertheless, because of impurities that exist in the available CMY inks, complete and equal intensity is made impossible, and filtering of some RGB light takes place, which renders a muddy brown color. Thus, with the addition of the black ink CMY color model takes place.

In the CMY model, first the white color is considered, and the suitable primary components are taken away, to yield a preferred color.

Example: If we subtract or decrease the red color from white color, the color that remains consists of green and blue, which is nothing but the cyan color.

If the above example is considered from another perspective, the quantity of cyan, the complementary color of red, can be used in order to control the amount of red, which is equal to one minus the amount of cyan color. The following figure 2.3 depicts a co-ordinate system using the three primary complementary colors which are cyan (C), magenta (M), and yellow (Y).

Figure 2.3: Color Co-ordinate System with CMY Colors



Source: Schaum's outline of theory and problems of computer graphics

In the figure 2.3, the corner of the CMY color cube, that is, at (0, 0, 0) represents white color, whereas the corner of the cube that is at (1, 1, 1) represents black color. CMY color model is mainly used for the production of printed materials. This is the reason why CMY color model is an important element in printing.

2.2 Setting the Color Attributes of Pixels

Before discussing how to set color attributes of pixels, let us understand the meaning of color attribute. The color attribute is nothing but the visual qualities of saturation, hue, or brightness.

Did you know? Famous physiologist Ewald Hering was the first to understand the color attributes during the 19th century.

The color attribute of pixel is nothing but the visual qualities of saturation, hue, or brightness with reference to a pixel. In computer graphics, the most primal process is the setting of color attributes of individual pixels. It is generally carried out by creating system library calls to write the respective values into the frame buffer. A three-element array, which is a type of aggregate data structure, is generally used to characterize the three primary color components. In spite of image type, which is nothing but direct coding versus lookup table, there are two possible protocols for the specification of pixel co-ordinates and color values.

In a single protocol, the application offers both color information and co-ordinate information simultaneously.

Example: Suppose rgb is a three-element array with rgb [0] = r, rgb [1], and rgb [2] = b. Therefore, a call of setting a pixel at location (x, y) in a 24 bit image to color (r, g, b) would look like setPixel (*x, y, rgb*).

Alternatively, if the image in the above example uses a lookup table and assuming that the color is defined in the table, the call would look like setPixel *(x, y, i*), where i is the entry address containing (r,g,b).

One more protocol is based on the existence of an existing color. This color is preserved by the system and can be set by calls that look like setColor (*rgb*) for direct coding or setColor (*i*) for lookup table representation. Now, the calls to set pixels need to provide only the co-ordinate information and would look like setPixel (*x, y*) for both image types. The graphical system will robotically use the most recently specified current color to carry out the operation.

Now, a lookup table's entries can be set by a call that looks like setEntry (*i, rgb*), which sets color (r, g, b) in the entry with address i. On the other hand, values in the lookup table can be interpreted back to the application with a call that looks like getEntry (*i, rgb*). This returns the color value in entry i array rgb.

Most of the times, a situation may arise, where two versions of the calls will specify the RGB values. One of these versions takes RGB values as floating point numbers in the range of [0.0, 1, 0], and the other version takes RGB values as integers in the range of [0, 225]. Even though, the floating point values are mapped by the graphics system into integer values before being written into the frame buffer.

To provide fundamental support for pixel based image processing, there are calls that look like getPixel *(x, y, rgb*) for direct coding, or getPixel *(x, y, i*) for the lookup table representation, in order to return the color or index value of the pixel at (x, y) back to the application.

There are few calls that read and write rectangular blocks of pixels. A useful example would be a call to set all pixels to a definite background color and then make a call that looks like clear() in order to achieve the goal.

2.3 Mandelbrot Set

The Mandelbrot set is a meticulous mathematical set of points, whose border generates a distinctive and easily recognizable two-dimensional fractal shape.

Did you know? The Mandelbrot set is named after Benoit Mandelbrot, and it is called as Apfelmännchen (apple-man) in German language.

Many programs are used to create the Mandelbrot set and other fractals. These programs use a range of algorithms to conclude the color of individual pixels and achieve well-organized computation.

One of the uncomplicated algorithms to create an illustration or representation of the Mandelbrot set is the Escape time algorithm. In the plot area, for each x, y point, a repeated and continuous calculation is carried out. In addition, the escape time algorithm is based on the behavior of that calculation, and a color is chosen for that pixel.

The x and y positions of each point are used as starting or initial values in an iterating or repeating calculation. The outcome of every repeated calculation is used as the starting values for the next value. The values are checked for all iterations, in order to know whether they have reached a critical escape condition or a bailout situation. If that specific condition is reached, then the calculation is stopped, the pixel is drawn, and the next x, y point is checked. For a number of starting values, the escape occurs swiftly, only after a small number of iterations. For starting values, which are very close to each other, but not in the set, possibly will take hundreds or thousands of iterations to escape.

Notes

Escape never occurs for the values within the Mandelbrot set.

Caution

Only the programmer or the user must choose how much iteration or depth is required in order to examine iterations. The higher the maximum number of iterations, the more detailed information emerges in the final image. However, it will take longer time to calculate the fractal image

It is important to know that the escape conditions can be easy or difficult. The reason is that no number with a real or imaginary part greater than 2 can be a part of the set. The main objective of a general bailout is to escape, when any of the co-efficients goes beyond 2.

The swiftness of values reaching the escape point is represented by the color of each point. Regularly the color black is used to explain values that fail to escape before the iteration limit. In addition, the brighter colors gradually are used for points that escape. This provides a visual demonstration of how many cycles were necessary before reaching the escape condition.

The escape time algorithm is popular for its simple calculations. However, it creates groups of colors, which can detract from an image's artistic value. This can be enhanced with the help of an algorithm known as Normalized Iteration Count. The normal iteration count provides a certain level of transition of colors between iterations. The algorithm connects a real number v with each value of z, by using the connection of the iteration number with the potential function. This function is given by the following formula:

$$\phi(z) = \lim_{n \to \infty} \left(\frac{\log|z_n|}{P^n} \right)$$

In this formula, z_n is the value after n iterations and P is the power for which, z is raised to the Mandelbrot set equation (z_n+1 = z_nP + c, P is generally 2).

If a large bailout radius N (Ex: 10100) is chosen, we have that

$$\frac{\log|z_n|}{P^n} = \frac{\log(N)}{P^{v(\approx)}}$$

for some real number v(z), and this is

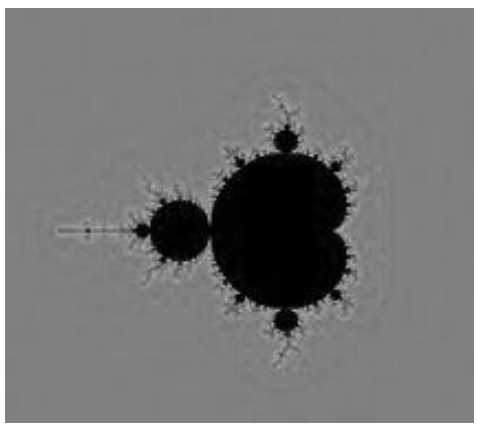
$$v(z) = n - \log_p \left(\frac{\log|z_n|}{\log(N)} \right)$$

and as n is the first iteration number such that $|z_n| > N$, the number subtracted from n is in the interval [0, 1].

For the purpose of coloring, a cyclic scale of colors and containing H colors numbered from 0 to H− 1 (for instance H = 500) is a must. The real number v(z) is multiplied by a fixed real number, determining the density of the colors in the picture, and integral part of this number modulo H is taken.

The following figure 2.4 depicts the Mandelbrot bug.

Figure 2.4: Mandelbrot Set



The figure 2.4 visualizes an area where -2 < *z*, real < 0.5 -1.25 < *z*, imag < 1.25 with *N* = 64. Most of the *z* values, which are outside the area lead *x* to diverge quickly, whereas the *z* values in the black region belong to the Mandelbrot set. As illustrated in Figure 2.4, the area is similar to the shape of a bug. You can see the most dynamic alterations between divergence and non-divergence along with the most significant variations in the number of iterations used in the divergence test. The brighter the picture, the longer it will take to conclude divergence for the equivalent *z*. In principle, the rectangular area can be decreased indefinitely to zoom in on any active region to illustrate more intricate details.

2.4 Julia Set

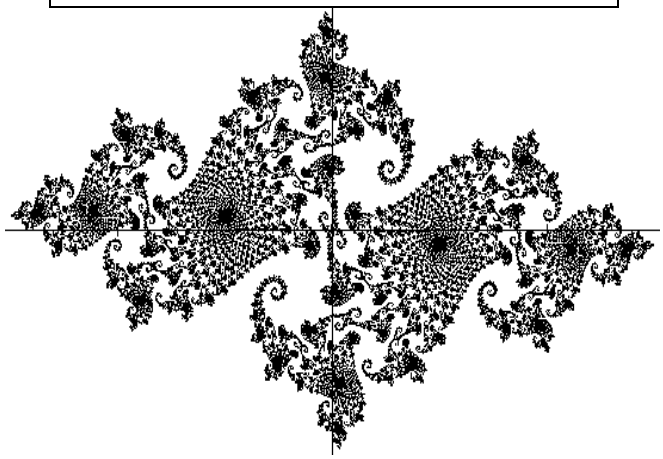
The Julia set is similar to the Mandelbrot set. If **z** is set to a fixed non-zero value and varies x_0 across the complex plane, a set of non-divergence numbers (values of x_0 that do not diverge under the given transformation) is obtained to form a Julia set.

Did you know? Julia sets are named after the famous French mathematician Gaston Julia. He was a French mathematician, who discovered Julia sets and was the first mathematician to explore their properties. His masterpiece on Julia set was published in 1918.

It is important to know that different z values lead to different Julia sets. The following image 1.8 is produced by making slight modifications to the pseudo code for the Mandelbrot set. The figure 2.5 shows the Julia set defined by z = -0.74543 + 0.11301 i with -1.2 < x0 , real < 1.2, -1.2 < x0 , imag < 1.2, and N = 128.

The following figure 2.5 depicts the Julia set.

Figure 2.5: Julia Set



Task

Research on web and gather information on the Julia set crop circle.

Knowing that the Julia sets are just a different parameterization of the same basic thing, you would expect there to be a deeper relationship between the Mandelbrot and Julia sets.

2.5 Summary

- Creation of a wide variety of colors starting from a small collection of primary colors is done by color model as it is an organized system.
- The RGB color model uses three primary colors, namely, Red (R), Green (G), and Blue (B) and it is also called as additive color model.
- CMYK is mainly used in color print production and it is a subtractive color model.
- In computer graphics, the most primal process is the setting of color attributes of individual pixels.
- The Mandelbrot set is a meticulous mathematical set of points, whose border generates a distinctive and easily recognizable two-dimensional fractal shape.
- The Julia set is similar to the Mandelbrot set.

2.6 Keywords

***Attribute*:** A property, quality or feature that differentiates an object or individual from other object or individual.

Fractal: A fragmented or rough geometric shape, which can be divided into parts.

Frame Buffer: Frame buffer is a video output device that drives a video display from a memory buffer containing a complete frame of data.

Fractal Image: Images created using fractal-generating software.

2.7 Self Assessment

1. State whether the following statements are true or false:
   (a) The RGB color models are those color models, which use printing inks for displaying colors.
   (b) In computer graphics, the most primal process is the setting of color attributes of individual pixels.
   (c) One of the complicated algorithms to create an illustration or representation of the Mandelbrot set is the Escape time algorithm.
2. Fill in the blanks:
   (a) The …… color models use printing inks for displaying colors.
   (b) RGB colors are also called as……. as they combine to create white color.
   (c) The …….. are named after the famous French mathematician Gaston Julia.
3. Select the suitable choice for every question:
   (a) The main objective of a general bailout is to escape, when any of the co-efficient goes beyond
      (i) 4
      (ii) 3
      (iii) 2
      (iv) 5
   (b) In a single protocol, the application simultaneously offers both.
      (i) Color information and co-ordinate information
      (ii) Protocol information n and color information
      (iii) Co-ordinate information and protocol information
      (iv) Protocol information and fragment information

2.8 Review Questions

1. "Julia set is a kind of mirror image of the Mandelbrot set". Discuss.
2. "In Mandelbrot set, it is important to know that the escape conditions can be easy or difficult". Explain.
3. "RGB colors are also called as additive colors". Explain.
4. "CMYK color model is called as subtractive model". Describe.

Answers: Self Assessment

1. (a) False
   (b) True
   (c) False
2. (a) Subtractive Color Models
   (b) Additive colors
   (c) Julia sets
3. (a) 2
   (b) Color information and co-ordinate information

2.9 Further Readings

Books

McConnell J., Computer graphics: Theory into practice.

Online link

http://mathworld.wolfram.com/MandelbrotSet.html
http://mathworld.wolfram.com/JuliaSet.html

Unit 3: Overview of Graphics I/O Devices

CONTENTS

Objectives
Introduction
3.1 Input Devices
3.1.1 Interactive Devices
3.2 Output Devices
3.2.1 Display Devices
3.2.2 Random Scan Display
3.2.3 Raster Scan Display Systems
3.3 Printers and Projectors
3.3.1 Printers
3.3.2 Projectors
3.4 Image Persistence
3.5 Resolution
3.5.1 Underscan and Overscan
3.5.2 Aspect Ratio
3.5.3 Measuring Color-depth and Computer Memory
3.5.4 Refresh Rate
3.5.5 The Megapixel Evolution
3.6 Frame Buffer
3.6.1 Types of Frame Buffers
3.7 Logical Functioning of I/O Devices
3.8 Summary
3.9 Keywords
3.10 Self Assessment
3.11 Review Questions
3.12 Further Readings

Objectives

After studying this unit, you will be able to:

- Explain input devices
- Describe interactive devices
- Comprehend output devices
- Define random scan displays
- Comprehend raster scan displays
- Discuss image persistence

- Define resolution
- Explain frame buffer
- Describe logical functioning of I/O devices

Introduction

Computer graphics is a field of computer science that deals with the study of generation of graphic images, such as line drawings and realistic depiction of natural objects. Production of these images requires both software and hardware. To generate and display graphic images various special purpose hardware are used. This unit gives an overview of graphic input and output devices, and the logical functioning of these devices.

The Input/Output devices play a major role to enable communication between the computer systems and the external world. Both the input and output devices include certain properties that are required for computer graphics to function effectively.

A large majority of computer graphics systems utilize some type of Cathode-Ray Tube (CRT) display and most of the fundamental display concepts are substantiated in CRT display technology. The three most common types of CRT display technologies are:

1. Direct view storage tube display
2. Calligraphic refresh display
3. Raster scan refresh display

The direct view storage tube display and calligraphic refresh display are line drawing displays, while the third is a point-plotting device. Few of the displays used in computer graphics are discussed in this unit.

3.1 Input Devices

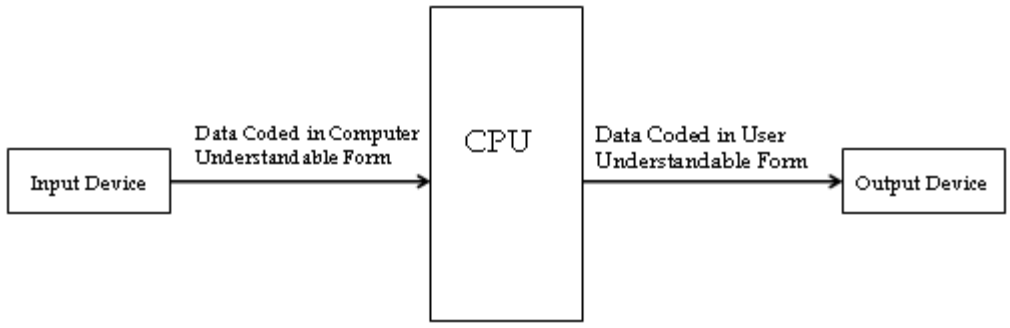
An input device refers to hardware devices that are used to send data to the computer. Therefore, we can say that input devices help in interacting with the computer.

3.1.1 Interactive Devices

The basic function of computer is to accept inputs from the user and process it to obtain the necessary results, depending on a sequence of instructions. According to computer terminology, devices that are capable of providing input and receiving output to and from the computer are known as hardware. An input device receives information from the user and translates it into a computer understandable language. That is, into a language that the CPU can understand. The information is processed in the CPU and the output is transferred to the output device in a user understandable language.

The figure 3.1 illustrates the communication between an input device, CPU, and output device.

Figure 3.1: Communication between Input Device, CPU and Output Device



The keyboard and mouse are the common interactive devices that help in input operation. Apart from the keyboard and mouse, there are many interactive devices that help in the input operation, such as the tablet, light pen, joystick, control dial, button, data glove, and touch screen.

The display terminals generally provide an alphanumeric keyboard to type commands and enter data for the program. However, in some applications the use of keyboard is inconvenient or inadequate. This kind of a situation may arise when the user has to add lines or symbols to the pictures on the screen. Even though the user can position the lines or symbols using coordinates, they can also perform this task by pointing at the screen.

The graphical interactions happen either by **positioning the items** or by **pointing at items.** There are a number of graphical input devices that are classified into pointing devices and positioning devices. Pointing devices refer to devices that are used to select items on the screen, and positioning devices refers to devices that are used to add new items on the screen. These devices provide position information, that is, they provide the coordinate of a point.

Notes

Pointing devices are also known as selectors, whereas positioning devices are known as locators.

Some of the pointing and positioning devices are discussed as follows:

1. ***Graphics Tablet:*** The graphics tablet is the most widely used device for pointing oriented operations. It is a flat, touch-sensitive device that is used with a stylus (stylus is used to indicate a location on the tablet surface). Anything that is written or drawn on this device is transferred automatically to the PC in the form of a graphic. Typically tablets provide two-dimensional coordinate information. The values returned are in tablet coordinates. This device provides an accurate method of selecting the positions of the coordinates. There are different principles used to implement tablets. Some of them are, using wires, sound waves and magnetic principles.

Notes

Graphic tablets are also known as digitizing tablets.

2. ***Joystick:*** A joystick is another locator device that controls the movement of the cursor. It provides three types of controls, such as digital, direct and glide. The digital control allows movement in limited directions, that is, up, down, left and right. The direct and glide control allows movement in every direction, that is, 360 degrees.

3. ***Trackball:*** The trackball functions similar to the joystick. This device comprises a ball that can be rotated in any direction using figures or palm. When the ball is moved the cursor moves accordingly. A potentiometer is attached to the ball to measure the direction of rotation. It can be used as a substitute to the mouse.
4. ***Mouse:*** Mouse is the most important input device which consists of an upside-down trackball mounted in a small, lightweight box. As you move the mouse across a surface, the ball rotates and drives the shafts of two valuators, that is, either potentiometers or digital shaft encoders. The movement of the shafts provide (x, y) coordinates. The Optical mouse and mouse that is based on magnetic principles use small light source and a small photo electric cell that produces light pulses, which are counted and converted into (x, y) coordinates.
5. ***Light Pen:*** The light pens are pencil shaped devices used to select screen positions by detecting the light coming from the screen. It consists of a light sensitive diode to sense the light from the screen. These screens are sensitive to only short burst of light that gets emitted from the light pens. Other lights such as the background light in a room are not detected by the light pen.

3.2 Output Devices

Output devices play a major role in a computer system. These devices are used to display or communicate the information that is processed by the computers. The output devices connected to the computer takes various forms, such as display, print, graphics or audio. Even though there are numerous output devices, only a few of them are related to computer graphics.

3.2.1 Display Devices

In terms of computer graphics display devices are output devices that help to display graphics. The graphical outputs can be of two forms:

1. ***Hard Copy:*** Hard copy refers to the physical form of output that is recorded from a computer onto a paper or some other durable surface, such as a microfilm. Output in the form of hard copy is permanent and stable. Generally, paper is the widely used form of hardcopy.
2. ***Soft Copy:*** Soft copy refers to the electronic version of an output, which resides in computer memory or on disk, and is displayed on a screen. It is not a permanent form of output as the hard copy. Outputs in the form of soft copy can be in audio visual or graphical form.

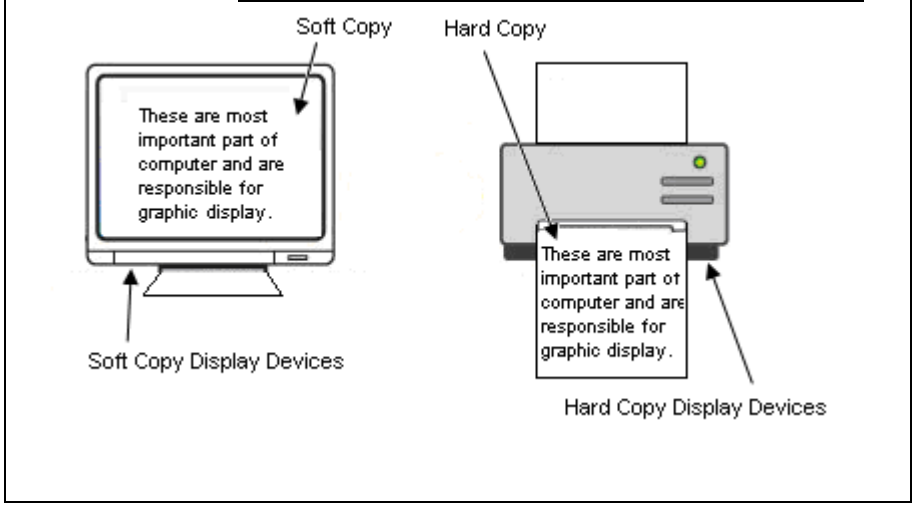
Depending on the hard copy and soft copy outputs, the display devices are classified into two types:

1. Hard copy display devices
2. Soft copy display devices

Both the hard copy and the soft copy display devices are capable of converting the electrical signals of a computer into visible images at higher speeds.

Figure 3.2 displays a soft copy from a soft copy display device (monitor), and hard copy from a hard copy display device (printer).

Figure 3.2: Types of Outputs



Source: Computer Graphics, Bhatia, K, I.K. International, Second Edition, Chapter 2-Interactive Devices, Page 10.

Printer and plotters are the most widely used hard copy display devices. The graphics stored on a computer's disk drive are printed using printers.

Monitors and flat panel display devices are the most commonly used soft copy display devices.

Monitors are used commonly as output device for computer graphics. Any monitor performs the basic function, which is to allow the users to interact with the computer through sight and the use of a Graphical User Interface (GUI).

3.2.2 Random Scan Display

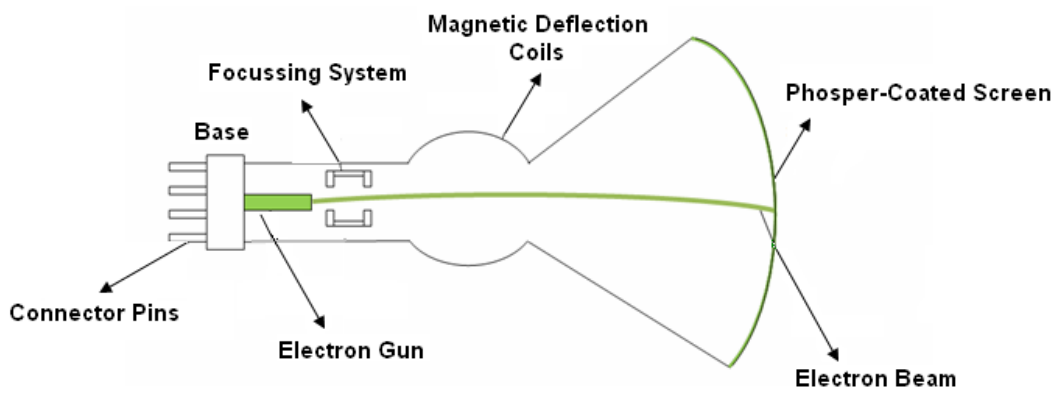
Random scan displays refers to direct view storage tube display and calligraphic display. Random scan displays are also known as line drawing displays.

Cathode Ray Tube (CRT) Model Technique

The most commonly used display device is the Cathode Ray Tube (CRT). In this display device, when an electronic beam falls on phosphorescent surface then it produces a beam of light. The electronic or magnetic fields can be used to focus the beam of light to a single point on the screen.

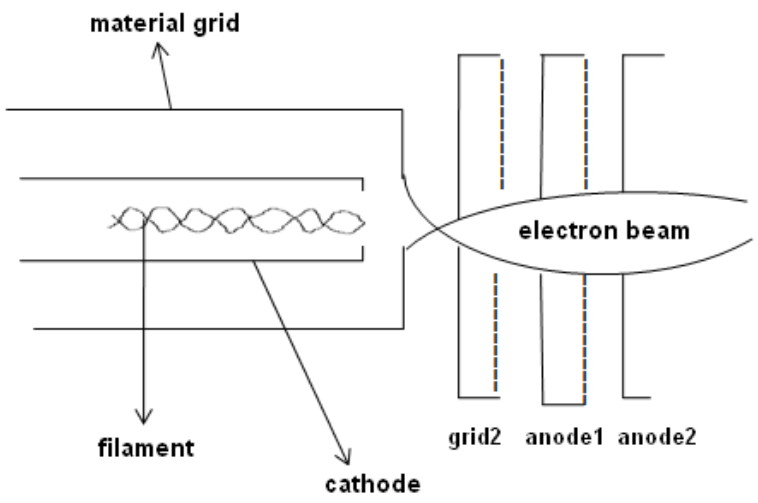
Figure 3.3 illustrates how a Cathode Ray Tube works.

Figure 3.3: Cathode Ray Tube



In figure 3.3, the conical glass tube is used for CRT tube. An electronic gun is placed at the narrow end. The electrons emitted from the gun are passed through the magnetic system called yoke. The electronic beam falls on the broader surface of the CRT tube. The broader surface of CRT tube is single coated with phosphorus which reflects the beam to make it fall on computer screen.

Figure 3.4: Focusing Grids



In figure 3.4, we see that the focusing grids such as one horizontal and another vertical help in focusing the beam on to the screen. The deflection of the beam depends on the direction (positive or negative) and the intensity of the fields. The suitable combinations of grids are used to focus the beam on to the screen. This helps to illuminate any point on the screen which in turn helps to illuminate any graphic picture, thereby making it easy to trace the graphic picture.

Direct View Storage Tube Display

Of all the CRT (Cathode Ray Tube) display technologies in use today, the Direct View Storage Tube (DVST) display is the simplest. The DVST is a CRT with a long persistence phosphor. The duration of visibility of the line of character is nearly up to an hour or till the time it is erased. The intensity of the CRT's electron beam is sufficiently increased in order to draw a line on the display. This affects the phosphor to get its bright state. To enable phosphor to get into the dark state, the entire tube has to be flooded with a specific voltage for half a second.

When the entire tube is flooded, all the lines and characters are also erased. This is one of the limitations for animation using this display. This method is not useful for interactive drawings. However, this display has many advantages which include a flicker free display, capability of displaying any number of lines and is easier to program.

Calligraphic Refresh Display

Calligraphic refresh display refers to a random scan display or a line drawing display. It uses very short persistence phosphor. Due to the short persistence, the picture has to be redrawn on the CRT a number of times per second. This redrawing on the CRT is called refreshing the CRT.

Caution The rate of refresh should not be less than 30 times per second. Otherwise a flickering image will be seen, which is quite irritating to the viewer.

In addition to the CRT, this display requires two more elements called the display buffer and the display controller. The display buffer is contiguous memory containing all the information required to draw the picture on the CRT. The display controller takes this information and gives it to the CRT to display at the refresh rate. The number of lines that are displayed depends on the size of the display buffer and the speed of the display controller.

The advantages of DVST over CRT are:

1. Refreshing is not required.
2. Displaying very complex pictures at very high resolution without flicker.

The disadvantages of DVST are:

1. Generally, it does not display color.
2. It is not possible to erase a selected part of the image.
3. It consumes several seconds to erase and redraw process for complex images.

3.2.3 Raster Scan Display Systems

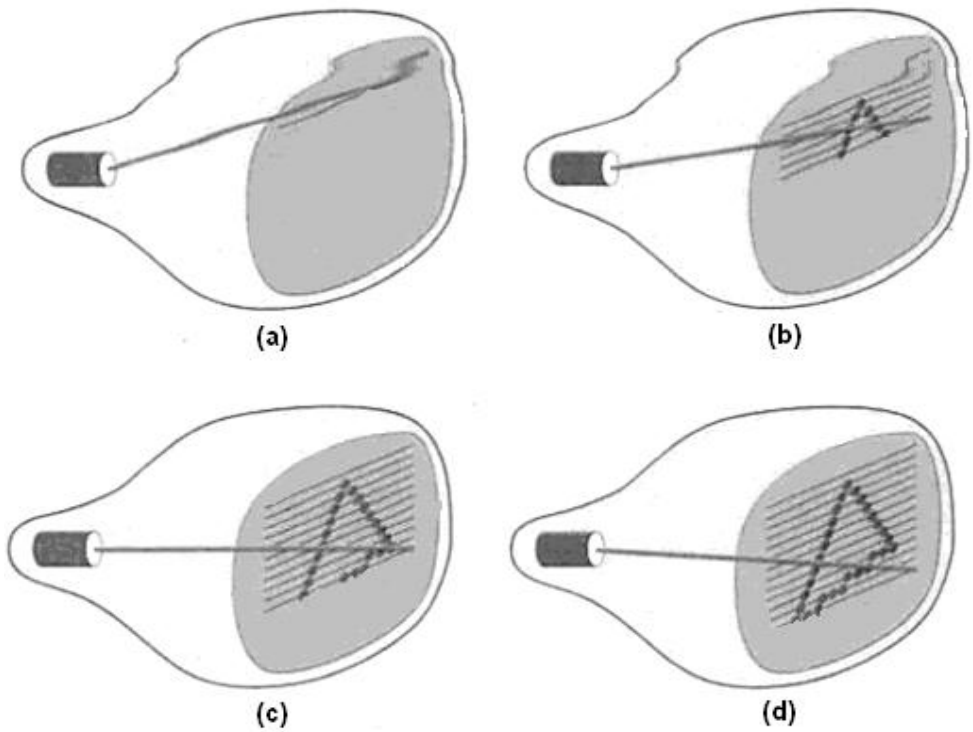
A raster scan display system is the rectangular pattern of image capture and reconstruction in televisions. Most raster displays have some specialized hardware to assist in scan converting output primitives into the pixmap, and to perform the raster operations, such as moving, copying, and modifying pixels or blocks of pixels. This hardware is called as a graphics display processor. The fundamental difference among display systems is how much the display processor does against how much must be done by the graphics subroutine package executing on the general-purpose CPU that drives the raster display.

Notes A graphics controller refers to the graphics display processor (emphasizing its similarity to the control units for other peripheral devices).

The relationship between the pixmap and the address space of the general-purpose computer's memory is another major differentiator in raster systems, whether the pixmap is part of the general-purpose computer's memory or separate.

The raster-scan display is based on a television technology. It is the most common type of graphics monitor which employs a CRT. An electron beam is swept across the screen in a raster-scan system, one row at a time from top to bottom. The beam intensity is turned on and off to create a pattern of illuminated spots as the electron beam moves across each row. The picture definition is stored in a memory called the refresh buffer or frame buffer. A set of intensity values hold this memory area for all the screen points. Then, the stored intensity values are retrieved from the refresh buffer and painted on the screen one row at a time. The figure 3.5 displays a raster-scan system that displays an object as a set of discrete points across each scan line.

Figure 3.5: A Raster-scan System



Source: Computer Graphics, Donald Hearn, M. Pauline Baker, C version, Second Edition., Chapter 2-Overview of Graphics Systems, Page 61.

A raster-scan system can store the intensity information for each screen point that makes it well suited for the realistic display of scenes containing subtle shading and color patterns. A screen point is referred to as a pixel or pel. Other examples of systems using raster-scan methods are TVs and printers.

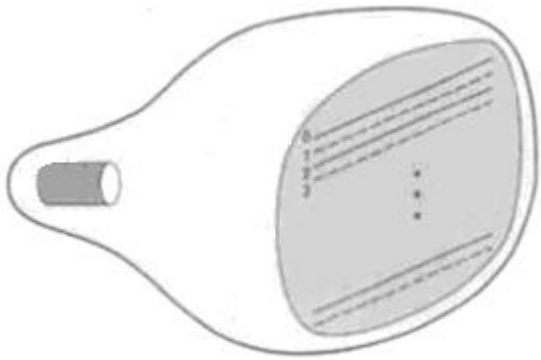
The capability of the raster system depends on the intensity range for pixel positions. In a black-and-white system, every screen point is either on or off. Therefore, only one bit per pixel is required to control the intensity of screen positions. For a bi-level system, a bit value of 1 indicates that the electron beam is to be turned on at that position, and a value of 0 indicates that the beam intensity is to be turned off. In addition, bits are required when color and intensity variations can be displayed. Up to 24 bits per pixel are included in high-quality systems, which can require several megabytes of storage for the frame buffer, depending on the resolution of the system. One 3 MB (megabytes) of storage for the frame buffer is required for a system with 24 bits per pixel and a screen resolution of 1024 x 1024. The frame buffer is commonly called a bitmap for a black-and-white system with 1 bit per pixel, and the frame buffer is often referred to as a pixmap for systems with multiple bits per pixel.

While some systems are designed for higher refresh rates, the refreshing on raster-scan displays are carried out at the rate of 60 to 80 frames per second. The refresh rates are described in units of cycles per

second, or Hertz (Hz), where a cycle corresponds to one frame. Using these units, a refresh rate of 60 frames per second is described as 60 Hz. The electron beam returns to the left side of the screen to begin displaying the next scan line. The return to the left of the screen, after refreshing each scan line, is called the horizontal retrace of the electron beam. At the end of each frame, the electron beam returns to the top left corner of the screen to begin the next frame.

On some raster-scan systems (and in TV sets), each frame is displayed in two passes using an interlaced refresh procedure. In the first pass, the beam sweeps across every other scan line from top to bottom. After the vertical retrace, the beam sweeps out the remaining scan lines. The figure 3.6 displays interlacing scan lines on a raster-scan display.

Figure 3.6: Interlacing Scan Lines on a Raster-scan Display



***Source*:** Computer Graphics, Donald Hearn, M. Pauline Baker, C version, Second Edition., Chapter 2-Overview of Graphics Systems, Page 62

As depicted in figure 3.6 first, all points on the even numbered (solid) scan lines are displayed, and then all points along the odd numbered lines (dashed) are displayed. Interlacing of the scan lines in this way allows us to see the entire screen displayed in one-half the time it would have taken to sweep across all the lines at once from top to bottom.

Interlacing is primarily used with slower refreshing rates. On an older, 30 frame-per-second, non-interlaced display, for instance, some flicker is noticeable. However, with interlacing, each of the two passes can be accomplished in 1/60th of a second, which brings the refresh rate nearer to 60 frames per second. This is an effective technique for avoiding flicker, proving that adjacent scan lines contain similar display information.

3.3 Printers and Projectors

Printers and projectors are output devices that are helpful in the display of graphics.

3.3.1 Printers

A printer is an important output device in graphics. The printer allows printing the information and images displayed on the monitor. Printers can be classified depending on their printing methodology, such as impact printers and non-impact printers. A line printer and dot matrix printers are the examples of impact printers. An ink- jet printer and laser printer are the examples of non impact printers.

There are various modern printers that make use of new printing technologies. Some of them are as follows:

1. ***Toner-based Printer:*** A laser printer is a toner-based printer. It produces high quality text and graphics rapidly. The laser printer employs a xerographic printing process. However, it differs from analog photocopiers. That is, in a laser printer the image is produced by the direct scanning of a laser beam across the printer's photoreceptor.

> *Notes* The LED is a toner-based printer that uses an array of LEDs instead of a laser to enable toner adhesion to the print drum.

2. ***Liquid Inkjet Printers:*** A liquid inkjet printer is most used printer that is operated by propelling different-sized droplets of liquid or molten material (ink) onto a page of any size.
3. ***Solid Ink Printers*****:** Solid ink printers, which are also known as phase-change printers, are a type of thermal transfer printer. They make use of solid sticks of CMYK-colored ink which are similar to the consistency of candle wax. These sticks are melted and fed into a piezo crystal operated print-head. The print-head sprays the ink on a rotating oil coated drum.
4. ***Dye-sublimation Printers*****:** A dye-sublimation printer (or dye-sub printer) is a printer which employs a printing process that uses heat to transfer dye to a medium, such as a plastic card, paper or canvas. The process is to lay one color at a time using a ribbon that has color panels. Dye-sub printers are intended primarily for high-quality color applications, including color photography, and are less well-suited for text.

> *Notes* Dye-sublimation printers are generally used in consumer photo printers.

5. ***Inkless Printers:*** Thermal printers work by selectively heating regions of special heat-sensitive paper. Monochrome thermal printers are used in cash registers, ATMs, gasoline dispensers and some older inexpensive fax machines. Colors are produced on special papers, with different temperatures and heating rates for different colors. One example is the ZINK technology.

> *Did you know?* ZINK technology is used to print pictures on special papers without using ink.

6. ***UV Printers:*** In 2007 the UV printers were still in the development phase. This type of a printer would make use of a special UV light bar that would write and erase the paper. However, it has to be noted that the text on the printed pages lasts only for 16 to 24 hours before fading.

> *Did you know?* Xerox Corporation is developing an inkless printer which will make use of a special reusable paper, which is coated with micrometers of UV light sensitive chemicals.

3.3.2 Projectors

Projectors are used to produce soft copy output. They are commonly used in classroom training or in conference rooms for presentation. Projectors are mainly of two types:

1. ***LCD Projectors*:** LCD is the acronym for Liquid Crystal Display. It is the established technology used by most of the top manufacturers. Most of today's LCD projectors contain three separate LCD glass panels, one each for red, green, and blue components of the image signal being fed into the projector. As light passes through the LCD panels, individual pixels can be opened to allow light to pass through, or closed to block the light. This activity modulates the light and produces the image that is projected onto the screen.
2. ***DLP Projectors*:** DLP is the acronym for Digital Light Processing. A single Digital Mirror Device (DMD) chip is used in DLP that has thousands of tiny mirrors, each representing a signal pixel. Video images are handled extremely well by DLP projectors.

Did you know? Texas Instruments Inc. (an American company based in Dallas, Texas, United States, well known for developing and commercializing semiconductor and computer technology) developed DLP projectors.

3.4 Image Persistence

Persistence is the amount of time taken by the emitted light from the screen to break down to its original intensity by one-tenth times. When the LCD monitors are run continuously for long time with a fixed image, a trace of electric charge is created near the electrode in the LCD module. This results in a ghost image of the previous image when the image is changed. This is known as image persistence.

Image persistence is not permanent. However, when a particular image is displayed for a long period of time, ionic impurities accumulate inside the LCD and are seen in the image that is displayed on the LCD screen. However, this may some times result in the retention of the image on the screen.

Example: In an LCD monitor the objects that can generate a persistent image are the desktop icons, task bar and background image. This is because these elements are static and are displayed on the screen for a long period of time. When other images are loaded on these locations a faint outline or image of the previous image can be seen.

Image persistence can be avoided by turning off the screen after a few minutes of idle time, or by using a screen saver with movements. This helps to avoid the display of a particular image for a long time.

3.5 Resolution

Resolution refers to the sharpness and clarity of an image. Resolution is mostly used to describe monitors, printers, and bit-mapped graphic images. Resolution indicates the maximum number of points that may be displayed without overlap on the CRT. In case of dot-matrix and laser printers, the term resolution indicates the number of dots per inch. Resolution is defined as the number of points per centimeter that may be plotted horizontally and vertically. Resolution depends on the phosphor type, the displaying intensity and the focusing and deflection systems used in the CRT.

The display resolution of a digital television or display device is the number of different pixels in each dimension that may be displayed. It may be an indefinite term as the displayed resolution is controlled by all different factors in CRT and flat panel or projection displays using fixed picture-element (pixel) arrays.

Example: A 300-dpi printer is one that can print 300 distinct dots in a line that is 1 inch long. This means that it can print 90,000 dots per square inch.

For graphics monitors, the screen resolution indicates the number of dots on the entire screen.

Example: A 640-by-480 pixel screen displays 640 different dots on all 480 lines or about 307,200 pixels.

This translates into distinct dpi measurements depending on the size of the screen.

Example: A 15-inch VGA monitor displays 50 dots per inch.

Did you know? Printers, monitors, scanners, and other input output devices are classified into high resolution, medium resolution, and low resolution.

3.5.1 Underscan and Overscan

The underscan mode displays the full video frame, which discloses the content on the recorded edges, but is not shown in the camera's flip-out LCD. The overscan mode zooms on the field monitor into the area that is visible on most of the televisions. The field monitor must be set to underscan mode if the video is viewed on computer monitor or shown with a projector. The field monitor must be set to overscan mode to check how the video is displayed on television.

The broadcast industry defined overscan as the central part of the image that can be viewed on a standard television set. Underscan is the full frame, which is only seen on a production monitor. Hence, the underscan shows the picture more clearly than the overscan.

3.5.2 Aspect Ratio

The aspect ratio of the image is the ratio of width of the image to the height of the image. Usually aspect ratio is represented in two numbers separated by the colon.

Example: If the aspect ratio of the image is X:Y, then 'X' is the width of the image and 'Y' is the height of the image. Then, both the width and height are measured in the same units.

If height is measured in inches then width is also measured in inches. Then, the aspect ratio is mathematically expressed as X:Y or X x Y.

In televisions, DVDs and blue-ray convert the images of unequal sizes to the clear picture is done by enlarging the image to fill the receiving format's area and deleting excess picture information. Different factors of the image are changed and scaled through different factors to achieve the aspect ratio. Aspect ratios are mathematically expressed as X:Y (pronounced "X-to-Y") and X x Y (pronounced "X-by-Y").

3.5.3 Measuring Color-depth and Computer Memory

Color-depth is the main factor on which resolution depends.

The three alternatives of color depth are:

1. 8-bit (256 colors)
2. 16-bit (65 thousand of colors)
3. 24/32-bit (16, 8 million colors)

Computer memory is measured in terms of bytes. Depending on the computer being used, screen resolutions and color depth can be adjusted.

Graphics memory and resolution or color depth is interrelated. If the graphics memory of the computer is known then the resolution or color-depths displayed by a computer can be calculated.

Today, the most commonly used graphics memory is between 8 MB and 64 MB and resolutions are between 800 x 600 and 1024 x 768.

3.5.4 Refresh Rate

The term "refresh-rate" refers to the display screen which is being updated and refreshed. For a stable flicker-free picture at least 70 refreshes per second is recommended (For every "refresh" the picture on your monitor is re-drawn). A refresh rate of 50 updates per second gives you a more 'flickery' display. The "refresh-rate" is measured in Hz (Hertz). (1Hz = 1 time per sec) Horizontal Sweep Frequency however, refers to the amount of horizontal pixel-lines that the monitor can output per unit. For example, a resolution of 640 (width) x 480 (height) means that the screen consists of 480 horizontal lines that are 640 pixels wide each. The Horizontal Sweep Frequency (measured in kHz=kilohertz) tells you how many of these horizontal lines the monitor draws every second. This job is done by your monitor, therefore, even if you have a very expensive graphics card in your computer, it is still the monitor that sets the upper limit for the quality of your display.

3.5.5 The Megapixel Evolution

A pixel is the most basic unit of the image and it is single point of the raster image. Generally, the pixels are arranged in rows and columns. The different pixels are combined together to form an image. The pixels vary in brightness and color values. Three components are used to represent the color image and each component is called as sub pixel. The sub pixel represents the single color that can be red or green or blue. The information is represented in the bits. Bits per pixel are the number of bits of information reserved per pixel of an image. Usually the digital images are made up of many pixels. One million pixels constitute the megapixel. The pixel density is the number of pixels per unit area on the sensor. The different cameras have different pixel density. Both megapixel count and pixel density are related. The smallest megapixel count corresponds to a very high pixel density. The resolution of an image depends on the pixels.

Elements of picture quality

The megapixel is one aspect that plays an important role in the quality of the camera or the photo that it produces. In addition, the camera sensor and optical quality of a lens play an important role in the quality of an image.

Apart from the sensor and lens, other elements that determine the quality of photos are:

- Appropriate lighting on a subject.
- Proper focus
- Taking a photo at high resolution

Did you know? In the case of a 16 by 24 foot image and in such case, the megapixels do not detect the clarity of the image. It depends on the lens and sensor of the camera.

Image resolution is the number of pixels that are counted horizontally or vertically in describing the image.

Let us now understand basicity in megapixel evolution. Digital cameras play an important role in today's growing world. The digital photography helps in taking the images. These can be uploaded to the computer within no time.

A camera is a lightproof box that allows a certain amount of light to enter at the right time. A photographic film is present inside the camera. The image is formed when light enters the box. This is because a chemical reaction takes place on the photographic film. The camera flash is used to light up a scene that is too dim so that it is clear on the film. Earlier, flash bulbs were used to burn magnesium metal with a brilliant white light. However, modern cameras use electric currents that moves through xenon gas and causes the gas to glow light.

The photographic film is coated with silver halide solution that is light sensitive. Hence, when light falls a chemical change in the silver halide crystals occurs. This reaction helps in the formation of the latent

image (negative image). Once the negative is ready, it is processed and transferred onto a photographic paper. An optical apparatus, known as the enlarger, is used to project the image of the negative onto a base, and finely control the focus, amount, and duration of light incident on the paper. A sheet of photographic paper is then exposed to the enlarged image that projects from the negative.

During exposure, the dodging and burning techniques are used to adjust values of the image. The processes involved in these include reducing and increasing the amount of incident light selectively for part or all of the exposure time. After exposure, the photographic printing paper is ready to be processed. The photographic paper is processed using chemicals in the following order:

1. Developing the print using a photographic developer
2. Rinsing with stop-bath
3. Fixing the image permanently with the use of photographic fixer
4. Washing to remove all the processing chemicals
5. Drying

In this sub-section we discussed the evolution of megapixel. Let us now discuss what a frame buffer is and the types of frame buffer.

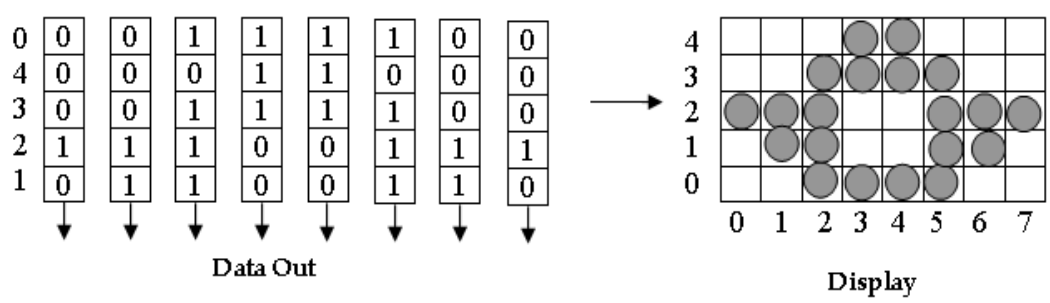
3.6 Frame Buffer

Frame buffer refers to the part of memory that is reserved for holding the entire bit-mapped image that is sent to the monitor. The frame buffer is typically stored in the memory chip on the video adapter. However, in some cases the frame buffer is stored in the main memory, and the video chipset is integrated into the motherboard design.

In a raster scan display system, the storage area is arranged as a two-dimensional table. Every row-column entry stores information such as brightness and/or color value of the corresponding pixel on the screen. In a frame buffer, 1 to 24 or more bits represent each pixel depending on the quality (resolution) of the display system and certain attributes of the pixel. A higher resolution gives better quality of the display or graphics.

Commands to plot a point or line are converted into intensity and color values of the pixel array or bitmap of an image by a process called Scan Conversion. Figure 3.7 depicts the frame buffer using eight 5-bit shift registers.

Figure 3.7: Frame Buffer Using Eight 5-bit Shift Registers



Source: Computer Graphics & Multimedia First Ediction-2009, Godse. A.P & Godse. D.A, Technical Publications, Chapter 1, Page 56

The refresh buffer in the display system, cycles row by row at speeds of 30 or 60 times per second to display the image. The digital/analog converter, which produces the necessary deflection signals to generate the raster scan, routes the intensity values picked up from the frame buffer. Interlacing produces a flicker-free image. First, it displays all odd-numbered scan lines from top to bottom and then

all even-numbered scan lines. The effective refresh rate to produce the picture becomes greater than 30 Hertz.

3.6.1 Types of Frame Buffers

Frame buffers use different kinds of memory. Initially, frame buffers made use of drums and disks with rotational frequency compatible to the rate of refresh. However, the rotating-memory frame buffers are now replaced by integrated-circuit shift registers due to lower costs.

A number of shift registers are used to construct a frame buffer. Each shift register uses one column of pixels on the screen. Every shift register contributes one bit per horizontal scan line. However, it is not very easy to change a given spot on the screen using shift registers. Therefore, they are suitable for applications that are interactive.

Random-scan integrated circuits are used in modern frame buffers where the pixel intensities are represented by 1, 2, 4, 8, 16 or 24 bits. Nearly, 8 bit per pixel is required for encoding text and simple images. However, to produce a good quality colored image 24 bits are required.

A color map is one of the best methods used to encode colored pictures. The pixel values in the frame buffer are considered as addresses of a look-up-table with entries for every pixel's red, green and blue components. The entry value is used to control the intensity or color on the CRT. Every color components can be defined to high precision offering accurate control over the colors displayed.

A multiple-plane frame buffer is another type of frame buffer where the frame buffer can be treated as consisting of several frames or planes. Each contains information (intensity and/or color) values of a separate image. An 8-bit per pixel frame buffer can be made to represent a single image with 8-bit of intensity precision, or it can represent two images, each of 4-bit intensity precision or eight black and white images with 1-bit intensity precision each. A variety of image mixing can be done.

Example: In animation systems, several moving objects can be displayed as separate planes.

3.7 Logical Functioning of I/O Devices

The I/O devices are capable of performing different logical functions. The functional capabilities of I/O devices are divided in four types. They are:

1. ***Locator Function:*** The locator function provides coordinating information in two or three dimensions. Basically, the coordinates are returned as normalized coordinates and may be either relative or absolute. The locator functions are performed by devices such as track ball, joystick, tablet, mouse, touch panel, and so on.

Example: The locator functions are performed by devices such as track ball, joystick, tablet, mouse, touch panel, and so on.

2. ***Valuator Function:*** The valuator function provides only a single value as a real number. This may be bound or unbound. A bound valuator has mechanical or programmed ends within a fixed range, whereas an unbounded valuator has an infinite range.
3. ***Choice or Button Function:*** The choice function selects and activates events or procedures which control the interactive flow or change the underlying task. It generally provides binary information.

Example: The keyboard is a specific example of collection of buttons or choice functions.

4. ***Pick Function*:** The pick function selects objects within the displayed picture. It picks up the selected objects.

Example: The pick functions are performed by devices such as a light pen.

3.8 Summary

- Input devices are used to interact with the computer.
- Interactive devices can be either a pointing device or a positioning device.
- Graphics tablet, joystick, trackball, mouse and light pen are some of the pointing and positioning devices.
- Output devices are used to display the processed information by the computer.
- The display devices provide graphical outputs in the form of hard copy or soft copy.
- The random scan displays are direct view storage tube display and calligraphic display.
- The raster scan display system refers to the rectangular pattern of image capture and reconstruction in televisions.
- Printers are used in printing the information and images displayed on the monitor.
- The various modern printers include toner-based printer, liquid ink-jet printer, solid ink printer, dye-sub limitation printer, inkless printers and UV printers.
- Projectors produce soft copy output. They are of two types: LCD projectors and DLP projectors.
- Image persistence refers to the ghost image that is formed when LCD monitors are run for a long time with a fixed image.
- Resolution is the sharpness and clarity of an image.
- Frame buffer is that part of the memory which is reserved to hold the entire bit-mapped image that is sent to the monitor.
- The I/O devices functioning capabilities are divided into four types, which include, locator function, valuator function, choice or button function and pick function.

3.9 Keywords

Graininess: A photo which has poor definition because of large grain size

KB: Kilo Bytes

MB: Mega Bytes

MFPs: Multifunction Printers

Mpx: Megapixel

One Byte: 8 bits

Overscan: The Field Monitor zooms into the area that will be visible on most televisions

Persistence: Image persistence method used to protect display from the burn-in problem

Resolution: The level of information on a display device, such as a monitor

Underscan: Displays the full video frame, which is visible only on a production monitor

3.10 Self Assessment

1. State whether the following statements are true or false:
   (a) Picture definition is stored in a memory area called the refresh buffer or frame buffer.
   (b) Each screen point is referred to as a pixel or pel.

(c) The keyboard is an example of Locator Function device.

(d) Interlacing of the scan lines allows us to see the entire screen displayed in one-half the time it would have taken to sweep across all the lines at once from top to bottom.

(e) In a raster scan display system, 1 to 24 or more bits represent each pixel depending on the quality of the display system and certain attributes of the pixel.

(f) The valuator function selects objects within the displayed picture.

2. Fill in the blanks:

(a) The..... devices refer to devices that are used to select items on the screen.

(b) A ..... refers to the electronic version of an output, which resides in computer memory or on disk, and is displayed on a screen.

(c) The ..... display is based on a television technology.

(d) The ..... printers are also known as phase-change printers.

(e) When a particular image is displayed for a long period of time, ... accumulate inside the LCD and are seen in the image that is displayed on the LCD screen.

3. Select a suitable choice in every question:

(a) Which of the following is an input device that can be rotated in any direction by fingers or palm?

(i) Joystick

(ii) Trackball

(iii) Mouse

(iv) Light pen

(b) Which of the following is a CRT with a long persistence phosphor?

(i) Random Scan display

(ii) Calligraphic Refresh Display

(iii) Raster Scan Display

(iv) Direct View Storage Tube Display

(c) Which of the following is a printer that employs a printing process that uses heat to transfer dye to a medium, such as a plastic card, paper or canvas?

(i) Dye-sublimation printer

(ii) Liquid Inkjet Printers

(iii) UV printers

(iv) Inkless printers

(d) The term ……. refers to the display screen which is being updated and refreshed.

(i) Aspect ratio

(ii) Underscan

(iii) Refresh-rate

(iv) Overscan

(e) Which of the following functions provide only a single value as a real number?

(i) Valuator function

(ii) Choice or button function

(iii) Pick function

(iv) Locator function

3.11 Review Questions

1. "The graphical interactions happen either by positioning the items or by pointing at items." Discuss.
2. "The graphical outputs can be of two forms." Justify.
3. "Random scan displays refer to direct view storage tube display and calligraphic display." Explain.
4. "A raster scan display system is the rectangular pattern of image capture and reconstruction in televisions". Explain.
5. "There are various modern printers that make use of new printing technologies." Justify.
6. Analyze the different types of projectors.
7. "When the LCD monitors are run continuously for long time with a fixed image, a trace of electric charge is created near the electrode in the LCD module." Explain.
8. "Resolution refers to the sharpness and clarity of an image." Discuss.
9. "The term "refresh-rate" refers to the display screen which is being updated and refreshed". Explain.
10. "The aspect ratio of the image is defined as the ratio of the width of the image to its height." Discuss.
11. "Frame buffer refers to the part of memory that is reserved for holding the entire bit-mapped image that is sent to the monitor." Discuss.
12. "The functional capabilities of I/O devices are divided into four types." Justify.

Answers: Self Assessment

1. (a) True (b) True (c) False (d) True (e) False (f) False
2. (a) Pointing (b) Soft copy (c) Raster-scan (d) Solid ink (e) Ionic impurities
3. (a) Trackball (b) Direct View Storage Tube Display
   (c) Dye-sublimation printer (d) Refresh-rate (e) Valuator function

3.12 Further Readings

Books

Bhatia, K, I.K., Computer Graphics, International, Second Edition
Hearn, Donald, & Pauline, M. Baker (2008). Computer Graphic, C version, 2nd Edition
Godse, A.P, (2009), Computer Graphics, Technical Publications
Ziang, & Roy, Avadhani, P.S., (2006). Computer Graphics, The McGraw-Hill Companies, 2nd Edition

Online link

http://www.unm.edu/~tbeach/terms/inputoutput.html
http://www.ehow.com/list_7184354_output-devices-computer-graphics.html
http://www.britannica.com/EBchecked/topic/288883/inputoutput-device
www.cse.iitm.ac.in
http://www.computerhope.com/jargon/o/outputde.htm

Unit 4: Scan Conversion I

CONTENTS

Objectives

Introduction

4.1 Methods – Analog and Digital

4.2 Line

4.3 Circle

4.4 Ellipse

4.5 Arcs and Sectors

4.6 Summary

4.7 Keywords

4.8 Self Assessment

4.9 Review Questions

4.10 Further Readings

Objectives

After studying this unit, you will be able to:

- Describe different methods of scan conversion
- Explain scan conversion of line, circle, ellipse, arcs and sectors

Introduction

An object is displayed on the computer screen after deciding as to which pixel of the screen needs to be highlighted. Various patterns are obtained on the computer screen by highlighting the required pixels. Most of the operators normally think in terms of more complex graphic objects such as cubes, cones, circles, and ellipses. Several algorithms have been developed to generate such complex graphic objects.

A computer may use different algorithms to produce raster images on raster devices just by turning the appropriate pixels ON or OFF. The process of representing continuous graphics objects as a collection of discrete pixels is known as scan conversion or rasterization. Scan conversion or scan rate converting is the process of altering the vertical or horizontal scan frequency of video signal for different purposes and applications. The instrument used for scan conversion is known as scan converter.

Did you know? Scan conversion acts as a bridge between TV and computer graphics technology.

This unit examines the procedures followed in drawing complex objects. It also discusses the various line drawing algorithms like Bresenham line drawing and circle drawing algorithm. This unit also explains the mathematical and algorithmic aspects of scan conversion.

4.1 Methods-Analog and Digital

The process of scan conversion involves altering the picture's data rate information and then inserting the new picture within appropriate synchronization signals.

Two methods for changing a picture's data rate are as follows:

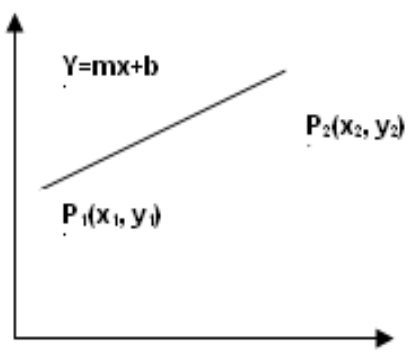
1. ***Analog Method:*** This method is also known as non retentive, memory-less or real time method. For conversion, this method uses a large number of delay cells. This method is mainly used for analog video.
2. ***Digital Method:*** This method is also known as retentive or buffer method. In this method, a picture is stored in a line or frame buffer with data rate speed (n1) and read with another data rate speed (n2). To improve the picture quality and to prevent the conversion artifacts, this method uses several picture processing techniques.

Example: In a 120/140Hz analog TV, there is a scan converter circuit which converts the vertical frequency (refresh rate) from standard 60/70Hz to 120/140Hz to achieve a low level of flicker which is important in large screen (high inch) TVs.

4.2 Line

In computer graphics, a line refers to a line segment, which is a part of a straight line that extends indefinitely in opposite directions. A line is defined by two endpoints and is represented by the line equation y=mx+b. In the equation y=mx+b, m is the slope and b the intercept of the line.

Figure 4.1: Defining a Line



In figure 4.1, the two endpoints are described by $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$. The line equation describes the coordinates of all the points that lie between the endpoints.

Caution Slope-intercept equation is not suitable for vertical lines.

Direct Use of the Line Equation

A line can be scan converted by following the steps given below:

1. Scan-convert the endpoints P1 and P2 to pixel coordinates (x1, y1) and (x2, y2) respectively
2. Set m = (y2-y1)/ (x2- x1) and b= y1-m x1
3. If $|m|\leq 1$, calculate the corresponding value of y using the equation and scan-convert (x, y) for every integer value of x between and excluding x1 and x2
4. If $|m|>1$, calculate corresponding value of x using the equation and scan-convert (x, y) for every integer value of y between and excluding y1 and y2

DDA Algorithm

The Digital Differential Analyzer (DDA) algorithm is one of the incremental scan-conversion methods. This approach is characterized by calculating at each step using the results obtained in the previous step.

Assume that in step 1, (x_i, y_i) has been calculated to be a point on line. As the next point, (x_{i+1}, y_{i+1}) must satisfy $\Delta y/\Delta x$ where $\Delta y=y_{i+1}-y_i$ and $\Delta x=x_{i+1}-x_i$. Hence we have:

$Y_i+1=y_i+m\Delta x$……...Eq (1)

$X_i+1=x_i+\Delta y/m$……...….Eq (2)

Equations (1) and (2) are used in DDA algorithms as follows:

1. When $|m|\leq 1$, assuming x1 < x2 start with x=x1 and y =y1 and set $\Delta x=1$. Calculate the y -coordinate of each point using yi+1= yi+m.
2. When $|m|\geq 1$, assuming y1<y2 start with x=x1 and y=y1 and set $\Delta y=1$. Calculate the x-coordinate of each point using xi+1=xi+1/m.

The above process is continued until x reaches x_2 or y reaches y_2 and all points are scan-converted to pixel coordinates.

> *Notes*
>
> The DDA algorithm is faster than the direct use of the line equation. This is because DDA algorithms calculate points on the line without any floating-point multiplication.

Bresenham's Line Algorithm

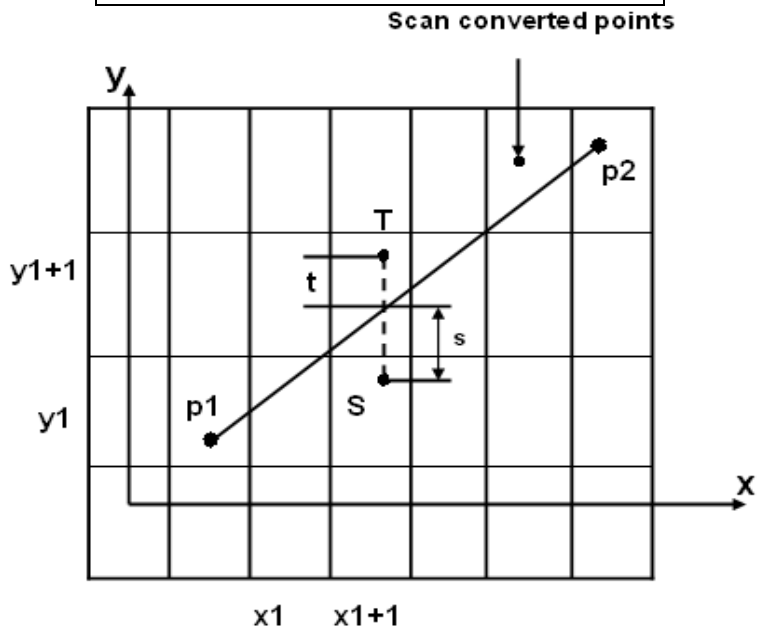
Bresenham's line algorithm is one of the highly efficient incremental methods for scan-converting lines. It is an algorithm that determines which points in an n-dimensional raster should be plotted in order to form a close approximation to a straight line between two given points. It is an accurate and efficient raster line-generating algorithm that uses only incremental integer calculations.

Bresenham's algorithm can be used to display lines, circles and other curves. The vertical axes show scan-line positions, and the horizontal axes identify pixel columns, that is, Bresenham's algorithm can be successfully used for drawing lines on a computer screen, subtracting, and bit shifting.

Bresenham's line algorithm works as follows. Assume that you want to scan-convert the line shown in figure 4.2, where 0<m<1. For this, start with pixel p_i (x_1, y_1), then select the subsequent pixels. After the pixels are chosen at any step, the next pixel is either the one to its right and down or one to its right and up due to the limit on m.

The following figure 4.2 depicts scan converting a line.

Figure 4.2: Scan Converting a Line



In figure 4.2, the coordinates of the last chosen pixel are (x_i, y_i). Choose the next pixel between the bottom pixel S and the top pixel T.

If S is chosen,

$x_{i+1}=x_i+1$

$y_{i+1}=y_i$

If T is chosen,

$x_{i+1}=x_i+1$

$y_{i+1}=y_i+1$

The actual y coordinate of the line at $x=x_{i+1}$ is:

$y=mx_{i+1}+b$

i.e., $y=m(x_i+1)+b$

The distance from S to the actual line in the y direction is:

$s=y-y_i$

The distance from T to the actual line in the y direction is:

$t=(y_i+1)-y$

Consider the difference between these two distance values: s-t

1. If s-t is less than zero, s<t and the closest pixel is S.
2. If s-t is greater than zero, s>t and the closest pixel is T.
3. If s-t is equal to zero then we have to choose T.

The difference may be represented in the following way:

$s-t= (y-y_i)-[(y_i+1)-y]$

$=2y-2y_i-1$

$=2m(x_i+1)+2b-2y_i-1$

If we substitute m by $\Delta y/\Delta x$ and introduce a decision variable $d_i=\Delta x(s-t)$, we get:

$d_i=2\Delta y*x_i-2\Delta x*y_i+C$ $(C=2\Delta y+\Delta x(2b-1))$

In the same way, we may write decision variable di+1 for the next step as follows:

$d_i+1=2\Delta y*x_i+1-2\Delta x*y_i+1+C$

$d_i+1-d_i=2\Delta y(x_{i+1}-x_i)-2\Delta x(y_{i+1}-y_i)$

As $x_{i+1}=x_i+1$ we get

$d_{i+1}=d_i+2\Delta y-2\Delta x(y_{i+1}-y_i)$

If the top pixel T is chosen then yi+1=yi+1

$d_{i+1}=di+2\Delta y-\Delta x$

If the bottom pixel S is chosen then yi+1=yi

di+1=di+2Δy

Hence we have

di+1={di+2(Δy-Δx) if di≥0

di+1={di+2Δy if di<0

d1 may be calculated from the original definition of the decision variable di:

d1=Δx [2m(x1+1) +2b-2y1-1]

=Δx [2(mx1+b-y1) +2m-1]

As mx1+b-y1=0 and m=Δy/Δx, we have

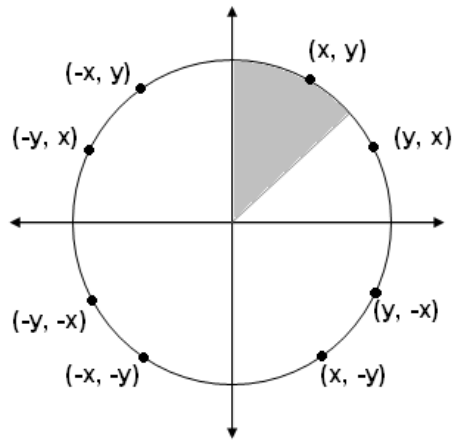
d1=2Δy-Δx

4.3 Circle

A circle is a symmetrical figure. Any circle-generating algorithms may be used to plot eight points for each value that the algorithm calculates. As shown in the figure 4.3, only one octant of the circle has to be generated. A successive reflection obtains the other parts. If the first octant (0 to 45^0) is generated the second octant can be obtained by reflection through the line y = x to yield the first quadrant. The results in the first quadrant are returned through line x = 0, to obtain those in the second quadrant.

The following figure 4.3 depicts eight-way symmetry of a circle.

Figure 4.3: Eight-way Symmetry of a Circle



Equation of a circle with (0, 0) as its centre is given by:

Fcircle (X, Y) = X2 + Y2 – R2

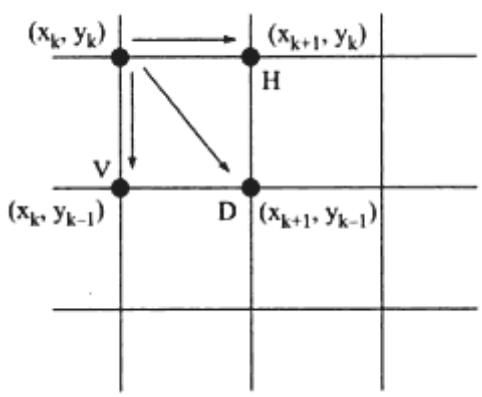
Checking the sign of the circle function determines the relative position of any point (X, Y), that is,

Fcircle (X, Y)< 0, if (X, Y) is inside the circle boundary

Fcircle (X, Y)= 0, if (X, Y) is on circle boundary

Fcircle (X, Y)> 0, if (X, Y) is outside the circle boundary

Figure 4.4: Equation of a Circle



In figure 4.4, the pixels are plotted at (X_k, Y_k), next we need to determine whether the pixel at position (X_{k+1}, Y_k) or one at position (X_{k+1}, Y_{k-+1}) or at position ((X_k, Y_{k-1}) is closer to the circle.

Firstly, let us check if the diagonal point is inside or outside the circle, i.e.

= $(X_{k+1})2 + (Y_{k-+1})2$ – R2 is < 0, = 0, or > 0

If $\Delta < 0$, then the diagonal point is inside the circle, i.e. select one of the two point (X_{k+1}, Y_k) or (X_{k+1}, Y_{k-1}), the one which is closer to the circle boundary. For this, the midpoint between these two points is checked i.e.

Pk = (Xk + 1) 2 + (Y k — 1/2) 2 - R2

If Pk > 0, this midpoint is outside the circle and the pixel on the scan line Yk-1 is closer to the circle boundary. If not the mid position is inside or on the circle boundary (Pk≤ 0) and we select the pixel on scan line Yk.

Similarly, If $\Delta > 0$, then the diagonal point is outside the circle, i.e. we have to select one of the two point (Xk+1, Y k-1) and (Xk, Y k-1), the one which is closer to the circle boundary. For this, once again we check for the midpoint between these two points, i.e.,

Pk = (Xk + 1/2) 2 + (Y k-1) 2 - R2

If Pk > 0, this midpoint is outside the circle and the pixel (Xk, Y k-1) is selected. Otherwise the mid-position is inside or on the circle boundary (Pk < 0) and we select the pixel (Xk, Y k-1).

If $\Delta = 0$

Pixel (Xk+1, Y k-1) is selected.

Bresenham's Midpoint Circle Algorithm for the First Quadrant is as follows:

Initialize the variables, Xk = 0 and Yk = R. Also, D = (Xk +1)2 + (Yk – 1)2 - R2

1. setpixel (Xk , Yk)

   If Yk ≤ 0 then step 7

   If D < 0 then step 2

   If D > 0 then step 3

   If D = 0 then step 5

2. Pk = (Xk +1)2 + (Yk – 1/2)2 - R2

   If Pk ≤ 0 then step 4

   If Pk > 0 then step 5

3. Pk = (Xk +1/2)2 + (Yk – 1)2 - R2

   If Pk ≤ 0 then step 5

   If Pk > 0 then step 6

4. Xk = Xk +1

   D = D + 2 Xk + 1

   goto step 1

5. Xk = Xk +1

   Yk = Yk +1

   D = D + 2 Xk - 2 Yk + 2

   goto step 1

6. Yk = Yk - 1

   D = D - 2 Yk + 1

   goto step 1

7. Finish

Midpoint Circle Drawing Algorithm

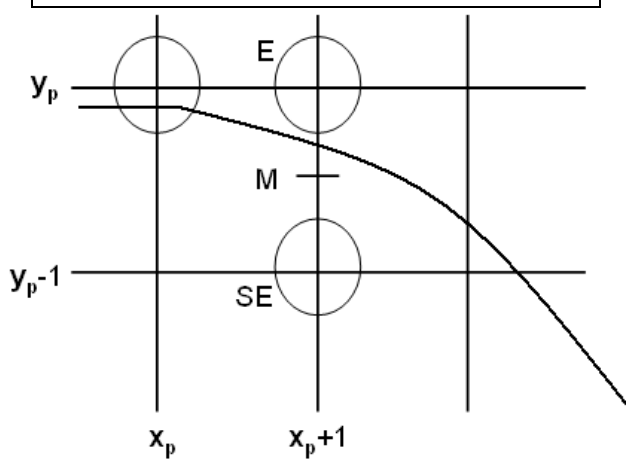
The midpoint circle drawing algorithm also uses the eight-way symmetry of the circle. An implicit representation of a function is used for the comparable elements of Bresenham's midpoint algorithm for circles. The function is as follows:

F(x, y) = x2 +y2 −R2 = 0

Note that the function is positive for the points outside the circle (or above the arc) and is negative for the points inside the circle (or under the arc).

Assume that we have drawn pixel ((xp, yp), and we must select the next pixel to draw (drawing clockwise around the boundary). Since the slope of the circular arc is between 0 and −1, our choice at each step is between the neighbor to the east E and the neighbor to the southeast SE. If the circle passes above the midpoint M, then either we go to E next or go to SE.

Figure 4.5: Midpoint Algorithm for Circle



Next, we need a decision variable. We take this to be the value of F (M), which is

D = F (M) = F (xp + 1, yp −1/2)

D = (xp+ 1)2 + (yp −1/2)2 −R2

If D < 0, then M is below the arc. Hence, the E pixel is closer to the line. On the other hand, if D _ 0 then M is above the arc, so the SE pixel is closer to the line.

Again, the new value of D will depend on our choice.

We go to E next: Then the next midpoint would have the coordinates (xp+2, yp (1=2)) and hence the new D value would be:

Dnew = F (xp + 2, yp −1/2)

Dnew = (xp+ 2) 2 + (yp −1/2)2 −R2

Dnew = (x2p+ 4xp + 4) + (yp −1/2)2 −R2

Dnew = (x2p+ 2xp + 1) + (2xp + 3) + (yp −1/2)2 −R2

Dnew = (xp+ 1) 2 + (2xp + 3) + (yp −1/2)22 −R2= D+ (2xp + 3):

Thus, the new value of D would be the current value + (2xp + 3).

We go to NE next: Then the next midpoint would have coordinates (xp + 2, Syp− 1 − (1=2)) and hence the new D value would be:

Dnew = F (xp + 2, yp −3/2)

Dnew = (xp+ 2)2 + (yp −3/2)2 −R2

Dnew = (x2p+ 4xp + 4) + (y2p −3yp +9/4) −R2

Dnew = (x2p+ 2xp + 1) + (2xp + 3) + (y2/p −yp +1/4) + (−2yp +8/4) −R2

Dnew = (xp+ 1) 2 + (yp −1/2)2 −R2 + (2xp + 3) + (−2yp + 2)

Dnew = D+ (2xp −2yp + 5)

Thus the new value of D would be the current value + (2(xp − yp) + 5).

The last issue is computing the initial value of D. Since we start at x = 0, y = R the first midpoint of interest is at x = 1, y = R−1=2, so the initial value of D is

Dinit = F (1 R−1/2)

= 1 + (R−1/2)2 −R2

= 1+R2−R+1/4−R2

=5/4−R:

You can make a very clever observation at this point is for testing if D is positive or negative. If we change the value of D, we do so by an integer increment. Thus, D is always of the form D0 + 1=4, where D0 is an integer. Such a quantity is positive if and only if D0 is positive. Hence you can ignore this extra 1=4 term, initialize Dinit = 1−R and the algorithm will work exactly in the same way that it should work.

4.4 Ellipse

An ellipse is a smooth curve which results about its horizontal and vertical axes. Like circles ellipse also show symmetry. Ellipse is a four-way symmetry. The equation of an ellipse is given by the following equation:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

Therefore,

F(x, y) = b 2x 2 + a2y 2 - a2b 2 = 0

The curvature of an ellipse is different at different points. Hence, we must divide the first quadrant into two unequal parts. Considering a tangent over this part of the ellipse, we see that it has a variable slope in the range [−∞, 0]. In the upper part, the curve is near the horizontal axis and in the lower part it is near the vertical axis. We can express the equation of the tangent to the ellipse in vector notation by computing the gradient of the ellipse, at any point (x, y). The x-component of the tangent vector is greater in the upper part, and the y-component is higher in the lower part. Thus, the point of division, corresponds to a point where both of these components are equal in magnitude.

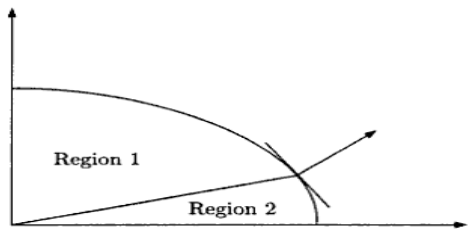
Now, the gradient of the elliptic curve at any point (x, y) is:

$$\nabla\{f(x,y)\} = \frac{\delta f}{\delta x} i + \frac{\delta f}{\delta y} j$$
$$= 2b^2 xi + 2a^2 yj$$

As mentioned earlier, at the point of division the x- and y- components are equal in magnitude, with the x- component greater in upper part and y-component in the lower part. Thus, if at any point on the ellipse, a2(Yp – ½) ≤ b2(Xp + 1), we switch from region 1 to region 2 as shown in Figure 4.6.

Figure 4.6 depicts two regions in the first quadrant of an eclipse.

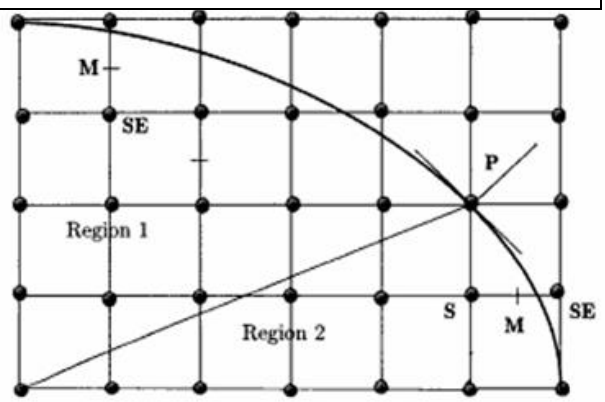
Figure 4.6: Two Regions in the First Quadrant of an Ellipse



Source: Computer Graphics, Second Edition, MALAY K, PAKHIRA , Chapter 2

Let us define decision variables **dold** for upper and **dnew** for the lower region respectively. For the upper region, f(X, Y) is evaluated at the midpoint (Xp +1, Yp – ½). The situation is described in figure 4.7.

Figure 4.7: Ellipse drawing using midpoint method



Source: Computer Graphics, Second Edition, MALAY K, PAKHIRA, Chapter 2

As in the case of a circle, to move to E we have:

dold = f (Xp +1, Yp – ½) = b2 (Xp +1)2 +a2 (Yp – ½)2 - a2b2

dnew = f (Xp +2, Yp – ½) = b2 (Xp +2)2 +a2 (Yp – ½)2 - a2b2

Therefore,

ΔE = dnew - dold = b2 (2Xp +3)

And to move to SE we have,

dnew = f (Xp +2, Yp – 3/2) = b2 (Xp +2)2 +a2 (Yp – 3/2)2 - a2b2

Therefore,

ΔSE = dnew - dold = b2 (2Xp +3) + a2 (-2Yp +2)

For the lower region, f (X, Y) is evaluated at midpoint (Xp + ½, Yp – 1).

To move to S,

dold = f (Xp +1/2, Yp – 1) = b2 (Xp +1/2)2 +a2 (Yp – 1)2 - a2b2

dnew = f (Xp +1/2, Yp – 2) = b2 (Xp +1/2)2 +a2 (Yp – 2)2 - a2b2

Therefore,

ΔS = dnew - dold = a2 (-2Yp +3)

To move SE,

dnew = f (Xp + 3/2, Yp – 2) = b2 (Xp +3/2)2 +a2 (Yp – 2)2 - a2b2

Therefore,

ΔSE = dnew - dold = b2 (2Xp + 2) + a2 (-2Yp + 3)

We must now compute the initial value of d, i.e. dstart. Let us start drawing at (0, b).

Then the first midpoint is (1, b - 1/2). Therefore,

f (1, b - 1/2) = b2 + a2 (b -1/2)2 +a2b2 = b2 + a2 (- b + 1/4)

During each of the iterations in the upper region, we must not only test the decision variable d and update the Δ functions, but also test switching regions by evaluating the gradient at the midpoint between E and SE. When we cross over to region 2 (lower region), we change our choice of two pixels to compare from E, SE to S, SE. At the same time, we change the value of our decision variable to d for region 2. If the last pixel chosen in region 1 is located at (Xp, Yp), then d is once again initialized at (Xp + ½, Yp - 1). We stop drawing when the Y-value becomes 0.

4.5 Arcs and Sectors

An arc is a part of the circumference of a circle as shown in figure 4.8 (a). Mathematically an arc can be generated using either the polynomial or trigonometric method. In the trigonometric method, the arc is generated using two angles. As shown in figure 4.8 (b), two angles θ1 and θ2 are used to generate the arc where, θ1 marks the beginning of the arc and θ2 marks the end of the arc. With the help of these two values an arc can be easily constructed.

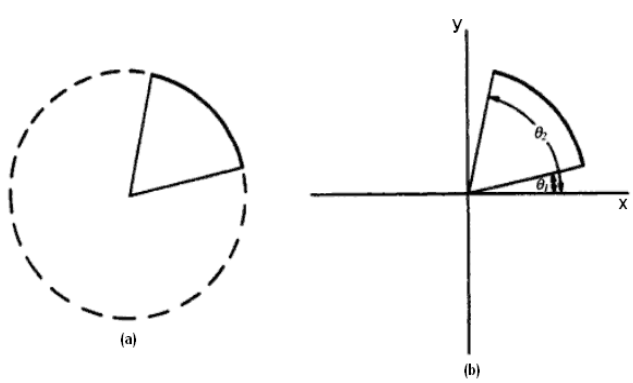
Notes

Since the arc is a part of the circumference of a circle. Hence, along with the two angles the important component that has to be taken into account is the radius of the circumference, i.e. distance between any point on the circumference and origin. Therefore, it is very important to consider all the three elements θ_1, θ_2, and radius while constructing the arc.

Both figures 4.8 (a) and (b) bring out the distinct relationship between the arc and circle. It also depicts how the angles θ1 and θ2 used to generate an arc. The rest of the steps that are followed to generate the arc are similar to those used for scan converting a circle, except that symmetry is not used. (Refer section 4.3 for scan converting a circle.)

Figure 4.8 illustrates an arc.

Figure 4.8: Illustration of an Arc

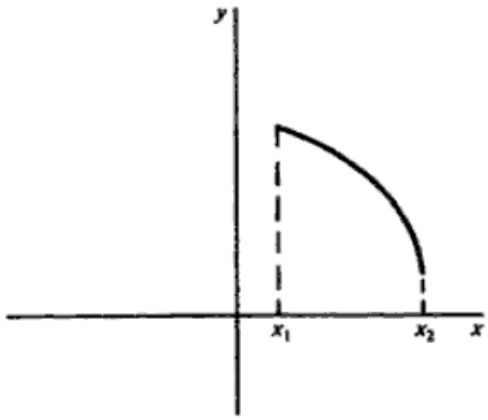


Source: Computer Graphics, II edition, ZHIGANG ZIANG, ROY PLASTOCK, SCHAUMS OUTLINES, Chapter 3

In polynomial method, the arc is generated using the point coordinate values (X, Y) of 2D space. In this method, the value of X is varied from X1, to X2 and the values of Y are found by evaluating the expression $\sqrt{r^2 - x^2}$ where, r is the radius of the arc and x is the point coordinate value in X-direction. Figure 4.9 depicts how the arc is generated using the value of x point coordinates x1 and x2, which are the start and end points of the arc respectively.

The following figure 4.9 depicts the arc generated using point coordinates.

Figure 4.9: Arc Generated Using Point Coordinates



Source: Computer Graphics, II edition, ZHIGANG ZIANG, ROY PLASTOCK, SCHAUMS OUTLINES, Chapter 3

Even though an arc appears to be part of a circle, it is not possible to draw the arc using Bresenham's circle drawing algorithm. According to Bresenham's algorithm, an arc's (x, y) coordinates should be the endpoints. But if the endpoints are required to be found, the common formulation becomes inefficient. This is as shown in figure 4.10 where the points are not positioned correctly to form an arc.

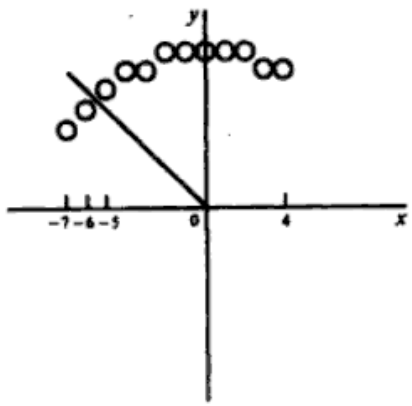
The endpoints for each 450 increment in the point coordinate values (i.e. rotate the point coordinate value by an angle 450) generate an arc. The endpoints of the arc must be tested to check whether the points form an arc, i.e., if the arc is created using ten points all the eight points which lie between the endpoints must be tested. Therefore, as per Bresenham's algorithm, the practice to draw an arc takes the time to calculate and test every point on the circle's parameter.

Caution

When using Bresenham's algorithm to generate an arc, there is always a danger of missing the end points of the arc. When the end points are missed, the software program that is written to generate the arc can be caught in an infinite loop.

The following figure 4.10 depicts arc generated using Bresenham's Algorithm.

Figure 4.10: Arc Generated Using Bresenham's Algorithm



Source: Computer Graphics, II edition, ZHIGANG ZIANG, ROY PLASTOCK, SCHAUMS OUTLINES, Chapter 3

Sectors

A sector is the portion of area that an arc creates in a circle. As shown in figure 4.8 (a), the triangular area of the circle created by the arc is called the sector of the circle. A sector is scan-converted by using any of the methods used for scan-converting an arc and then scan-convert the two lines forming the arc.

Example: Assume that a sector whose center is at point (h, k) is to be scan-converted. First, you need to scan-convert an arc from θ1 to θ2. Next, a line would be scan-converted from (h, k) to (r cos(θ1) + h, r sin(θ1) +k). A second line would be scan-converted from (h, k) or to (r cos(θ2) + h, r sin(θ2) +k).

4.6 Summary

- The process of representing continuous graphics objects as a collection of discrete pixels is known as scan conversion or rasterization.
- The two methods for changing a picture's data rate are analog method and digital method.
- Digital Differential Analyzer (DDA) algorithm is one of the incremental scan-conversion methods.

4.7 Keywords

2-D: Two-Dimensional

3-D: Three-Dimensional

Symmetrical: The two entities exhibiting equivalence among their constituents such as size, shape and position of parts.

Tangent: The line meeting a curve or surface at a point and has same direction as curve or surface at the point.

4.8 Self Assessment

1. State whether the following statements are true or false:
   (a) Various patterns are obtained on the computer screen by highlighting the required pixels.
   (b) In computer graphics, a line refers to a line segment, which is a part of a straight line that extends definitely in opposite directions.
   (c) A sector is the portion of area that an arc creates in a circle.
2. Fill in the blanks:
   (a) The instrument used for scan conversion is known as …….
   (b) The…… algorithm is one of the highly efficient incremental methods for scan-converting lines.
   (c) Ellipse is a …….symmetry.
3. Select the suitable choice for every question:
   (a) How many methods for changing a picture's data rate exist?
      (i) One (ii) Two (iii)Three (iv) Four
   (b) Bresenham's line algorithm is one of the highly efficient incremental methods for scan-converting which among the following?
      (i) Line (ii) Circle (iii)Ellipse (iv) Polygon

4.9 Review Questions

1. "Even though an arc appears to be part of a circle, it is not possible to draw the arc using Bresenham's circle drawing algorithm." Do you agree? Justify.
2. "The process of representing continuous graphics objects as a collection of discrete pixels is known as scan conversion or rasterization." Elaborate.
3. "The Digital Differential Analyzer (DDA) algorithm is one of the incremental scan-conversion methods characterized by calculating at each step using the results obtained in the previous step." Explain with the help of an example.
4. "Bresenham's algorithm can be successfully used for drawing lines on a computer screen, subtracting, and bit shifting." Discuss.
5. "It is very important to consider all the three elements θ1, θ2, and radius while constructing the arc." Do you agree? Justify.

Answers: Self Assessment

1. (a) True (b) True (c) True
2. (a) Scan Converter (b) Bresenham's line (c) four-way
3. (a) Two (b) Line

4.10 Further Readings

Books

K. Malay, Pakhira , Computer Graphics, 2nd Ed.

N.S.Amarendra, D.U.Arun, Computer Graphics, The McGraw-Hill Companies

Online link

http://whatis.techtarget.com/definition/0,,sid9_gci212200,00.html

http://www.tutornext.com/perimeter-polygons/909

Unit 5: Scan Conversion II

CONTENTS

Objectives

Introduction

5.1 Polygon

5.2 Region Filling

5.3 Glyph

5.4 Aliasing and Anti-aliasing

5.5 Summary

5.6 Keywords

5.7 Self Assessment

5.8 Review Questions

5.9 Further Readings

Objectives

After studying this unit, you will be able to:

- Explain scan conversion of polygon
- Discuss region filling
- Define glyphs
- Explain aliasing and anti-aliasing

Introduction

Scan conversion is the process of converting vertical or horizontal scan frequency of video signals for different purposes and applications. The scan converters are the devices that perform the scan conversion. In the previous unit, we have already discussed the two different methods of the scan conversion process. They are:

1. ***Analog Method***: This type of conversion is done using delay cells that are useful for analog video.
2. ***Digital Method***: In this method, the picture given in application program is digitized into a pixel values that are stored in the frame buffer.

We also discussed the method of scan converting circle, line, ellipse, and arcs and sectors. This unit discusses the method of scan converting the polygon. This unit also gives an overview on region filling, glyphs, aliasing and anti-aliasing.

5.1 Polygon

A polygon is a closed 2D object or figure that has three or more sides. Polygons are named based on the number of sides it has.

The table 5.1 shows a list of polygons and the number of sides corresponding to them.

Table 5.1: List of Polygons

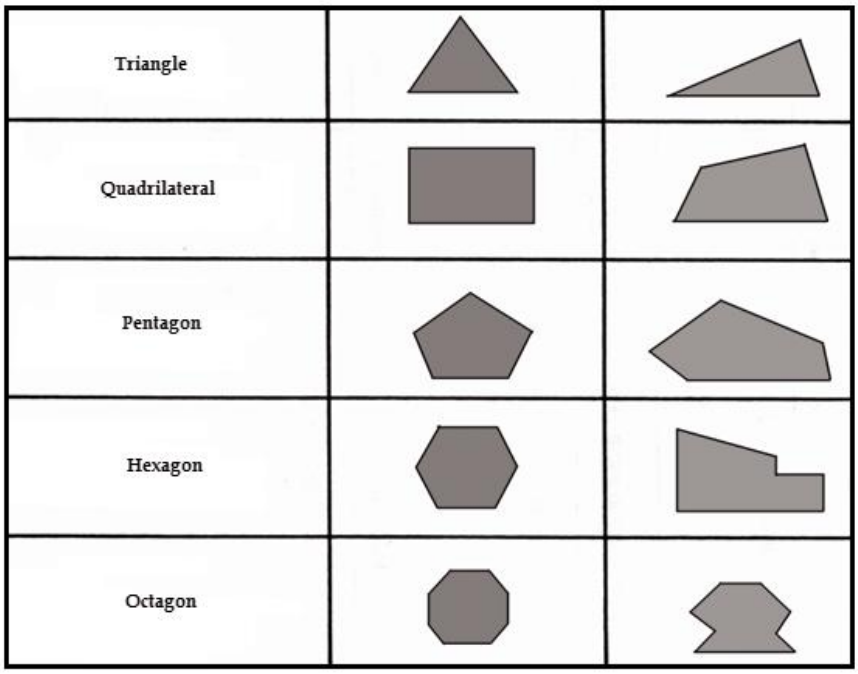
Number of Sides	Polygon name
3	Triangle
4	Quadrilateral
5	Pentagon
6	Hexagon
7	Heptagon
8	Octagon
9	Nonagon
10	Decagon
11	Undecagon
12	Dodecagon

Notes

Based on the sides and angles the polygons are classified as regular and irregular polygons.

Figure 5.1 shows the list of regular and irregular polygons.

Figure 5.1: List of Regular and Irregular Polygons



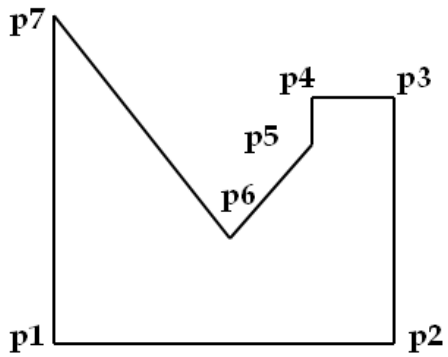
Cluster of polygons are used to represent closed contours.

Two important properties are required to scan convert polygons:

1. ***Spatial Coherence*:** This property states that the adjacent pixels of the polygons usually have the same characteristics.
2. ***Scan Line Coherence*:** This property states that scan line will have the same characteristics.

Many algorithms are used to scan convert a polygon. Parity scan conversion algorithm is one of the most popular algorithms. This algorithm is based on parity bit. Figure 5.2 (a) illustrates the parity scan conversion algorithm. Here, P1, P2, P3, P4, P5, P6, and P7 are the parity pairs.

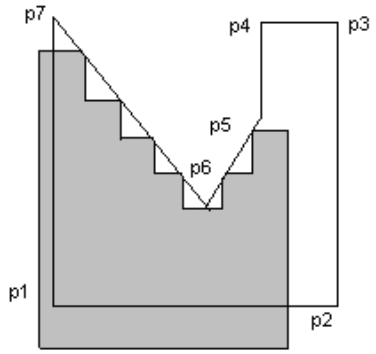
Figure 5.2 (a): Parity Scan Conversion Algorithm



When the scan lines intersect, polygon pairs set the parity flag to 1. The parity bit is set to check whether the pixel is within the polygon. If it is within the polygon the parity flag is set to 1 else to 0. Thus, if the parity flag is 1, then the scan line is inside the polygon and if the parity flag is zero then the scan line is outside the polygon.

When the parity flag is 1, the pixels inside the polygon gets the respective polygon color, if the parity flag is set to zero the polygon pixels takes the background color as shown in figure 5.2 (b).

Figure 5.2 (b): Parity Scan Conversion Algorithm

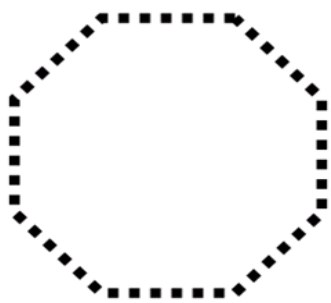


5.2 Region Filling

The process of coloring the object or image area or region is called region filling. The region of the image or the object is defined at the pixel level. At pixel level, the region is defined in terms of bounding pixels that outline the entire image or interior pixels that define the object on the screen. Based on this, two algorithms are defined for filling the object. They are:

1. ***Boundary-Defined Region*:** Figure 5.3 (a) shows the boundary defined region of a polygon. The algorithm used to define this is called the boundary-fill algorithm. This algorithm begins with a starting pixel or seed from which the filling starts in side the region. The algorithm continuously checks for the image's boundary. The boundary acts as the end of filling. All the pixels that lie within the boundary are filled. The algorithm works on an arbitrary region by checking and filling the non-boundary pixels. All the pixels that are not the part of the boundary are filled. The filling seed is considered as the reference point or start pixel, then the neighboring pixel is processed by the algorithm to check whether it is the boundary pixel or not. If the neighboring pixel is a non-boundary pixel it is filled and acts as the seed for next filling. If it is the boundary then filling stops for the current pixel. Thus, the algorithm processes and checks for non-boundary pixels and fills them.

Figure 5.3 (a): Boundary-Defined Region



(a)

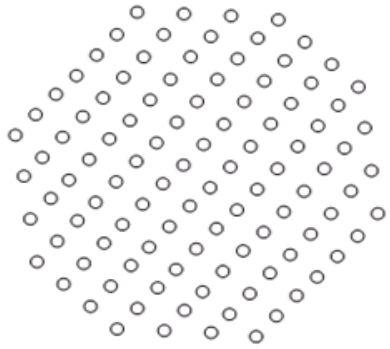
Notes

When the filling of the pixel takes place the pixel acquires a color that is defined by the algorithm.

2. ***Interior-defined Region*:** Figure 5.3(b) shows the interior-defined region of a polygon. The algorithm used to define this is called flood-fill algorithm. Even this algorithm begins with a seed inside the polygon. The pixel inside the polygon region has the region's original color i.e., the interior-deigned region of the polygon has a pre-defined color, when the algorithm finds the original color then the pixel is changed to a new color. This pixel acts as the seed for the next pixel and processes it. If the pixel does not has the region's color then that pixel is consider as the end of the polygon. The process continues till the colors of all the pixels are changed from the original to the new color. Thus, in a flood-fill algorithm the filling is done by chaining the original color to the new color.

The figure 5.3(b) depicts the interior-defined region of a polygon.

Figure 5.3 (b): Interior-Defined Region



(b)

5.3 Glyph

A glyph is nothing but a symbol or a graphical symbol that provides a form of a character. Here, character refers to any language's alphabet or number. A glyph can be a numeric or alphabetic font or some other symbol that pictures an encoded character.

Did you know? Glyph is pronounced as GLIHF and it originated from a Greek word meaning carving.

An ideal characterization of characters and glyphs and the relation between characters and glyphs can be understood with the help of table 5.2.

Table 5.2: Relation Between Characters and Glyphs

Characters	Glyphs
A character conveys distinctions in meaning or sounds.	A glyph conveys distinctions in form.
A character does not have any type of intrinsic appearance.	A glyph has no intrinsic meaning.
One or more characters may be depicted by one or more glyph representations	One or more glyphs may not be depicted by one or more character representations
In the Unicode standard, a character is stated to be an abstract entity.	In the Unicode standard, a glyph is not stated to be an abstract entity.

Notes

Two or more glyphs which have the same importance, whether used interchangeably or chosen depending on context, are called as allographs of each other.

Figure 5.4 illustrates the various glyphs representing the character '**a.**'

Figure 5.4: Various Glyphs Representing the Character 'a'

Let us now discuss the different types of glyphs. There are four major types of glyphs. They are:

1. ***One-Dimensional Glyphs*:** Different types of dataset of one-dimension can be easily mapped to any glyph of one attribute. There are many better ways to represent a one-dimensional data than glyphs. The reason is that the humans do not have any problems looking at one-dimension of scalar data as the substitution is in the speed of perception.
2. ***Two-Dimensional Glyphs*:** Two-dimensional glyphs lend itself to slightly more variation compared to one-dimensional glyphs. : Two-dimensional glyphs can:
   (a) Map both dimensions onto position in 2-D.
   (b) Map one-dimension to 1-D position and the other to glyph by changing attribute
   (c) Map both dimensions onto non-positional glyph attributes.
3. ***The Star Glyph*:** A star glyphs is a glyph with a certain number of points displayed within an imaginary circle.
4. ***Polygonal Glyphs*:** Polygonal glyphs can have different kind of geometries. In other words, the glyph itself is a polygon or a polygonal object in 2-D and 3-D.

Notes

Some characteristics of a glyph such as size, color, orientation, are always data driven.

Task

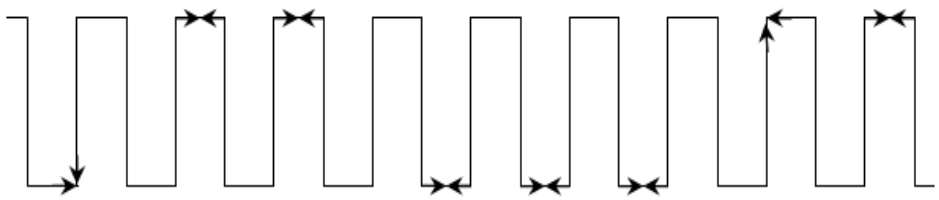
Research and find out information on Mayan glyphs translations.

5.4 Aliasing and Anti-aliasing

Aliasing refers to an effect that causes different signals to become indistinguishable at the time of sampling signals. In other words, when a quickly changeable signal is sampled infrequently then the signal represents the lower frequency. The lower frequency then replaces the frequency of the original signal. This is called as aliasing.

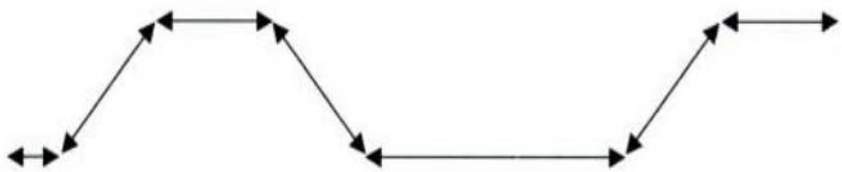
Figure 5.5 (a) shows a rapidly varying square wave is sampled uniformly at the instants shown by dots. Based on these samples, a lower frequency signal appears to be a square wave as shown as Figure 5.5 (b).

Figure 5.5 (a): Aliasing Signals in Square Wave



Source: Computer Graphics, The McGraw-Hill Companies, Amarendra N Sinha, Arun D Udai, Chapter 3

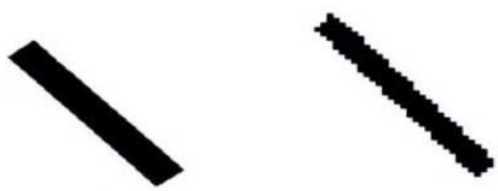
Figure 5.5 (b) : Aliasing in Sampling of Signals



Source: Computer Graphics, The McGraw-Hill Companies, Amarendra N Sinha, Arun D Udai, Chapter 3

The process of separation of the pixels in raster display also results in aliasing. The jagged appearance of a black rectangle is caused by the pixel display in a fixed rectangle as shown in Figure 5.6.

Figure 5.6 : Aliasing of a Rectangle



Source: Computer Graphics, The McGraw-Hill Companies, Amarendra N Sinha, Arun D Udai, Chapter 3

Aliasing occurs for many objects where its attributes are calculated only if the size of a pixel is bigger than the object. However, it will not affect the picture. Alternatively, if the small object covers the point at which the pixel attributes are calculated, then it may influence those attributes. Figures 5.7 (a) and (b) shows the effect.

If the centre of the pixel is used to determine the attributes then the entire pixel would exhibit those attributes of the small objects as shown in Figure 5.7 (a). However, Figure 5.7 (b) illustrates pixels object that would be ignored or lost (long or thin objects ignored), particularly in an animation sequence.

Figure 5.7 (a): Aliasing Effect on a Small Object

Source: Computer Graphics, Second Edition, MALAY K, PAKHIRA , Chapter 9

Figure 5.7 (b): Aliasing Effect in Animation

Source: Computer Graphics, Second Edition, MALAY K, PAKHIRA , Chapter 9

Anti-aliasing

Anti-aliasing is a method of deceiving the eye where a jagged edge is shown as a smooth end. Anti-aliasing is often used in games and on graphics cards. Especially, in games the ability to smoothen the edges of images goes a long way in creating a realistic 3D image on the screen. However, it is important to know that anti-aliasing actually does not smooth out any edges of the images as it merely deceives the eye. Let us refer to figure 5.8 (a) as an example to demonstrate the effects of anti-aliasing.

Figure 5.8 (a): Effect of Anti-aliasing

In figure 5.8 (a), the letter on the left is blown up and it is without any anti-aliasing. The letter on the right has anti-aliasing applied to it. In the blown up form it looks like it is simply blurred but if the size is reduced you may see the difference as shown in figure 5.8 (b).

The figure 5.8(b) depicts the effect of anti-aliasing.

Figure 5.8 (b): Effect of Anti-aliasing

Now look closely at the two letters shown in figure 5.8 (b). You can still tell that the letter on the left is jagged but the letter on the right looks a lot smoother. It is very important to know that the image has been shrunk down back to normal size and it is not been altered in any other way. Therefore, it is understood that anti-aliasing brings a much more pleasing image to the eye.

5.5 Summary

- A polygon is a closed 2D object or figure that has three or more sides.
- The process of coloring the object or image area or region is called region filling.
- A glyph can be a numeric, alphabetic font or some other symbol that pictures an encoded character.
- The characteristics of the glyph are always data driven such as size, color, orientation.
- Aliasing refers to an effect that causes different signals to become indistinguishable at the time of sampling signals.
- Anti-aliasing is a method of deceiving the eye where a jagged edge is shown as a smooth end.

5.6 Keywords

***Coherence*:** This is the quality when all the parts are clearly connected.

***Contour*:** This is the outline of a figure, body or mass.

***Intrinsic*:** This is an inherent property of a system or a material itself.

***Jagged*:** This is the state of an object marked by irregular projections on the edge or surface.

5.7 Self Assessment

1. State whether the following statements are true or false:
   (a) The process of aliasing involves altering the picture's data rate information and then inserting the new picture within appropriate synchronization signals.
   (b) A polygon is a closed 3D object or figure that has three or more sides.
   (c) Cluster of polygons is used to represent closed contours.
2. Fill in the blanks:
   (a) …… refers to an effect that causes different signals to become indistinguishable at the time of sampling signals.
   (b) The algorithm used to define the interior-defined region of a polygon is called ……
   (c) A …… is nothing but a symbol or a graphical symbol that provides a form of a character.
   (d) …….. is often used in games and on graphics cards.

3. Select the suitable choice for every question:

   (a) A 9 sided polygon is known as:

   (i) Dodecagon (ii) Nonagon

   (iii) Undecagon (iv) Decagon

   (b) Parity scan conversion algorithm is used to scan convert which among the following?

   (i) Arc (ii)Ellipse

   (iii) Polygon (iv) Line

   (c) A glyph with a certain number of points displayed within an imaginary circle is known as:

   (i) One-Dimensional Glyph (ii) Two-Dimensional Glyph

   (iii) Star Glyph (iv) Polygonal Glyph

5.8 Review Questions

1. "Two important properties are required to scan convert polygons." List and briefly explain the properties.
2. "A glyph can be a numeric, alphabetic font or some other symbol that pictures an encoded character." Briefly discuss the difference between a glyph and a character.
3. "Anti-aliasing is a method of deceiving the eye where a jagged edge is shown as a smooth end." Explain.
4. "Two algorithms are defined for filling an object." List and discuss the two algorithms.
5. Briefly discuss the four major types of glyphs.
6. "Polygons are named based on the number of sides it has." Discuss with examples.
7. Explain the process of aliasing.

Answers: Self Assessment

1. (a) False (b) False (c) True
2. (a) Aliasing (b) Flood-Fill Algorithm (c) Glyph (d) Anti-Aliasing
3. (a) Nonagon (b) Polygon (c) Star Glyph

5.9 Further Readings

Books

K. Malay, Pakhira , Computer Graphics, 2nd Ed.

N.S.Amarendra, D.U.Arun, Computer Graphics, The McGraw-Hill Companies

Online link

http://whatis.techtarget.com/definition/0,,sid9_gci212200,00.html

http://www.tutornext.com/perimeter-polygons/909

Unit 6: 2-D Transformation

CONTENTS

Objectives

Introduction

6.1 Transformation Matrix

6.1.1 Translation Transformation

6.1.2 Scaling Transformation

6.1.3 Reflection Transformation

6.1.4 Rotation Transformation

6.1.5 Shear Transformation

6.2 Homogeneous Coordinate System

6.3 Composite and Inverse Transformations

6.4 Affine Transformation

6.5 Summary

6.6 Keywords

6.7 Self Assessment

6.8 Review Questions

6.9 Further Readings

Objectives

After studying this unit, you will be able to:

- Define transformation matrix
- Explain homogeneous co-ordinate system
- Discuss composing and inverting of 2-D transformation
- Explain affine transformation

Introduction

Any object or image represented in a graph with only X and Y axis is called two-dimension or 2-D object or image. The graph with X and Y axis is called 2-D space. Any object represented in this space can be mapped or transformed. Mapping or transformation means changing or modifying the various parameters of an object used to define the object such as length, angle, area, position, and so on. Transformations that cause such modifications are called linear transformations.

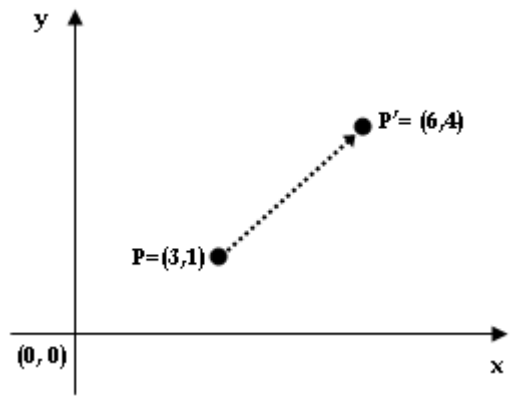
Matrix is the mathematical tool used to carry out linear transformation. This provides easy and accurate transformation values of the object before and after transformation. In computer graphics, a graphic designer can write programs to represent an object in the matrix form and apply transformations on the object. Thus, the use of matrix to transform a 2-D object not only helps the programmer to define the object but also to have a good control over the object that is being transformed.

6.1 Transformation Matrix

2-D graphic objects can be represented using a set of points on a simple graph of X and Y axis. The coordinate values of the points that make the object are used to determine the position and size of the object. A 2-D transformation is nothing but mapping or transforming a 2-D point. When 2-D transformation is applied to a 2-D point, it is transformed to another 2-D point in the 2-D space. This means that a 2-D point P=(x, y) transforms to P'=(x', y') after transformation.

Example: As shown in figure 6.1, P is a 2-D point, which is defined by P= (3, 1). After transformation the point is now at P'= (6, 4).

Figure 6.1: Simple Example of 2-D Transformation of a 2-D Point



Notes

In 2-D transformation, vectors and matrices play an important role in defining the 2-D graphic object.

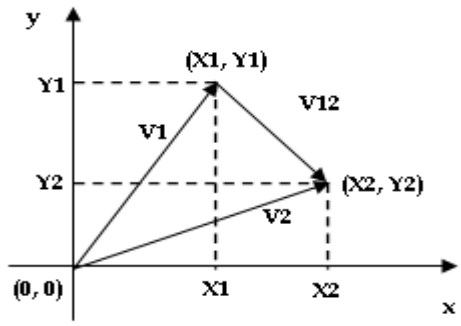
1. In vector form, the coordinate values are represented as points that define the 2-D object. By changing the coordinate values, the object can be moved from one region to another region of the screen or 2-D space.

2. In matrix form, the vector coordinates of the object are represented as matrix and transformation is applied to the object.

A vector is a row array that represents the displacement of a point in a 2-D space. The point defines the X and Y coordinates at a fixed position. As shown in figure 6.2, vector V1 is the displacement of the point (X1, Y1) from the origin (0, 0), similarly V2 is the displacement of the point (X2, Y2) from the origin (0, 0). V12 is the displacement between the points (X1, Y1) and (X2, Y2).

The figure 6.2 depicts a vector and point representation in 2-D space.

Figure 6.2: A Vector and Point Representation in 2-D Space



Notes

A vector (i.e. V1) represents the direction and magnitude of the line and a point (i.e. X1, Y1) represents a fixed position in 2-D space.

The points (X1, Y1) and (X2, Y2) are called the point coordinates and help to determine the position and shape of an object in 2-D space.

Example: Let P= (X, Y) be a point coordinate in a 2-D space. This point can be represented in matrix form as,

$$P = \begin{pmatrix} X \\ Y \end{pmatrix}$$

A 2-D object can undergo the following transformation in a 2-D space,

1. Translation
2. Scaling
3. Reflection
4. Rotation
5. Shearing

These transformations help the graphic designer to modify the 2-D objects. The technique that the designer has to adopt is explained in the following sub-sections.

6.1.1 Translation Transformation

Translation transformation is used to move the object within the X and Y directions of the 2-D space. The translation matrix for 2-D transformation is given by,

$$T=\begin{pmatrix} d_x \\ d_y \end{pmatrix}$$

Where, d_x and d_y are translation values in X and Y direction.

The translation matrix is added to the point coordinate matrix of the object individually to perform translation transformation on the object.

Let P be the point coordinate matrix and P' be the point coordinate matrix after translation. The translation equation is given as,

P'=T+P

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} d_x \\ d_y \end{pmatrix} + \begin{pmatrix} x \\ y \end{pmatrix}$$

$$P' = \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x + d_x \\ y + d_y \end{pmatrix}$$

Example: Consider a rectangular object whose point coordinates are (0, 0), (2, 0), (2, 2) and (0, 2). The object has to be moved 2 points in X direction and 3 points in Y direction.

The translation transformation matrix is written as,

$$T = \begin{pmatrix} d_x \\ d_y \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$$

The point coordinates of the rectangular object is added to the transformation matrix individually.

i.e. point coordinate (2, 0) is represented in matrix form as,

$$P = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$$

P'=T+P

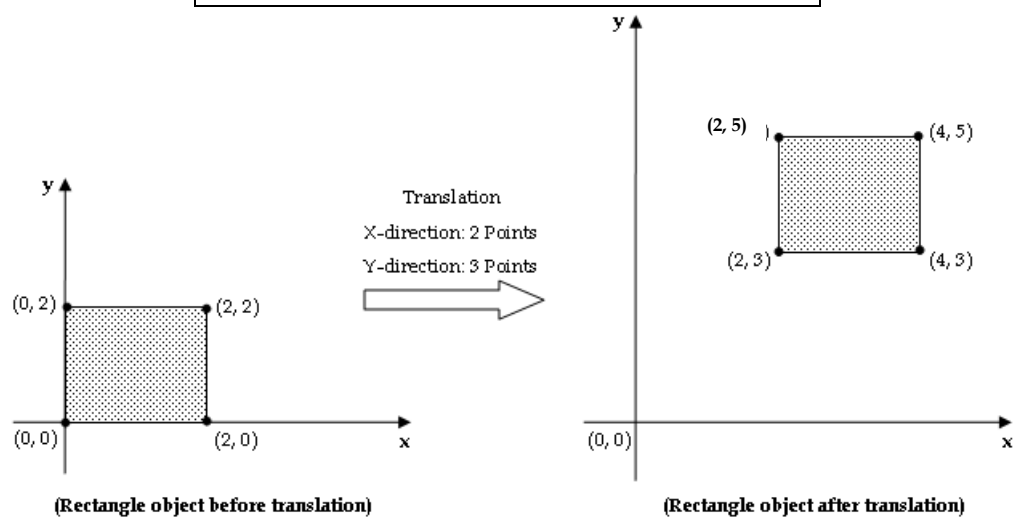
$$P' = \begin{pmatrix} 2 \\ 3 \end{pmatrix} + \begin{pmatrix} 2 \\ 0 \end{pmatrix}$$

$$P' = \begin{pmatrix} 4 \\ 3 \end{pmatrix}$$

Similarly, all the respective translation transformation object points are calculated. The point coordinates of the object after translation are (2, 3), (4, 3), (4, 5) and (2, 5). The figure 6.3 shows the rectangular object before and after translation by 2 points in X direction and 3 points in Y direction.

The figure 6.3 depicts the translation transformation in a 2-D space.

Figure 6.3: Translation Transformation in a 2-D Space



6.1.2 Scaling Transformation

Scaling transformation is done to resize the object, i.e. the dimension of the object is changed. The object is scaled in the X direction of the 2-D graph, by multiplying all the X coordinate points of the object by a scaling factor S*x*. Similarly, the Y coordinates of all the points of the object are multiplied by the scaling factor S*y*. These scaling factor values define the amount by which the object has to be scaled in X and Y direction.

Notes

You can use same or different scaling factor values for X and Y coordinates, while scaling the object.

1. If you use same scaling factor value for X and Y coordinates, uniform scaling of the object takes place in X and Y directions.

2. If you use different scaling factor values, then uneven scaling takes place in X and Y directions.

The following is a simple matrix multiplication carried out to find the scaled coordinate (x′, y′). Equation 6.1 is the product of scaling matrix $\begin{pmatrix} S_x & 0 \\ 0 & S_y \end{pmatrix}$ and point coordinate matrix $\begin{pmatrix} x \\ y \end{pmatrix}$

$$(x', y') = \begin{pmatrix} S_x & 0 \\ 0 & S_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

$$= \begin{pmatrix} S_x x + 0 \\ 0 + S_y y \end{pmatrix}$$

$$(x', y') = \begin{pmatrix} S_x x \\ S_y y \end{pmatrix} \text{ ..... (Eq.6.1)}$$

Here,

The scaling matrix $\begin{pmatrix} S_x & 0 \\ 0 & S_y \end{pmatrix}$ is used to perform the scaling transformation on the object. S*x* and S*y* are the scaling factors. All the point coordinates of the object have to be multiplied with the scaling matrix to scale the object.

Notes

When performing matrix multiplication we need to make sure that the number of columns in the first matrix is equal to number of rows in the second matrix.

Let A and B are two 2X2 matrices, i.e. two rows and two columns,

$$A=\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

$$B=\begin{pmatrix} p & q \\ m & n \end{pmatrix}$$

As you can see the number of columns in matrix A is equal to number of rows in matrix B. Therefore, matrix multiplication can be carried out.

$$A*B=\begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} p & q \\ m & n \end{pmatrix}$$

$$A*B=\begin{pmatrix} a.p+b.m & a.q+b.n \\ c.p+d.m & c.q+d.n \end{pmatrix}$$

The resultant matrix is also a 2X2 matrix.

If matrix A is a 2X2 matrix and B is a 2X1 matrix the matrix multiplication is carried out as shown,

$$A=\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

$$B=\begin{pmatrix} p \\ q \end{pmatrix}$$

As you can see the number of columns in matrix A is equal to number of rows in matrix B. Therefore, matrix multiplication can be carried out.

$$A*B=\begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} p \\ q \end{pmatrix}$$

$$A*B=\begin{pmatrix} a.p+b.q \\ c.p+d.q \end{pmatrix}$$

The resultant matrix after multiplication is a 2X1 matrix.

Example:

Consider a scaling transformation that shrinks the object in X and Y directions uniformly by a factor 2.

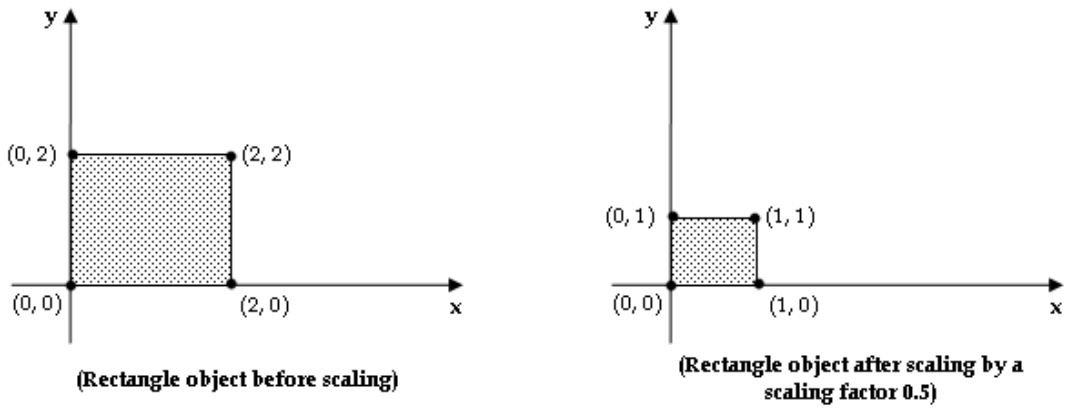
This means that the scaling factors Sx and Sy are 0.5. Therefore the scaling matrix is written as,

$$\begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix}$$

The figure 6.3 shows a rectangle object that is scaled by a factor 0.5 in X and Y directions of the 2-D space. The four points of the rectangle object (0, 0), (2, 0), (0, 2) and (2, 2) are multiplied individually with the scaling matrix to scale the rectangle. The point coordinate values of the rectangle after multiplying with the scaling matrix are (0, 0), (1, 0), (0, 1) and (1, 1). As you can see the rectangle after scaling is uniformly scaled on all sides as shown in figure 6.4.

The figure 6.4 depicts the uniform scaling of a rectangular object.

Figure 6.4: Uniform Scaling of a Rectangle Object

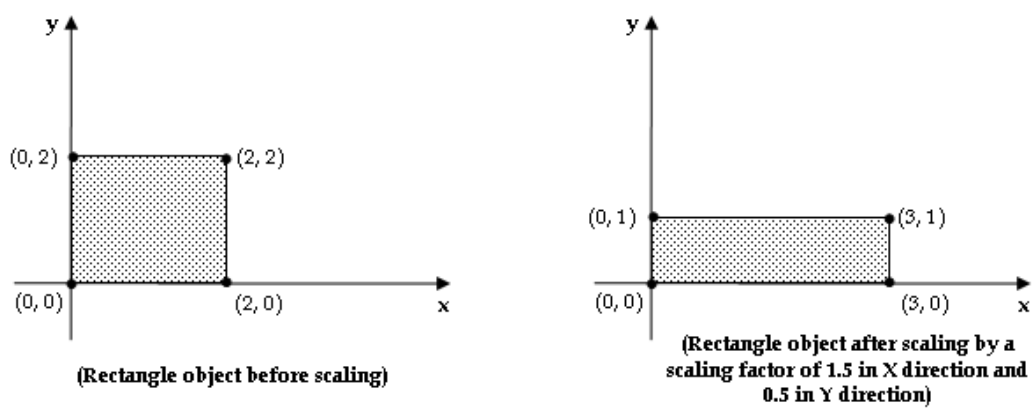


Example: Consider an example where the object is scaled unevenly, i.e., the object is scaled by different scaling factors in X and Y directions. Let the scaling factor be in the X direction, $S_x = 1.5$ and Y direction, $S_y = 0.5$. The scaling matrix can be represented in the matrix form as,

$$\begin{pmatrix} 1.5 & 0 \\ 0 & 0.5 \end{pmatrix}$$

The figure 6.4 shows a rectangle object that is scaled by a factor 1.5 in the X direction and 0.5 in the Y direction of the 2-D space. The four points of the rectangle object (0, 0), (2, 0), (0, 2) and (2, 2) are multiplied individually with the scaling matrix to scale the rectangle. The point coordinate values of the rectangle after multiplying with the scaling matrix are (0, 0), (3, 0), (0, 1) and (3, 1) as shown in figure 6.5.

Figure 6.5: Uneven Scaling of a Rectangle Object



> **Task**
>
> A rectangle object with coordinate values (2, 0) (5, 0), (5, 1) and (2, 1). Calculate the values of the rectangle after it is scaled 0.5 points in X-direction and 1.5 in Y-direction. Plot the rectangle before and after applying the scaling in 2-D space or graph.

6.1.3 Reflection Transformation

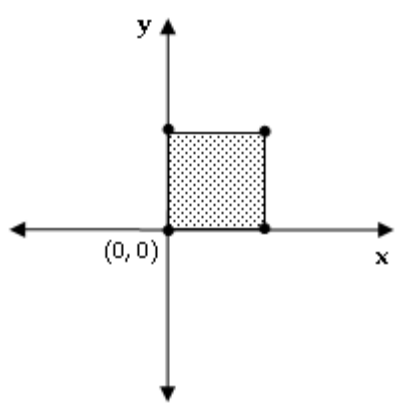
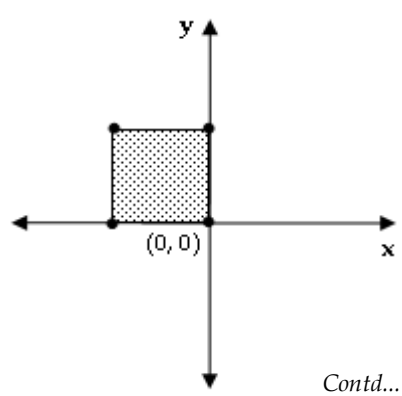
Reflection transformation of a 2-D object creates a mirror image of the object either across the X or Y axis of a 2-D space. The reflection transformation of an object is similar to scaling the object with a negative scaling factor. The reflection transformation matrix to create:

1. Reflection about the Y axis: $\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$
2. Reflection about the X axis: $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
3. Reflection about the origin: $\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$
4. Reflection within the same quadrant: $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

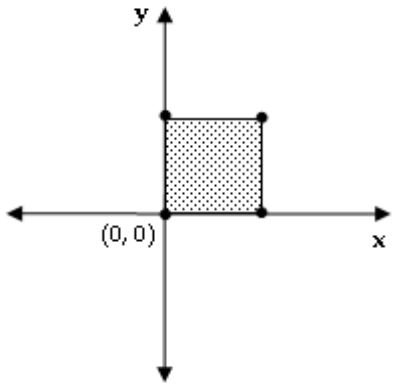
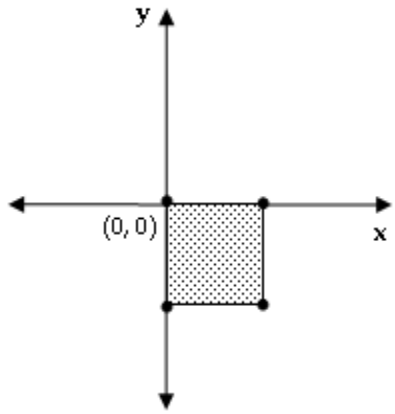
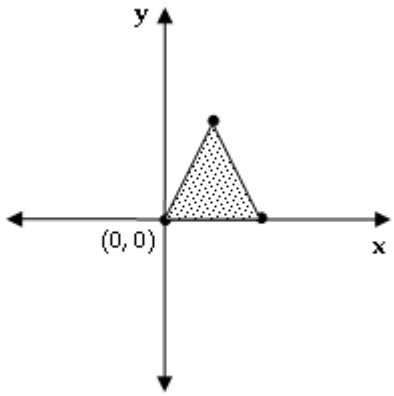
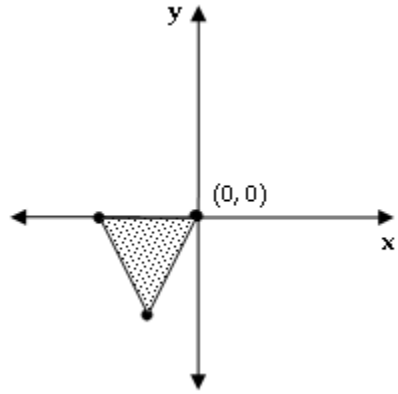
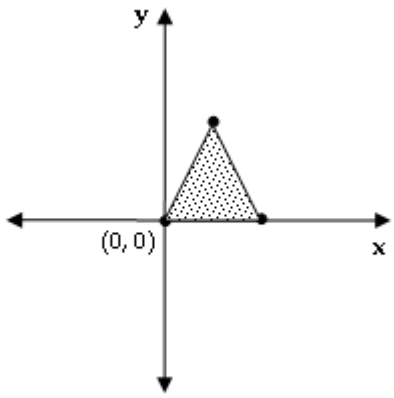
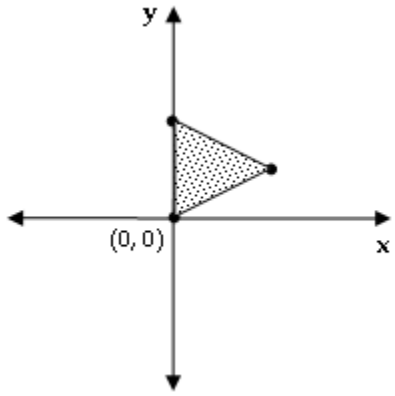
> **Notes**
>
> The scaling factor while performing reflection transformation is 1, but the magnitude, i.e., -1 and +1 changes with respect to the type of reflection either about X axis or Y axis.

When the reflection transformation matrix is multiplied with the point coordinates of a 2-D object the respective reflections are obtained, which are as shown in table 6.1.

Table 6.1: Types of Reflections

Reflection	Original Object	Reflected Object
About the Y axis		

Contd...

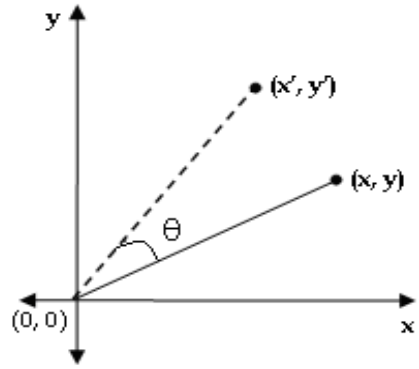
About the X axis		
About the origin		
Within the same quadrant		

6.1.4 Rotation Transformation

The rotation transformation rotates the object by an angle θ. As shown in figure 6.6, the point coordinate P = (x, y) is rotated by an angle θ. The rotation takes place with respect to the origin and in counterclockwise direction. After rotation the point coordinate P is transformed to a new position P' = (x', y').

The figure 6.6 depicts the rotation of a point coordinate.

Figure 6.6: Rotation of a Point Coordinate



The rotation matrix to rotate the object by an angle θ counterclockwise is given by:

$$R(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

Multiply all the point coordinates (point coordinate matrices) of the object with the rotation matrix to rotate the object by an angle θ. The following is the matrix multiplication of rotation matrix R (θ) and point coordinate matrix (P) to obtain the rotated point coordinate matrix (P′).

$$P' = R(\theta) \cdot P$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \cdot \cos\theta - y \cdot \sin\theta \\ x \cdot \sin\theta + y \cdot \cos\theta \end{pmatrix} \quad \text{…….. (Eq.6.2)}$$

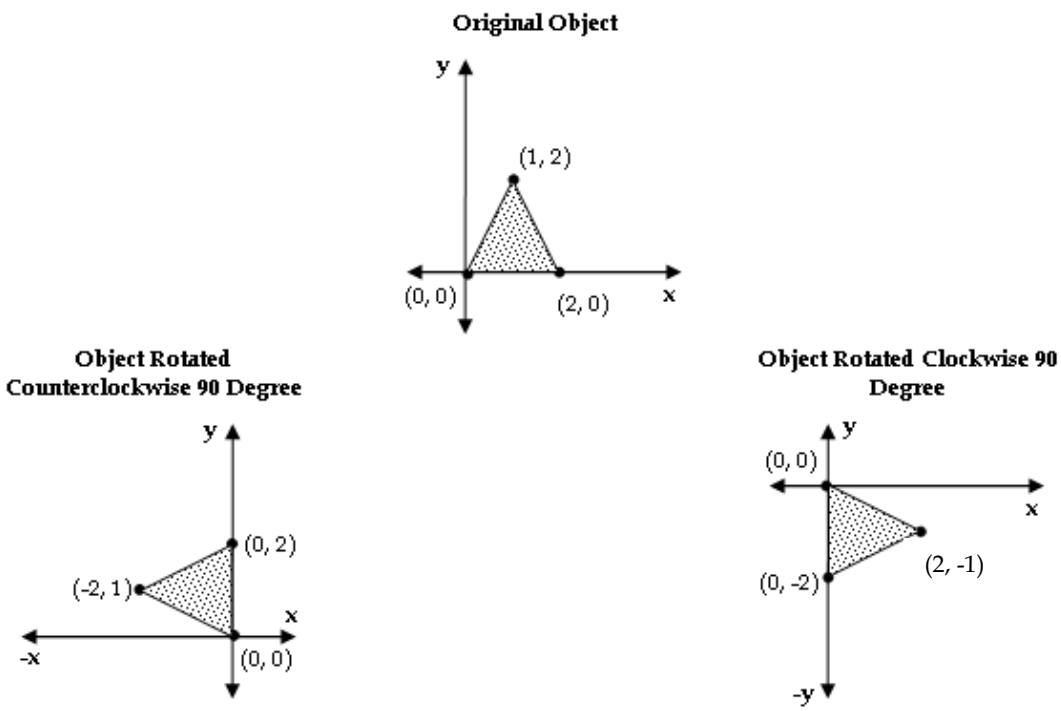
The equation 6.2 is used to calculate the coordinates of the object that is to be rotated counterclockwise by an angle θ. All the coordinates of the object have to be multiplied with the rotation matrix and the rotated coordinates of the original object is calculated. Similarly, we can rotate the object clockwise. The equation 6.3 is used to rotate the object by an angle θ clockwise.

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \cdot \cos\theta + y \cdot \sin\theta \\ -x \cdot \sin\theta + y \cdot \cos\theta \end{pmatrix} \quad \text{…….... (Eq.6.3)}$$

Example: A triangle object is to be rotated counterclockwise by an angle 90 degree, with reference to the origin of the 2-D space. The triangle object has three point coordinates (0, 0), (2, 0), and (1, 2). Using the equation 6.2, the point coordinate values of the triangle are calculated. The point coordinate values of the triangle after rotating it by an angle 90 degree are (0, 0), (0, 2), and (-2, 1). The object is rotated clockwise by using the equation 6.3. The coordinates of the object rotated clockwise are (0, 0), (0, -2), and (-1, 2).

Figure 6.7 is a graphical representation of the triangle object before and after rotation.

Figure 6.7: Triangle Rotated by an Angle 90 Degree



6.1.5 Shear Transformation

The shear transformation makes an object slant either in X or Y direction and is called as X-shear and Y-shear respectively. In X-shear, the y-coordinate values of the object remain constant, but the x-coordinate values are changed. This makes the vertical lines (lines that are parallel to Y-axis or lines that are perpendicular to X-axis) to tilt left or right. Similarly, in Y-shear, the x-coordinate values of the object remain constant, but the y-coordinate values are changed. This makes the horizontal lines (lines that are parallel to X-axis or lines that are perpendicular to Y-axis) to transform up or down.

The X-shear transformation matrix is given by,

Sh_x=**Error! Bookmark not defined.** $\begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix}$, where, 'a' is the X-shear factor.

The Y-shear transformation matrix is given by,

Sh_y=**Error! Bookmark not defined.** $\begin{pmatrix} 1 & 0 \\ b & 1 \end{pmatrix}$, where, 'b' is the Y-shear factor.

Notes

The object transforms **right** during X shear and transforms **up** during Y shear, when the value of '**a**' and '**b**' are **positive**. If the values are **negative**, the object tilts **left** during X-shear and transforms **down** during Y-shear.

Consider a point coordinate (x, y), represented in the matrix form as,

$$P=\begin{pmatrix} x \\ y \end{pmatrix}$$

Let, P' be the coordinate after shear transformation,

P'=Sh.P

Here, Sh can be either X or Y-shear.

If it is X-shear then,

$$P'=\begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix}$$

$$P'=\begin{pmatrix} x+ay \\ y \end{pmatrix}$$

If it is Y-shear then

$$P'=\begin{pmatrix} 1 & 0 \\ b & 1 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix}$$

$$P'=\begin{pmatrix} x \\ bx+y \end{pmatrix}$$

Example: Consider a rectangle object with point coordinates (0, 0), (2, 0), (2, 2) and (0, 2). The rectangle undergoes X-shear with the shear factor value 1.

The shear transform matrix for X-shear is given as,

$$Sh_x=\begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix}=\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

Here, the value of the shear factor, a=1.

After multiplying the point coordinates of the rectangle with the X-shear transform matrix individually we obtain the following coordinate points that define the rectangle after X-shear transform are (0, 0), (2, 0), (4, 2) and (2, 2).

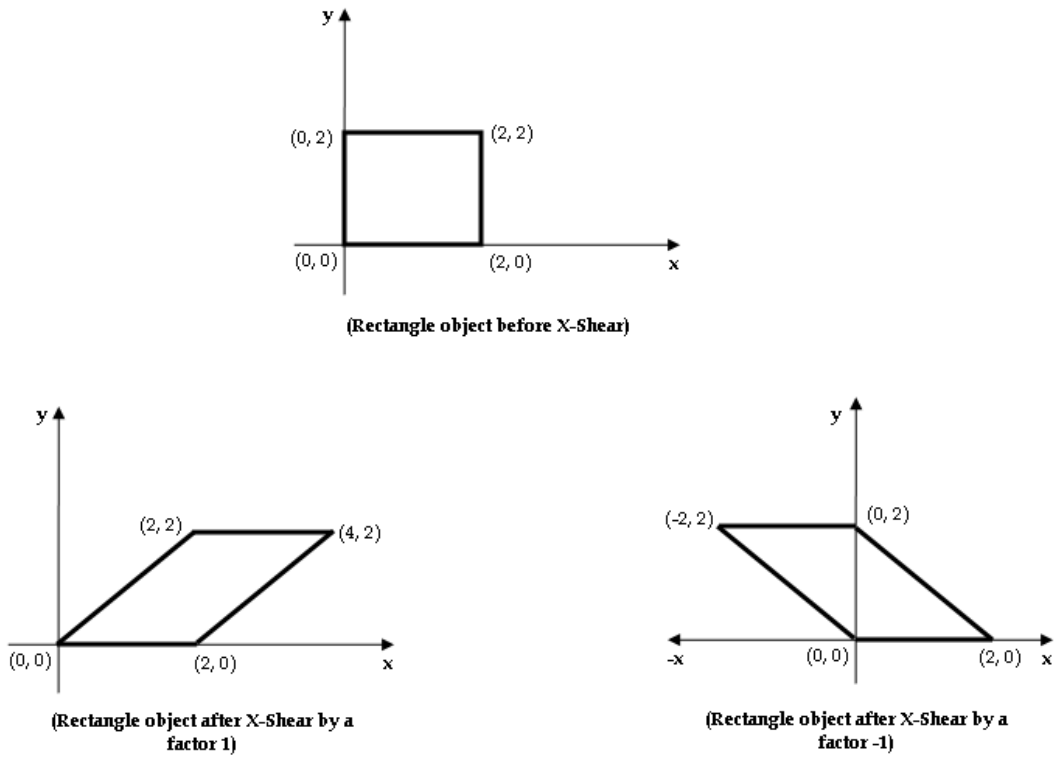
If the value of a=-1, then the X-shear transform matrix is given as,

$$Sh_x=\begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix}$$

Then the value of the rectangle after X-shear transform are (0, 0), (2, 0), (0, 2) and (-2, 2).

The figure 6.8 shows the original rectangle and X-shear transformed rectangle for shear factors 1 and -1.

Figure 6.8: X-Shear Transformation



Task

Plot a pentagon on the 2-D space and apply shear transformation in X and Y directions. Let the X-shear value be -1 a Y shear value be 1. Compare the shapes of the pentagon objects before and after shear transformation.

6.2 Homogeneous Coordinate System

Homogeneous coordinate system is developed based on the concept of infinity. With respect to point coordinate system, infinity is a point that does not exist. However, many mathematicians have mathematically proved that, with the concept of infinity many geometric concepts and computations can be simplified and computed easily. Homogeneous coordinate system is one such mathematical technique that makes use of the concept of infinity to work with 2-D objects in computer graphics and computer-aided design.

Consider two real numbers 'b' and 'z'.

p=b/z, where 'p' is a real number.

Let the value of 'b' be constant and the value of 'z' be varied, i.e., as the value of 'z' decreases the value of 'p' increases. When the value of 'z' tends to zero the value of 'p' tends to infinity.

Therefore, if the value of 'z' is non-zero then the value of 'p' is equal to 'b/z' and is represented as p= (b, z). If the value of 'z' is zero, the value of 'p' is infinity and is represented as p = (b, 0). Thus, the infinity concept can be represented as (b, z) or b/z.

When this concept is applied to the xy-coordinate plane, the x and y is replaced with x/z and y/z respectively, i.e. a function f(x, y) =0 becomes f(x/z, y/z) =0.

Consider a line represented in an equation form, Ax + By + C =0. Replace x and y with x/z and y/z respectively. The equation now becomes,

A(x/z) + B(y/z) + C =0

Multiply the above equation by z, then it becomes,

Ax + By + Cz =0.

This is for first order polynomial equation of a line.

The second order polynomial equation of a line is given as,

$Ax^2 + 2Bxy + Cy^2 + 2\text{-}Dx + 2Ey + F = 0$

In the above equation replace x and y with x/z and y/z respectively and multiply it with z^2, then the equation becomes,

$Ax^2 + 2Bxy + Cy^2 + 2\text{-}Dxz + 2Eyw + Fz^2 = 0$

From the above two polynomial equations, it is clear that the degree of all terms are equal, i.e. in a first degree polynomial the terms x, y, and z are of degree one. And in second degree polynomial all terms x^2, y^2, xy, xz, yz and z^2 are of degree two. Hence, we observe that after introducing z in a polynomial of degree n, the terms of the polynomial still hold the degree n. Such polynomials are known as homogeneous polynomials and [x y z] are the homogeneous coordinates.

Linear transformations such as translation, scaling, rotation, and shearing can be easily described using homogeneous coordinate system. The point coordinate (x, y) in two-dimensions is represented by a vector of three numbers [a b c]. The three components of the vector are interpreted as coordinates in a three-dimensional space. Therefore, a point coordinate (x, y) in two-dimensions is transformed into a homogeneous coordinate system by using a non-zero number z. The homogeneous representation of (x, y) is [zx zy z].

Here, the number z is known as the homogeneous coordinate or scale factor.

Notes

The point coordinate values can be obtained from the homogeneous representation by dividing the elements of homogeneous representation by the homogeneous coordinate z.

1. Point coordinates **to** Homogeneous representation:

   (x, y) ➔ [zx zy z]

2. Homogeneous representation **to** Point coordinates:

   [a b c] ➔ (a/c, b/c)

Example: Consider a point coordinate (2, 3). This point coordinate is transformed to homogeneous representation as [2 3 1]. Here, the value of homogeneous coordinate, 'z' is equal to '1'. If the value of z=2, then the homogeneous representation is [4 6 2].

The transformation such as scaling and rotation can be done by multiplying the point coordinate matrix with the scaling and rotation matrix respectively, and translation transformation is obtained by adding the translation matrix with the point coordinate matrix.

This is a major limitation, since different types of matrix operations have to be performed for different transformations, i.e., if you need to perform transition, scaling, and rotation, you have to first apply transition transformation to the object, followed by scaling transformation and then rotation transformation.

If all transformations are combined into one matrix, it will be easy to perform series of transformations by using a single transformation matrix. This helps the graphic designer to write efficient code to perform 2-D transformations. This can be achieved only if all the matrices are of same size. However, in linear transformation, scaling and rotation matrices are of the size 2X2 and translation matrix is of the size 2X1.

With the use of homogeneous coordinate system, it is easy to convert matrix of different sizes into matrices of same size. The basic transformations in 2-D matrices can be converted into matrices of size 3 x 3 using homogeneous coordinate system. The transformation matrices of various transformations are as follows:

Translation matrix = $\begin{pmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{pmatrix}$

Scaling matrix = $\begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$

Rotation matrix = $\begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$

X-Shear matrix= $\begin{pmatrix} 1 & a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

Y-Shear matrix= $\begin{pmatrix} 1 & 0 & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

The point coordinate matrix is represented as

$$P = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

The homogeneous coordinate matrices of point coordinate values are multiplied (matrix multiplication) with the homogeneous transformation matrices to perform the respective transformation. The point coordinate matrix after the transformation is represented as:

$$P' = \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix}$$

Notes

The matrix multiplication for 3X3 matrix is as show,
Let A and B are two 3X3 matrices,

$$A=\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$$

$$B=\begin{pmatrix} k & l & m \\ n & o & p \\ q & r & s \end{pmatrix}$$

$$A*B=\begin{pmatrix} a.k+b.n+c.q & a.l+b.o+c.r & a.m+b.p+c.s \\ d.k+e.n+f.q & d.l+e.o+f.r & d.m+e.p+f.s \\ g.k+h.n+i.q & g.l+h.o+i.r & g.m+h.p+i.s \end{pmatrix}$$

The resultant matrix after multiplication is a 3X3 matrix.
If matrix A is 3X3 and B is 3X1 then the multiplication is carried out as shown,

$$A=\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$$

$$B=\begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

$$A*B=\begin{pmatrix} a.x+b.y+c.z \\ d.x+e.y+f.z \\ g.x+h.y+i.z \end{pmatrix}$$

The resultant matrix after multiplication is a 3X1 matrix.

Example: The 2-D point coordinate (2, 4) is scaled by a scaling factor 2 in X and Y directions of the 2-D space.
To apply the transformation we need to obtain the homogeneous representation of the point coordinates.
The homogeneous representation of (2, 4) is [2 4 1]. This is represented in matrix form as,

$$P=\begin{pmatrix} 2 \\ 4 \\ 1 \end{pmatrix}$$

$$\text{Scaling factor}=\begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Let P′ be the matrix value after scaling,
P′ = Scaling factor * P

$$=\begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 2 \\ 4 \\ 1 \end{pmatrix}$$

$$P' = \begin{pmatrix} 4 \\ 8 \\ 1 \end{pmatrix}$$

Therefore the value of P′ in homogeneous representation is [4 8 1]. The point coordinate (2, 4) after scaling by a factor=2 is (4, 8).

Caution

We have learnt in the beginning of the unit that translation transformation is performed by adding translation matrix and point coordinate matrix. However, in homogeneous coordinate system, the translation transformation is performed by multiplying the transformation matrix and homogeneous representation of the object's point coordinate values.

6.3 Composite and Inverse Transformations

The linear transformations that are represented using matrices can be easily composed, i.e., combined. It can also be inverted or reversed, i.e., to get the original object from the transformation. The use of matrix and homogeneous representation helps to perform transformations such as translation, scaling, rotation, reflection, and searing using a single matrix.

Composite Transformation

Now let us learn how to compose or combine two transformations? This can be achieved by simple matrix multiplication. Consider A and B as the matrices of any two linear transformations which you want to apply to a 2-D object. The effect of applying 'A' transformation first and then 'B' transformation to the object can be achieved by multiplying A and B transformation, and then applying the resultant transformation to the object.

Example:

Consider that the rectangle object with coordinate values represented in homogeneous coordinate system representation are [0 0 1], [2 0 1], [2 2 1] and [0 2 1]. The rectangle has to be scaled by a scaling factor 0.5 in X and Y directions. After scaling the matrix it has to be rotated by an angle 90 degree counterclockwise about the origin.

The scaling matrix is given by,

$$\text{Scaling matrix} = \begin{pmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

The coordinates of the rectangle after applying scaling transformation are [0 0 1], [1 0 1], [1 1 1], and [0 1 1].

Rotation matrix to obtain the rotation of 90 degree counterclockwise,

$$= \begin{pmatrix} \cos 90 & -\sin 90 & 0 \\ \sin 90 & \cos 90 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

The rotation transformation is applied to the scaled matrix. The coordinates of the scaled rectangle after applying rotation transformation are [0 0 1], [0 1 1], [-1 1 1] and [-1 0 1].

Figure 6.9 shows all the three rectangle objects, i.e., original object, original object after scaling transformation, and rotation of the scaled object.

The same result can be obtained by multiplying the scaling matrix and rotation matrix and then multiplying the point coordinate matrix values of the rectangle

with the product matrix of scaling and rotation matrices.

The product of scaling matrix and rotation matrix:

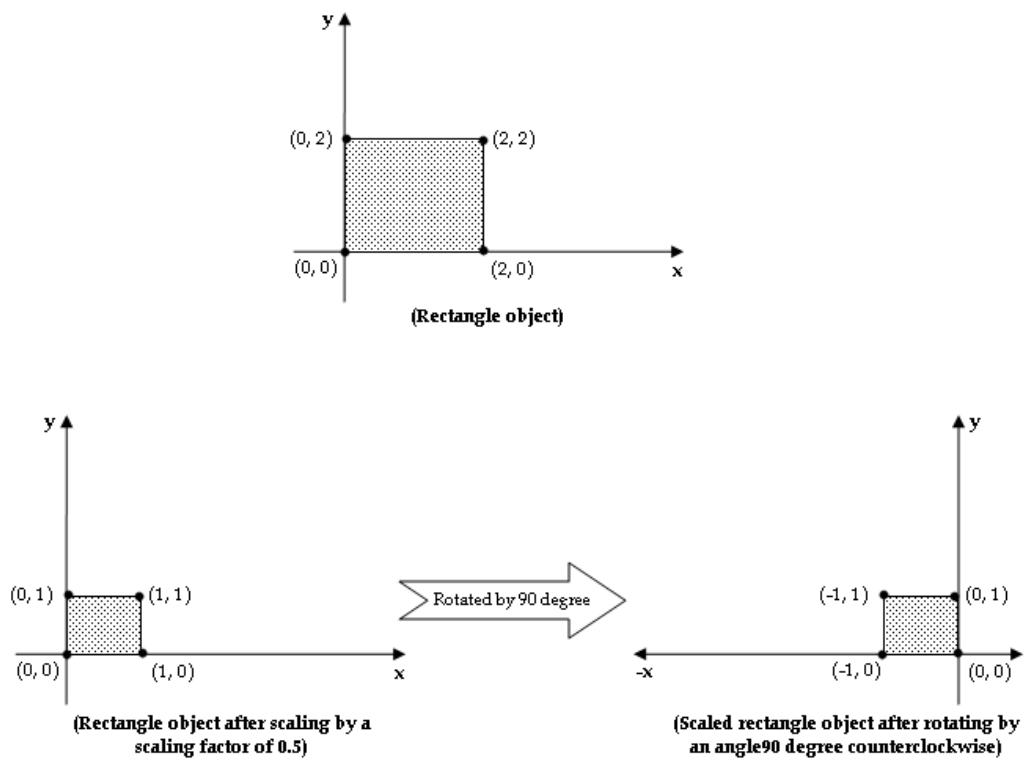
$$T = \begin{pmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & -0.5 & 0 \\ 0.5 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

where, T is the product of scaling matrix and rotation matrix.

When the homogeneous representation of coordinates of the rectangle [0 0 1], [2 0 1], [2 2 1] and [0 2 1] are multiplied individually with T matrix then the following coordinates are obtained [0 0 1], [0 1 1], [-1 1 1] and [-1 0 1] respectively. By comparing these values with the values obtained when both the transformations were carried out independently one after the other, is found to be the same.

Therefore, the example proves that, two or more transformations can be combined with the help of matrix multiplication.

Figure 6.9: Scaling of a Rectangle Object



Task

Apply scaling, translation, rotation, and X-shear transformation to a rectangle using homogeneous coordinate system.

Inverse Transformation

It is possible to obtain the original coordinate values of an object from its transformation. This is achieved using inverse matrix. The matrix multiplication of the coordinate values of the transformed object with the respective inverse transformation matrix, gives the original coordinate values of the object.

Let us assume that P′ is a point coordinate of an object after transformation, T is the transformation matrix, and P is the original coordinate of the object. The inverse of a transformation matrix T is T^{-1}.

Then, $P = T^{-1}.P'$

The inversion transformation matrices of various transformations are:

Inverse translation matrix = $\begin{pmatrix} 1 & 0 & -d_x \\ 0 & 1 & -d_y \\ 0 & 0 & 1 \end{pmatrix}$

Inverse scaling matrix = $\begin{pmatrix} \frac{1}{S_x} & 0 & 0 \\ 0 & \frac{1}{S_y} & 0 \\ 0 & 0 & 1 \end{pmatrix}$

Inverse rotation matrix = $\begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$

Inverse X-shear matrix= $\begin{pmatrix} 1 & -a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

Inverse Y-shear matrix= $\begin{pmatrix} 1 & 0 & 0 \\ -b & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

Example: The homogeneous coordinates of a rectangle scaled by a scaling factor 0.5 in X and Y directions are [0 0 1], [1 0 1], [1 1 1], and [0 1 1]. To find the homogeneous coordinates of the original rectangle (i.e. before scaling), the homogeneous coordinates of the scaled rectangle need to be individually multiplied with the inverse scaling matrix.

Let the inverse scaling matrix with the scaling factor 0.5 in X and Y direction be S,

$$S = \begin{pmatrix} \frac{1}{0.5} & 0 & 0 \\ 0 & \frac{1}{0.5} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$S = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

By multiplying the homogeneous coordinates of the scaled rectangle with the

inverse scaling matrix S individually, the homogeneous coordinates of the original object (object before scaling) is obtained. The coordinates are [0 0 2], [2 0 1], [2 2 1], and [0 2 1].

Therefore, with the help of inverting transformation the original 2-D object can be obtained from the transformed 2-D object.

Notes

The inverse of a 3X3 matrix is calculated as shown.
Let A be a 3X3 matrix,

$$A=\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$$

$$A^{-1}=\frac{1}{|A|}\begin{pmatrix} \begin{vmatrix} e & f \\ h & i \end{vmatrix} & \begin{vmatrix} c & b \\ i & h \end{vmatrix} & \begin{vmatrix} b & c \\ e & f \end{vmatrix} \\ \begin{vmatrix} f & d \\ i & g \end{vmatrix} & \begin{vmatrix} a & c \\ g & i \end{vmatrix} & \begin{vmatrix} c & a \\ f & d \end{vmatrix} \\ \begin{vmatrix} d & e \\ g & h \end{vmatrix} & \begin{vmatrix} b & a \\ h & g \end{vmatrix} & \begin{vmatrix} a & b \\ d & e \end{vmatrix} \end{pmatrix}$$

Here,

$|A| = \{ a[e.i-h.f]-b[d.i-g.f]+c[d.h-e.g]\}$

$\begin{vmatrix} e & f \\ h & i \end{vmatrix}=[e.i-f.h]$

Therefore,

$$A^{-1}=\frac{1}{a[e.i-h.f]-b[d.i-g.f]+c[d.h-e.g]}\begin{pmatrix} e.i-f.h & c.h-b.i & b.f-c.e \\ f.g-d.i & a.i-c.g & c.d-a.f \\ d.h-e.g & b.g-ah & a.e-b.d \end{pmatrix}$$

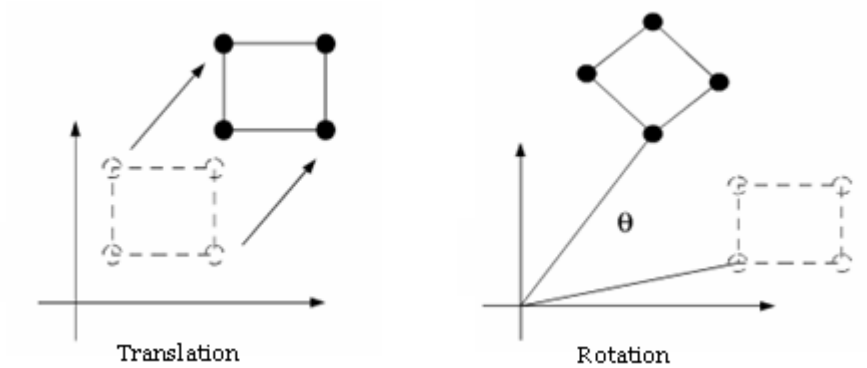
6.4 Affine Transformation

Transformations such as translation, scaling, rotation, and shearing are all affine transformations. When 2-D objects are transformed they may acquire different shapes. Affine transformation helps to overcome such problems. Affine transformations transform lines into lines, rectangle into rectangle, triangle into triangle, and so on.

An affine transformation can be defined as a transformation that fixes some points of the object and transforms other points of the object. This transformation preserves proportions on lines, however after transformation the angles or lengths of the object can vary.

The figure 6.10 depicts translation and rotation.

Figure 6.10: Translation and Rotation



As shown in figure 6.10, the square object is transformed using translation transformation. The rectangle remains as a rectangle even though the position of the rectangle is changed. Similarly, as shown in figure 6.10, when the rectangle is rotated using rotation transform, the angle of the lines with respect to the origin varies, but the rectangle remains as a rectangle. Such transformations are called affine transformation.

Affine transformation is useful while performing combination transformation. The combining of rotation and scaling transformation can also be called as rotation-enlargement transformation. Consider point coordinates P=(x, y).

When rotation-enlargement transformation is applied to this point, the transformation point P'=(x', y').

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = s \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Where, 's' is the scaling factor and α is the angle of rotation.

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = s \begin{pmatrix} x.\cos\alpha - y.\sin\alpha \\ x.\sin\alpha + y.\cos\alpha \\ z \end{pmatrix}$$

The equation is simplified as,

$x' = s\left(x.\cos\alpha - y.\sin\alpha\right) = (s.\cos\alpha)\,x - (s.\sin\alpha)\,y$

$y' = s\left(x.\sin\alpha + y.\cos\alpha\right) = (s.\sin\alpha)\,x + (s.\cos\alpha)\,y$

$z' = s\,(z)$

Let,

$a = s.\cos\alpha$

$b = -\,s.\sin\alpha$

Then,

$x' = ax + by$

$y' = ax - by$

The scaling factor s is given by

$s=\sqrt{a^2+b^2}$

The angle of rotation is given by,

$\alpha = \tan^{-1}\left(-b/a\right)$

Therefore,

The affine transformation of R^n is of the form R^n.

i.e., f(x): $R^n \rightarrow R^n$

F(p)=Ap+B

Where, A is the transformation applied to the point P and B is a constant.

Thus, the affine transformation can be used to perform transformation by combining two or more transformations simultaneously.

6.5 Summary

- 2-D transformations are carried out by representing coordinate values of the 2-D object in matrix form.
- Various transformations that can be applied to 2-D objects are translation, scaling, reflection and shearing.
- Translation transformation moves the object from one position to another in the 2-D space.
- The transformation is carried out by multiplying the coordinate values of the object individually with the respective transformation matrix.
- Scaling transformation is done to resize and reshape the object. The object's dimension can be varied using scaling transformation.
- Reflection transformation is used to obtain the mirror image of the object.
- The object can be rotated with respect to the origin, clockwise or counterclockwise using rotation transformation.
- Shear transformation is used to slant the image in X and Y directions of the 2-D space.
- Homogeneous transformation matrix is derived using the homogeneous coordinate system. This matrix represents 2-D objects as 3-D objects by including a non-zero number as third element.
- Homogeneous transformation matrix is used to combine any two transformations by simply multiplying the transformation matrices.
- The inverse of the transformation matrix is used to derive the original object from transformed object.
- Linear transformations such as translation, scaling, reflection, rotation, and shearing are called affine transformations.

6.6 Keywords

***Affine Transformation*:** This is the combination of same transformations such as translation, rotation or reflection on an axis.

Degree of a Polynomial: It is the highest degree for a term with non-zero coefficient in a polynomial that is expressed as the sum or difference of terms.

***Homogeneous*:** The different parts or elements that are of same kind.

***Shear Transformation*:** This is the transformation in which one coordinate is fixed and other coordinate or coordinates are shifted.

6.7 Self Assessment

1. State whether the following statements are true or false:

   (a) Translation transformation is used to move the object in the X and Y directions of the 2-D space.

   (b) The object is scaled in the X direction of the 2-D graph, by adding all the X coordinate points of the object by a scaling factor Sx.

   (c) The counterclockwise rotation matrix used for rotation transformation is $\begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$.

   (d) The shear transformation makes the object to slant either in X direction or Y direction and it is called Y-shear and X-shear respectively.

   (e) The basic transformations in 2-D matrices can be converted into matrices of size 3 x 3 using homogeneous coordinate system.

   (f) Scaling and shearing transformation can be combined using homogeneous coordinate system.

2. Fill in the blanks:

   (a) The translation matrix is ..... to the point coordinate matrix of the object.

   (b) .....transformation is done to resize the object.

   (c) The X-shear transformation matrix is represented in matrix form as.....

   (d) If (x, y) is a point coordinate then it is represented in homogeneous coordinate system as.....

   (e) The inverse transformation matrix of scaling is.....

3. Select a suitable choice for every question:

   (a) Identify the translation matrix.

   (i) $\begin{pmatrix} 0 & d_x \\ 0 & d_y \end{pmatrix}$

   (ii) $\begin{pmatrix} 0 & 0 \\ d_x & d_y \end{pmatrix}$

   (iii) $\begin{pmatrix} d_x & 0 \\ 0 & d_y \end{pmatrix}$

   (iv) $\begin{pmatrix} d_y & d_x \\ 0 & 0 \end{pmatrix}$

(b) Which among the following is the scaling matrix used in scaling transformation?

(i) $\begin{pmatrix} S_x & S_y \\ 0 & 0 \end{pmatrix}$

(ii) $\begin{pmatrix} S_x & 0 \\ S_y & 0 \end{pmatrix}$

(iii) $\begin{pmatrix} S_x & 0 \\ 0 & S_y \end{pmatrix}$

(iv) $\begin{pmatrix} 0 & S_y \\ S_x & 0 \end{pmatrix}$

(c) If the Y-shear factor is 1, identify the shear transformation matrix.

(i) $\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$

(ii) $\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$

(iii) $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

(iv) $\begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$

(d) Identify the translation matrix in homogeneous coordinate system.

(i) $\begin{pmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{pmatrix}$

(ii) $\begin{pmatrix} 1 & 0 & -d_x \\ 0 & 1 & -d_y \\ 0 & 0 & 1 \end{pmatrix}$

(iii) $\begin{pmatrix} 0 & 1 & -d_x \\ 1 & 0 & -d_y \\ 0 & 0 & 1 \end{pmatrix}$

(iv) $\begin{pmatrix} 0 & 1 & d_x \\ 1 & 0 & d_y \\ 0 & 0 & 1 \end{pmatrix}$

(e) Which among the following is the inverse transformation matrix of clockwise rotation transformation?

(i) $\begin{pmatrix} \cos\theta & \sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$

(ii) $\begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$

(iii) $\begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$

(iv) $\begin{pmatrix} \cos\theta & \sin\theta & 0 \\ \sin\theta & -\cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$

6.8 Review Questions

1. "2-D transformation is nothing but mapping or transformation a 2-D point." Explain.
2. "Translation transformation moves object from one position to another in 2-D space." Discuss.
3. How do you scale a 2-D object using transformation matrix?
4. "Reflection and scaling transformation are related." Explain.
5. "Reflection about Y axis is different from reflection about X axis." Do you agree? Justify.
6. "Rotation transformation rotates the point coordinates by an angle clockwise or counterclockwise." Explain with an example.
7. Do you think X-shear and Y-shear are different? Discuss.
8. "Homogeneous coordinate system represents 2-D object as 3-D objects." Explain.
9. Can you combine scaling and shear transformations? Discuss.
10. "The coordinates of rectangle object can be obtained after it undergoes X-shear transformation." Explain.
11. "Scaling is affine transformation." Do you agree? Justify.
12. "The affine transformation of Rn is of the form Rn." Explain.

Answers: Self Assessment

1. (a) True (b) False (c) True
   (d) False (e) True (f) True

2. (a) Added (b) Scaling (c) $\begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix}$

(d) [x y w] (e) $\begin{pmatrix} \frac{1}{S_x} & 0 & 0 \\ 0 & \frac{1}{S_y} & 0 \\ 0 & 0 & 1 \end{pmatrix}$

3. (a) $\begin{pmatrix} 0 & d_x \\ 0 & d_y \end{pmatrix}$ (b) $\begin{pmatrix} S_x & 0 \\ 0 & S_y \end{pmatrix}$ (c) $\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$

(d) $\begin{pmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{pmatrix}$ (e) $\begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$

6.9 Further Readings

Books

Xu, J. (2009).Practical WPF Charts and Graphics. USA:Apress.
Vince, J.A. (2006). Mathematics for Computer Graphics. London: Springer-Verlag.
Foley, J.D. (2006). Computer graphics: Principles and practice, 2nd ed in C. USA: Addison-Wesely Publishing Company, Inc.

Online link

http://www.willamette.edu/~gorr/classes/GeneralGraphics/Transforms/transforms2-D.htm
http://www.ia.hiof.no/~borres/cgraph/math/twod/p-twod.html
http://en.wikipedia.org/wiki/Transformation_matrix
http://www.radiogirmit.com/
http://www.blancmange.info/notes/maths/vectors/homo/
http://www.geom.uiuc.edu/docs/reference/CRC-formulas/node15.html
http://mathworld.wolfram.com/AffineTransformation.html

Unit 7: 2-D Viewing

CONTENTS

Objectives

Introduction

7.1 Concept of Window and Viewport

7.2 Window-to-Viewport Mapping

7.3 Graphic Pipeline

7.4 Panning and Zooming

7.5 Summary

7.6 Keywords

7.7 Self Assessment

7.8 Review Questions

7.9 Further Readings

Objectives

After studying this unit, you will be able to:

- Comprehend concept of window and viewport
- Describe window to viewport mapping
- Explain graphic pipeline
- Define panning and zooming

Introduction

2-D viewing, otherwise known as two-dimensional viewing, is a process used to view an image from a specific perspective. In this era, a computer can perform all graphic works. With the coming of computer graphics, it is easier to read, understand, and depict any type of image. In few years, we may also come up with software or application that can help us control the computers through icons and pictures rather than just by typing. Graphics images can be depicted either in a two-dimensional or three-dimensional plane.

With the improvement in technology, two-dimensional and other graphic types have become common. However 2-D is still the widely used graphic type. Two-dimensional computer graphics are graphical images produced with the help of digital computers, in which two-dimensional visual techniques are used. 2-D graphics consist of 2-D geometric models, like image compositions, pixel art, digital art, photographs, and text. Traditional painting and drawing use 2-D graphics. 2-D computer graphics basically comprise two types; namely:

1. ***Raster Graphics:*** Raster graphics comprise arrays of pixels. Pixels can be of different color or shade. The editing of these graphics is done at the pixel level. Raster images are mostly used on old computer and video games, graphics related calculator games, and many mobile phone games.
2. ***Vector Graphics:*** Vector graphics comprises paths. Paths are used to illustrate the images by setting a mathematical relationship between points within an image. Vector graphics are used on photographic images.

Did you know? Two-dimensional computer graphics generally do not involve the need for any kind of three-dimensional internal representation of objects or characteristics in the computer. However, some 2-D software employs 3-D techniques and ideas.

This unit discusses the concept of window and viewport and how they are mapped. We will also discuss the different stages involved in the graphic pipeline. The panning and zooming concepts will also be discussed in this unit.

7.1 Concept of Window and Viewport

The region of the scene or an object bounded by a rectangular area is known as a window. A window simply means the frame outlines that define the part of a scene or object that is visible from a particular position at a particular time.

The zone of the monitor screen within which a selected object is viewed is known as a viewport. The zone is termed as viewport since it defines the area of the screen which can be viewed. The viewport is generally seen as a rectangular area of the monitor screen in which the selected view is seen.

Notes A window or viewport can be of any shape. However, the analysis for shapes other than a rectangular area is complex. The overall method of analysis for all shapes will be similar to the method of analysis followed for rectangles.

Window Definitions

There are few ways of defining a window. Firstly, a window should be described in any length dimension, both horizontally and vertically.

Example: Feet, meters and miles

As a common practice, the corners of a window are defined with reference to the world coordinate origin. The dimensions are set into the computer as (x, y) data and subsequently changes will be applied to the computer model.

The figure 7.1 depicts a window and a viewport.

Figure 7.1: A Window and Viewport

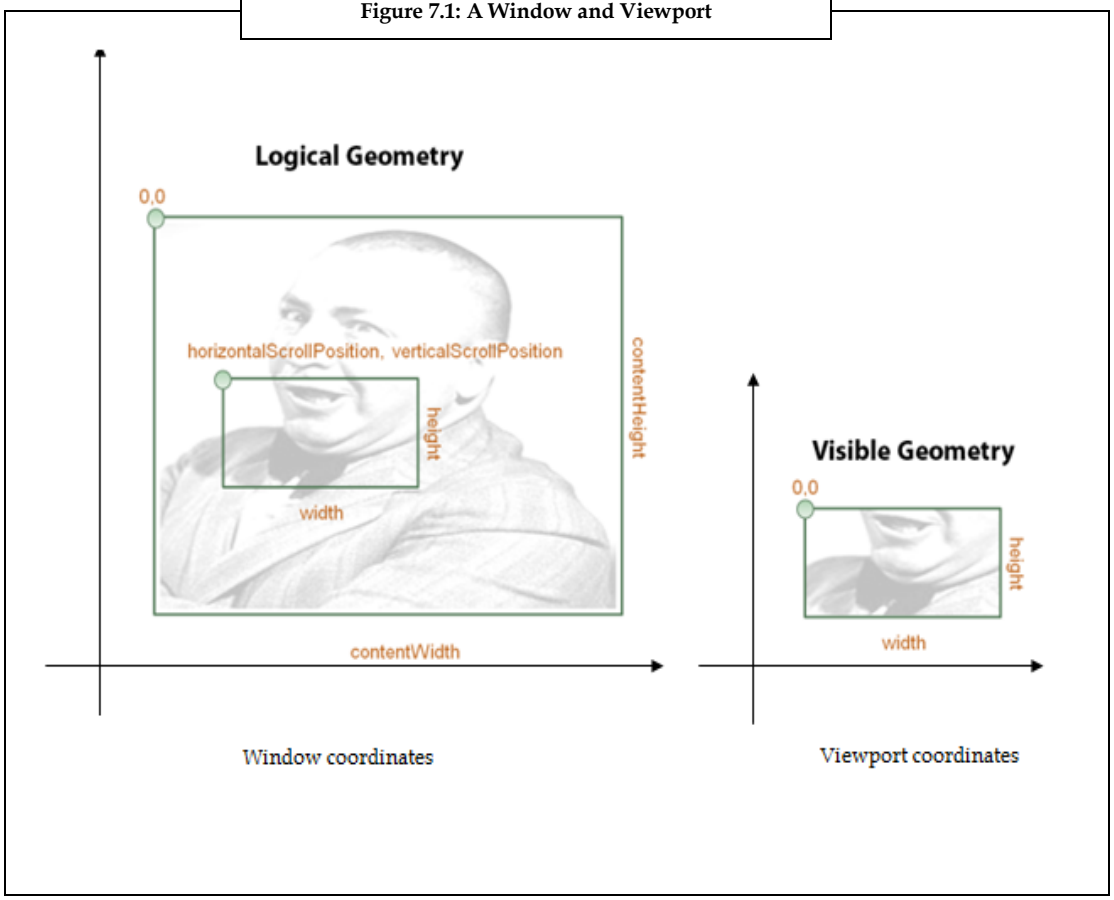


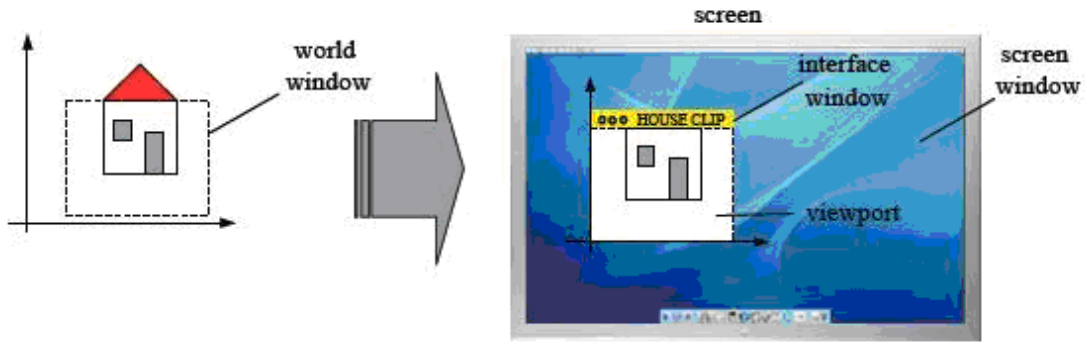
Figure 7.1, illustrates the window coordinates, which include the logical geometry whereas viewport coordinate include the visible geometry out of logical geometry. The sequence given is a common sequence of the window limits. It shows the selection of visible geometry from the logical geometry. In most of the graphic packages, the order of input is defined.

Viewport Definitions

A viewport is defined in screen coordinates (X, Y). A viewport can also be defined in the Interface Coordinates. The benefit of defining it in an Interface coordinate is that the viewing transformation relationship can be derived on this basis and can be applied to any screen and resolution.

The figure 7.2 depicts an object defined in the interface and viewport coordinates.

Figure 7.2: An Object Defined in the Interface and Viewport Coordinates



Source: http://www.cs.mtsu.edu/~jhankins/files/4250/notes/WinToView/WinToViewMap.html

Notes

Windows can also be termed as co-ordinates.

Figure 7.2 illustrates how to project a part of the image using viewport coordinates. Here, the part of the image that is required to be projected on the screen window is selected out of the interface window.

Window-viewport Relationship

The window-viewport relationship can be explained with the help of a figure. Figure 7.3 depicts an image viewed from a window and the same image viewed in a viewport.

Figure 7.3: An Object in Window and an Image in Viewport

Window

In figure 7.3, (a) symbolizes the clear rectangle within which the landscape is visible. This is known as the Window. The image which is seen through a window is the restricted insight of the world and is generally depicted in a rectangular format. The rectangular area is defined in length dimensions, with reference to origin, to which the object space is corresponded. The object space refers to the space occupied by the object in a 2-D or 3-D plane.

Example: Consider a moving train. As the train moves, the eyes move with relation to the train, and thus the view through the window also changes. The outside view is seen more or less to the right or left or it is seen more or less to the top or bottom depending on the shift of the view. Although the image remains constant.

The diagrams and other manual graphics of draftsmen and artists are built in the real world object spaces (space occupied by object in the real world), but the window need not be an opening to the real world. It might be an artificial illustration like a drawing or a photograph of a real object, a graphical depiction of an abstract concept, a scientific diagram, or a statistical chart. Therefore, a window is defined as a view of an image or an object represented through computer graphics.

Viewport

A viewport is used to depict the image of any object in the window. The image can be of any solid object or a portion of it. In figure 7.3, (b) depicts a viewport as a screen of a computer monitor. A user can view the part of the image as viewed through a window. Viewport is defined as the part of the screen in which the window scene is displayed. The viewport can also be defined as the representation of a window in a two-dimensional medium.

7.2 Windows-to-Viewport Mapping

In the earlier section we discussed the concept of window and viewport. In this section we will learn the mapping of an object from a window onto the viewport.

The process of transforming a two-dimensional world coordinate to device coordinate is known as window to viewport mapping. The objects present in the world or clipping window are mapped to the viewport. The interface window on the screen displays the viewport. To display a part of the scene, the clipping window is selected and then the viewport positions the scene on the output device.

Figure 7.4 illustrates the mapping of an image.

Figure 7.4: Window-to-Viewport Mapping

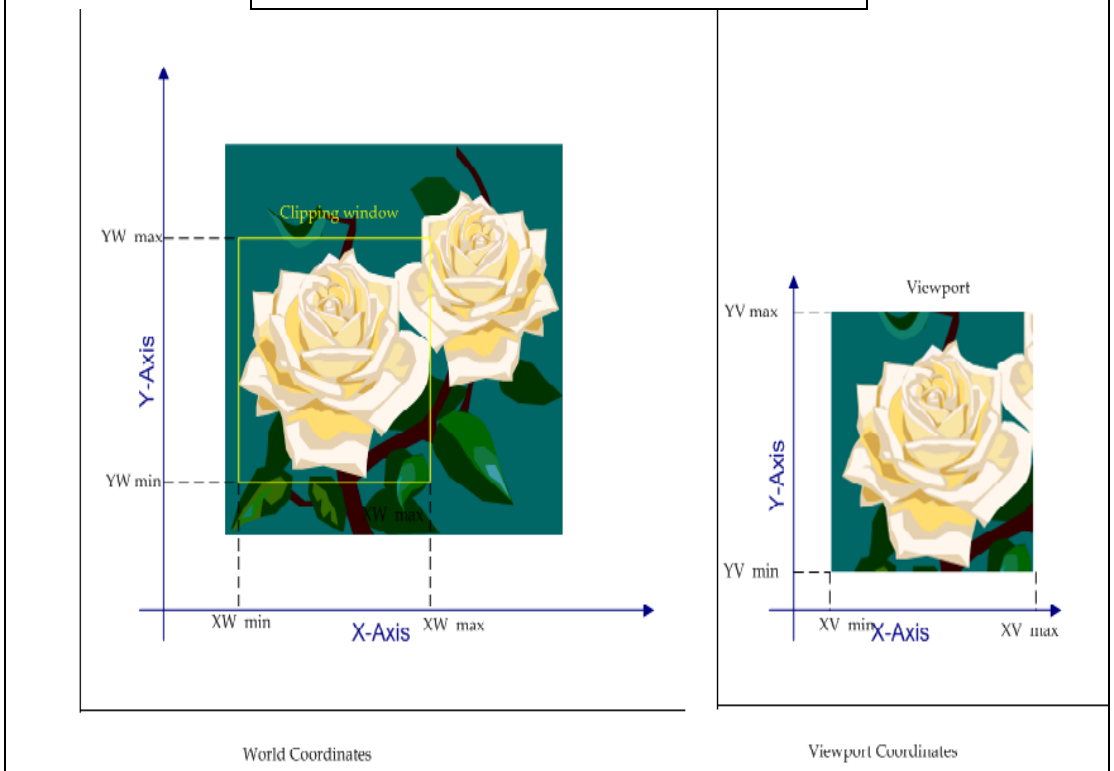


Figure 7.4 given above discusses how an image in the window is mapped onto the viewport.

Development of the Window-to-Viewport Mapping

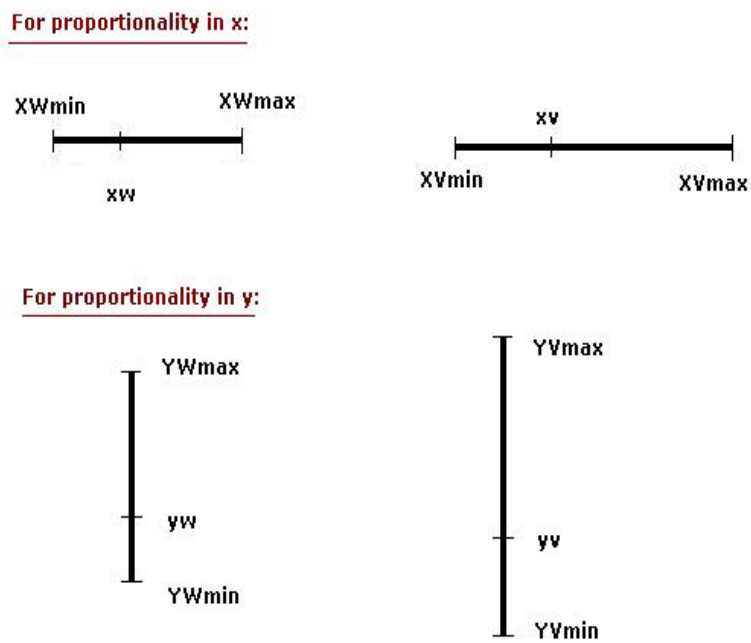
Window to viewport mapping is achieved through in-built developing formulae. Since, mapping of the object should be achieved in equal ratio. The formulae start at a point in the world window like (XW, YW). The formula helps to produce a corresponding point in the viewport coordinates, (XV, YV). The mapping has to be proportional.

Example: If XW is 40 per cent of the way from the right edge of the world window, then XV is 30% of the way from the right edge of the viewport. Similarly, if YW is 40per cent of the way from the top edge of the world window, then YV is 40 percent of the way from the top edge of the viewport.

The formula is used to produce a corresponding point in viewport coordinates, say (XV, YV). The mapping achieved is proportional. Let us assume that in figure 7.4, if XW as well as YW is 30% away from the bottom edge of the world window, then YV and XV of the viewport is also 30% away from the bottom edge.

This proportionality has been discussed with the help of figure 7.5.

Figure 7.5: Mapping Proportionality



The proportionality depicted in the figure above must be equal in the following ratios.

$$\frac{xv - xv_{\min}}{xv_{\max} - xv_{\min}} = \frac{xw - xw_{\min}}{xw_{\max} - xw_{\min}} \quad \text{…….. Eq.(7.1)}$$

Equation 7.1 indicates that mapping of object is equal in x axis of both the world coordinate and the viewport coordinate.

$$\frac{yv - yv_{\min}}{yv_{\max} - yv_{\min}} = \frac{yw - yw_{\min}}{yw_{\max} - yw_{\min}} \quad \text{……..Eq.(7.2)}$$

Equation 7.2 indicates that mapping of object is equal in y axis of both the world coordinate and the viewport coordinate.

After solving both equations 7.1 and 7.2 for the unknown viewport position (xv, yv), equations 7.3 and 7.4 become true:

$$xv = S_x xw + t_x \quad \text{…….Eq.(7.3)}$$

$$yv = S_y yw + t_y \quad \text{…….Eq.(7.4)}$$

The scale factors (Sx, Sy) are:

$$S_x = \frac{xv_{\max} - xv_{\min}}{xw_{\max} - xw_{\min}}$$

$$S_y = \frac{yv_{\max} - yv_{\min}}{yw_{\max} - yw_{\min}}$$

The translation factors (Tx, Ty) are:

$$t_x = \frac{xw_{\max} xv_{\min} - xw_{\min} xv_{\max}}{xw_{\max} - xw_{\min}}$$

$$t_y = \frac{yw_{\max} yv_{\min} - yw_{\min} yv_{\max}}{yw_{\max} - yw_{\min}}$$

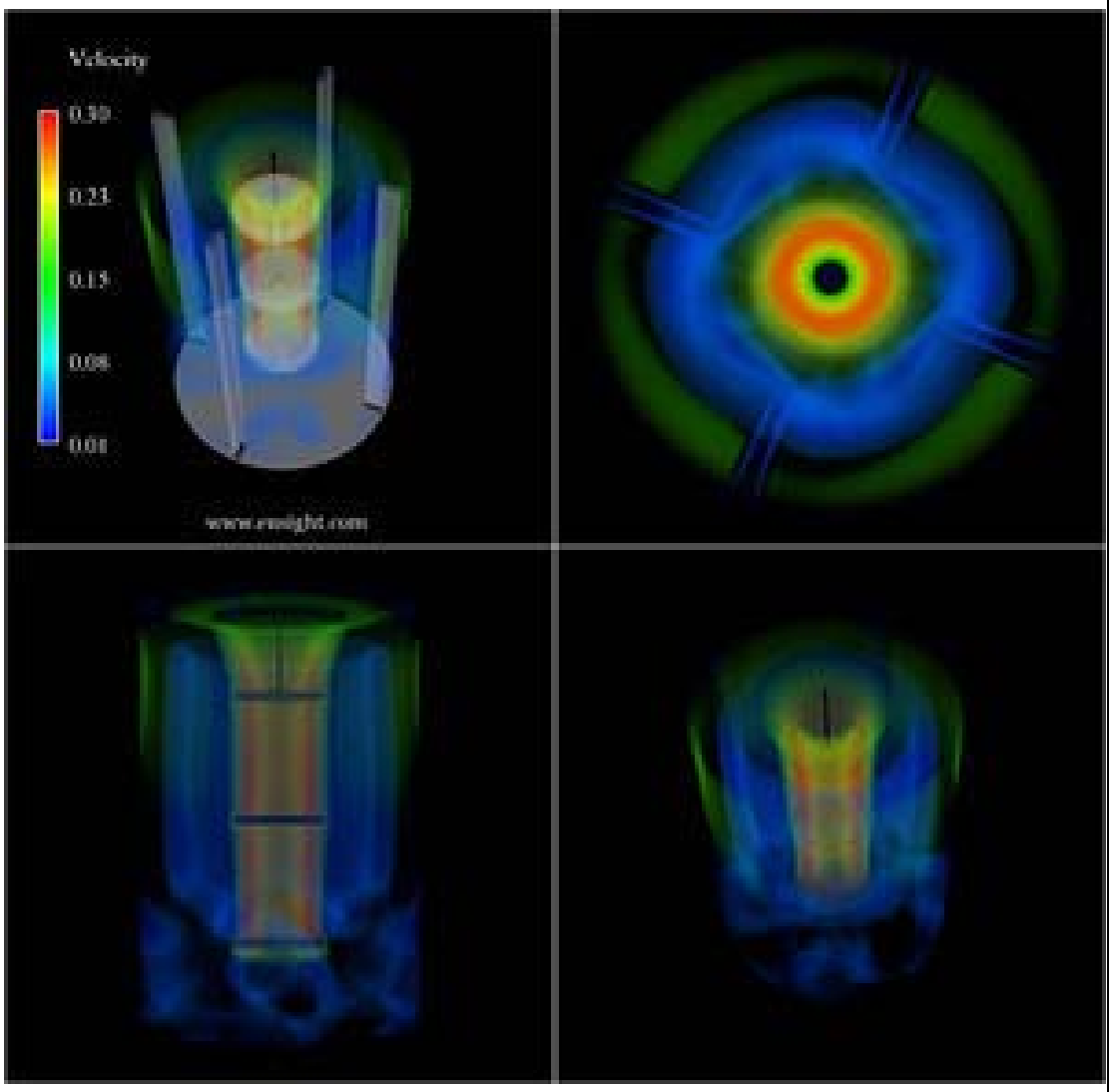
In the interface window, the position of the viewport can be changed by allowing objects to be viewed at different positions. Multiple viewports can also be used to display different sections of a scene at different screen positions. To manipulate the size and proportion of the object, the dimensions of the viewport must be changed. By mapping different dimensioned clipping windows on a fixed sized viewport, a zooming effect is being given to the object. The image will have a vague look if the aspect ratio of the world window and the viewport are different.

Tiling Using the Window-to-Viewport Transformation

Tiling is defined as the number of copies drawn of the same image in rows and columns across the interface window so that they cover the entire window. A picture that has been copied several times is called a motif. In computer graphics, tiling is acquired by keeping the window static and changing the viewport several times. The picture is redrawn every time the viewport is changed.

Figure 7.6 depicts an image that has been redrawn in different viewports in 2-D plane.

Figure 7.6 : Image Drawn Several Times on Different Viewports



7.3 Graphic Pipeline

A graphic pipeline is defined as a series of stages which helps a user to create a digital image of a model.

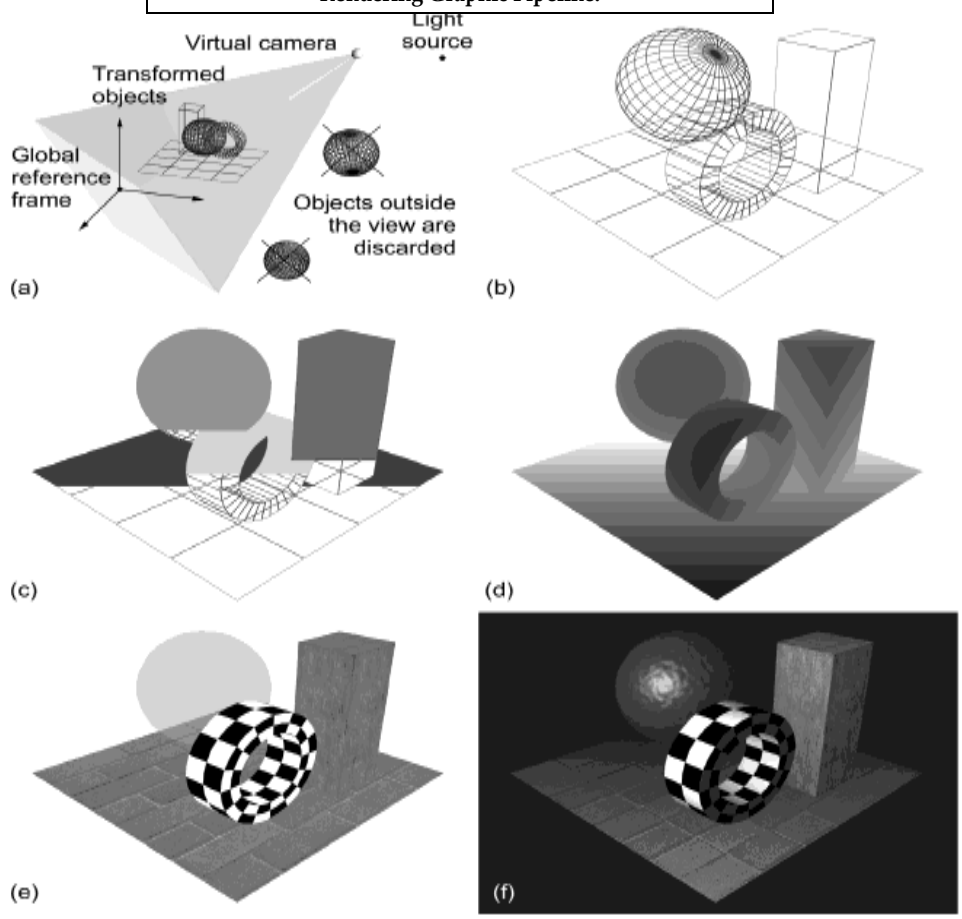
A graphical pipeline comprises the following stages:

1. A bus interface or front end interface is the interface which helps in sending and receiving data.
2. A vertex processing is the process which converts each vertex to a two-dimensional screen positioning.
3. A clipping is a process which helps in removing parts of the image that need not be visible on the screen.
4. A primitive assembly is a stage used to set up vertices that are collected and converted into triangles.

5. The rasterization is a process wherein all the triangles are filled with pixels known as fragments. Fragments can result in a frame buffer if there is a change or if it is not hidden.
6. The occlusion is a stage in which the pixels that are hidden are removed.
7. The parameter interpolation is a process in which the values of each pixel that were rasterized are computed on the basis of color, texture, and fog.
8. The pixel shader stage is the stage wherein the texture and final colors are added to the fragments.
9. The pixel engine is the stage in which mathematically the final color, its coverage and transparency is combined. The result is a depth value for the pixel.
10. The frame buffer controller stage is the last stage which acts as an interface to the memory used to hold the actual pixel values as displayed on the second screen.

Example: Figure 7.7 illustrates operations on primitives in the standard direct rendering graphic pipeline.

Figure 7.7: Operations on Primitives in the Standard Direct Rendering Graphic Pipeline.



Source :
http://docs.google.com/viewer?a=v&q=cache:pBO3osoKvBEJ:graphics.cs.aueb.gr/cgvizbook/slides/CGVIZ_Unit_1.pdf+graphic+pipeline+algorithm+%282-D%29&hl=en&gl=in&pid=bl&srcid=ADGEESibvWTGkvSDJHqKY4KSinleD5yeDYYosXTGmEWs3zinC9z4qejXM8ArGW-cSamvNHDRlzHER_Dvzn_6Vsc1ilHKXHRstK2f7mDchJHqGwr97sGx7BJvZM-GnU50AUZtjJ9dEAnq&sig=AHIEtbSMHAGSzELxiTpAAwc9dTfUYMmSpg

In figure 7.7, the following stages are covered:

1. Geometry transformation is done to a common reference frame and view frustum culling.
2. The primitives are done after viewing transformation, projection, and backface culling.
3. The rasterization is the next stage.
4. The fragment depth sorting is done where the darker a shade, the nearer the corresponding point is to the virtual camera.
5. The material color estimation is done after the fragment depth sorting.
6. The shading and other fragment operations are prepared such as fog, and so on.

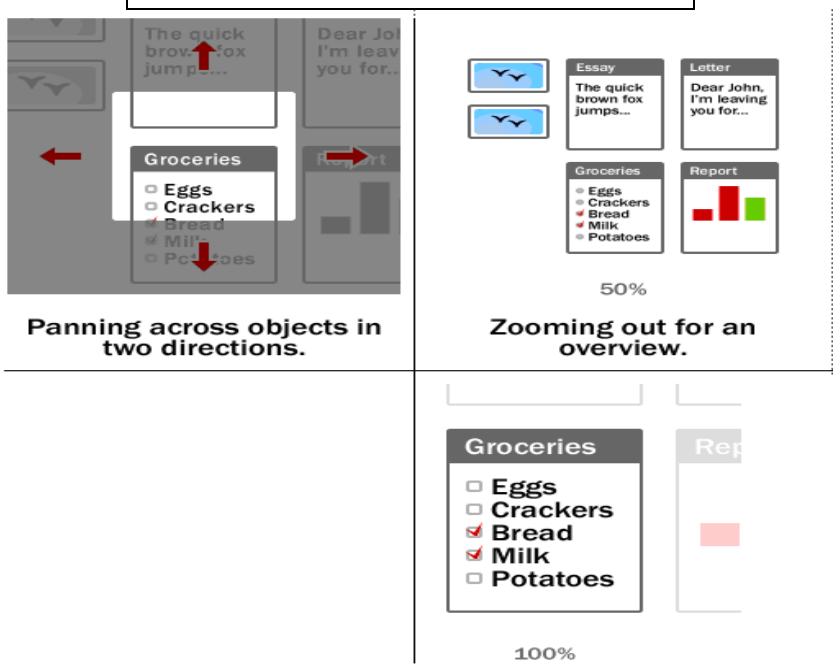
Thus, a clear view is obtained at the end of the pipeline process.

7.4 Panning and Zooming

Did you know? Panning is technique where the central objective is to get a clear image rather than its background. However, if the result is a little blurred, then it adds to the feeling of motion in the shot.

Figure 7.8 gives us an insight into the effects of panning and zooming on objects.

Figure 7.8: Effects of Panning and Zooming on Objects



Source: http://www.cs.mtsu.edu/~jhankins/files/4250/notes/WinToView/WinToViewMap.html

Panning

The horizontal movement of a camera when a camera scans an object is called as panning. In technical terminology, panning is nothing but moving a camera in synchronization with the subject which is moving in a parallel direction.

Example: Consider a person who wants to capture the image of an animal. As the animal flies by, the camera also should be moved in such a way that the movement of the subject matches the object's speed perfectly.

Appropriate panning refers to capturing motion. The feeling of motion and speed is generated by the panning technique. The object is shown clearly and the surrounding is shown as a vague impression. Figure 7.9 depicts the panning technique.

Figure 7.9 : Panning

Figure 7.9 depicts a person cycling. Notice the sharpness of the image. The surrounding is hazy but the image of the person cycling is clear.

Notes

The panning technique does not have any rule. Flash of the device can also be used while panning. Flash works only if the subject is near or if the flash is powerful enough to have an impact. But this can freeze the subject while giving the background motion blurred.

Caution

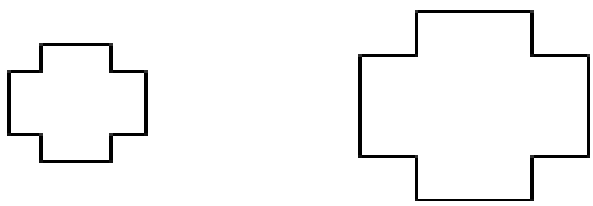
Be careful while approaching the panning technique for the first time. Do not use panning all day long. Sometimes shots must be done at faster speeds. This would result in giving variety of shots. It is not necessary that the subject will always have the clarity.

Zooming

Zooming is a technique where the size of the object can be changed to view the object in detail. Zooming is also known as a transformation which helps in scaling or blowing a pixel or a portion of the pixel with the instructions given by the user. The zooming process reproduces the pixels along the consecutive scan lines.

Example: Consider an image to be zoomed at the factor of two. This means each pixel is used four times, twice on each of the two successive scan lines. Figure 7.10 illustrates the example.

Figure 7.10 : Effect of Zooming by a Factor of Two



A Zooming User Interface (ZUI) is a graphical environment where users can change the scale of the viewed area in order to see more detail or less, and browse through different documents. The virtual surface can be panned across in two-dimensions by zooming the object.

Example: When a text object is zoomed, it may be represented as a small dot, then a thumbnail of a page of text, then a full-sized page and finally a magnified view of the page.

The ZUI uses zooming as the symbol for browsing through hyperlinked information. If a zoomed page has an object inside it, it is possible to further zoom the image to understand the image in detail. This allows recursive nesting and an arbitrary level of zoom.

If the detail of the resized object is changed, to fit the relevant information into the current size, then this process is known as semantic zooming.

Did you know? The ZUI interface paradigm is considered as a flexible and realistic successor to the traditional windowing GUI. At present, effort is spent in developing GUIs rather than developing ZUIs.

It is seen that the panning technique acts as a qualifier to the zooming transformation. Panning helps in moving the scaled portion of the image to the center of the screen. The entire screen is filled on the basis of the scale factor. By doing this, there is an increase in the zoom area in all the directions of the screen even if the image portion is close to the screen limit.

7.5 Summary

- Computer graphics comprises all the animations and pictures which is not text or sound.
- Raster graphics consist of arrays of pixels.
- Vector graphics consists of paths.
- The region of the scene or an object bounded by a rectangular area is known as a window.
- The zone of the monitor screen within which a selected object is viewed is known as a viewport.
- The process of transforming a two-dimensional world coordinate to device coordinate is known as window-to-viewport mapping.
- Tiling is described as the number of copies drawn of the same image in rows and columns across the interface window so that they cover the entire window.
- The graphic pipeline is characterized as a series of stages which helps a user to create a digital image of a model.
- The horizontal movement of a camera when a camera scans an object is called as panning.
- Zooming is a technique where the size of the object can be changed to view the object in detail.

7.6 Keywords

***Clipping*:** It is the act of cutting the object to the desired shape and size.

***Interpolation*:** The act of introducing or inserting between other elements or parts.

***Zooming*:** To cause text or graphics in a window to appear larger on the screen.

***ZUI*:** Zooming User Interface. A ZUI is a type of graphical user interface (GUI). A zooming user interface (ZUI) is used to change the size of the object. The information appears on the virtual desktop that is created using the vector graphics.

7.7 Self Assessment

1. State whether the following statements are true or false:

   (a) Zooming is a technique that helps users to slide the camera across the scene, taking in different parts of it at different times.

   (b) Raster graphics comprise arrays of pixels.

   (c) To manipulate the size and proportion of the object, the dimensions of the viewport must be changed.

   (d) Occlusion is a stage in which the pixels that are hidden are removed.

   (e) The zone of the monitor screen within which a selected object is viewed is known as a window.

   (f) A window should not be described in feet or meters or miles or in any length dimension horizontally and vertically.

2. Fill in the blanks:

   (a) ……. is a technique where the size of the object can be changed to view the object in detail.

   (b) A viewport is defined in ..... coordinates.

   (c) If the detail of the resized object is changed, to fit the relevant information into the current size, then this process is known as.....

   (d) A ..... is used to change the size of the object.

   (e) …… graphics comprises paths.

3. Select a suitable choice for every question:

   (a) Which of the following is a process where all the triangles are filled with pixels known as fragments?

      (i) Occlusion (ii) Rasterization

      (iii) Semantic zooming (iv) Tiling

   (b) The horizontal movement of a camera when a camera scans an object is called as:

      (i) Zooming (ii) Graphic pipeline

      (iii) Semantic zooming (iv) Panning

   (c) Which of the following is a process in which the values of each pixel, which were rasterized, are computed on the basis of color, texture, and fog?

      (i) Vertex processing (ii) Clipping

      (iii) Parameter interpolation (iv) Occlusion

   (d) A picture that has been copied several times is known as:

      (i) Motif (ii) Tiling

      (iii) Viewport (iv) Window

   (e) The corners of ..... are defined with reference to the world coordinate origin:

      (i) a window (ii) a viewport

      (iii) a graphic pipeline (iv)an image

7.8 Review Questions

1. "Zooming is a technique where the size of the object can be changed to view the object in detail." Explain.
2. "With the improvement in technology, 3-D and other graphic types have become common, but still 2-D is widely used." Discuss.
3. Discuss panning process in detail.
4. "The zone of the monitor screen within which a selected object is viewed is known as a viewport." Explain.
5. "A window should be described in feet or meters or miles or in any length dimension horizontally and vertically." Justify
6. "Graphic pipeline is defined as a series of stages which helps a user to create a digital image of a model." Discuss.
7. "The process of transforming a two-dimensional world coordinate to device coordinate is known as window to viewport mapping." Explain.
8. Discuss the window-to-viewport relationship with the help of an example.
9. "Zooming is a technique where the size of the object can be changed to view the object in detail." Discuss.
10. Discuss the different stages of a graphic pipeline.
11. "Mapping of object should be achieved in equal ratio." Justify.
12. "The panning technique does not have any rule." Discuss.

Answers: Self Assessment

1. (a) False (b) True (c) True (d) True (e) True (f) False
2. (a) Zooming (b) Screen (c) Semantic zooming (d) ZUI (e) Vector
3. (a) Rasterization (b) Panning (c) Parameter interpolation (d) Motif (e) Window

7.9 Further Readings

Books

N Krishnamurty, Introduction to Computer Graphics, Tata Mc Graw Hill
Malay. K. Pakhira, Computer Graphics, Multimedia and Animation
Zhigang Xiang, Roy. A. Plastock, Computer Graphics, Second Edition, Tata Mc GrawHill

Online link

http://graphics.stanford.edu/courses/cs448a-01-fall/lectures/lecture2/walk002.html

http://en.wikipedia.org/wiki/Computer_graphics

http://neohumanism.org/2/2-D/2-D_computer_graphics.html

http://ezinearticles.com/?2-D-and-3-D-Computer-Graphics&id=1462411

http://www.allgraphicdesign.com/graphics/comptutergraphics101.html

http://www.graphics.cornell.edu/online/tutorial/

http://www.digital-photography-school.com/mastering-panning-to-photograph-moving-subjects

http://ezinearticles.com/?Computer-Graphics---The-Graphics-Pipeline&id=3692113

Unit 8: 3-D in Computer Graphics

CONTENTS

Objectives

Introduction

8.1 3-D Representations

8.1.1 Splines

8.1.2 Surfaces

8.1.3 Curves

8.1.4 Add-ons: Polygon Rendering

8.2 3-D Transformation

8.2.1 Translation

8.2.2 Scaling

8.2.3 Rotation

8.2.4 3-D Clipping

8.3 3-D Viewing

8.3.1 The Viewing Pipeline

8.3.2 Camera Coordinates

8.3.3 Viewing Volume

8.4 Summary

8.5 Key words

8.6 Self Assessment

8.7 Review Questions

8.8 Further Readings

Objectives

After studying this unit you will be able to:

- Explain 3-D representations
- Elaborate on 3-D transformations
- Discuss 3-D viewing

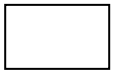
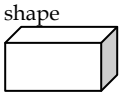
Introduction

Computer graphics is a vast field that composes of all the graphical images aspects. They can be categorized into 2-D, 3-D, 5-D and animated graphics. This unit mainly deals with the creation of images with respect to 3-D scenarios. Today, we have multiple applications for computer images like simulation and training, medical images, special effects for films, and so on.

Three dimensional graphics deal with the generation of images from nonrepresentational data of an object. The 3-D object models are projected on a 2-D plane. These models are converted to graphical images when they are displayed on the computer screen. They are not only limited to virtual space because they can also be printed. They are the realistic renderings of the surfaces, illumination sources, and so on.

The 3-D rendering process is implemented to display a model as 2-D image. The 3-D rendering process can also be used in fields that are not related to graphics like simulation. 3-D modeling deals with the creation of geometrical data for 3-D computer graphics.

Table 8.1 : Difference between 2-D and 3-D Technique

2-D Technique	3-D Technique
It is two-dimensional representation.	It is three-dimensional representation.
Virtual third dimension can be created which is static.	Third dimension is real, not virtual.
Includes typography, cartography, and advertising. Similar to painting.	Similar to photography and sculpting.
A 2-D shape 	A 3-D shape 

8.1 3-D Representations

A three-dimensional model displays an item or picture in such a way that it appears to be physically present on the screen of a computer. The item on the screen that appears flat (2-D item) to the human eye is displayed with dimensions that include height, width and depth.

3-D items are represented with the aid of digital computers and 3-D software. The two main Application Programming Interfaces (APIs) used for generating images are Open GL and Direct3-D. The creation of 3-D graphics can be divided into stages such as content creation, scene layout and rendering.

Content Creation

Content creation stage involves activities such as 3-D modeling, texturing and animation. The content that can be used in the next stage is usually created in this stage. It includes modifying the surface of object or material properties such as lighting, color, diffuse, and so on. The objects created in this stage are in elementary or basic form which is helpful in the process of animation. Modeling of the object is done with the aid of some software programs and applications.

Example: Software programs used for modeling are:

1. Moray
2. Light wave modeler
3. Poser, Bryce
4. City Engine

Scene Layout Setup

It is the arrangement of virtual objects, cameras, lights and other units that are involved in the production of still image or animation. This phase is known as key framing in animation. It aids in the creation of complicated movements in scenes. The objects are set up in frames and the stages are added into each of these frames.

Rendering

Rendering is the process of creation of actual 2-D or 3-D images from the scenes prepared in screen layout setup stage. For interactive media the rendering is taken at a rate of 20 to 120 frames (images) per second. It is comparatively slower in the case of non-interactive media.

Notes

Following are some of the rendering systems that can be used as plug ins to various 3-D applications:

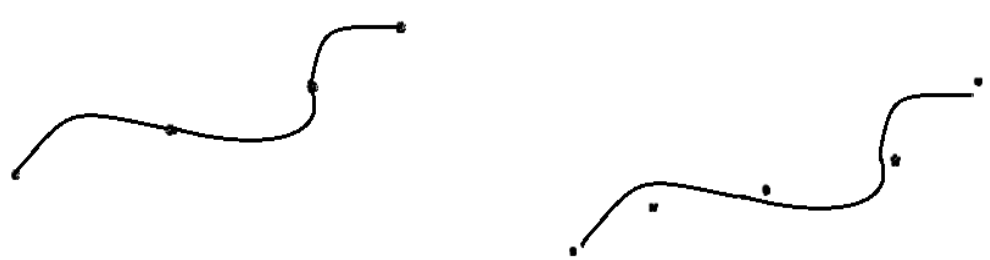
1. V-Ray
2. Yaf Ray
3. Indigo renderer
4. Mental ray
5. Maxwell

They are usually included in 3-D software packages

8.1.1 Splines

Spline is defined as a parametric curve that can be well-defined by set of coordinate positions called control points. A spline is a flexible strip and the term 'spline' is the term taken from engineering drawing and it is used to draw smooth curves through a specified set of points on the plane. These points are called control points and are adjusted by the user to regulate the shape of curve. The curve passes through all the points and it connects all the control points, this is also called interpolation. But if the polynomials require to be fitted to the path that does not pass through the points then the curve approximates the set of control points. The figure 8.1 displays the interpolation and approximation spline.

Figure 8.1 : Spline



Source:
http://wikihelp.autodesk.com/Product_Help/Autodesk_Inventor/Autodesk_Inventor_2011/103Parts/12322-Dsketches/1247Splines

The 3-D splines are curves whose radius keep changing constantly and pass through a series of control points. The spline points can be restricted in a partial or complete manner. These spline points can be added to the existing geometry while drawing the curve or at a later stage. The endpoints of a spline are square in shape and the control points are diamond shaped to give a clear understanding. The control points are known as ducks or knots.

The operations that can be performed on splines other than changing its shape are:

1. Adding points along the curve
2. Examining the radius of curvature along the curve
3. Showing the constraints
4. Changing the fit method
5. Closing open splines

Splines are used in computer animation to:

1. ***Define Spatial Shapes*** - the shapes of two or three-dimension objects. Here, the points on the surface of the object are the knots.
2. ***Define the Path of an Object through Space*** – Here, the points on the path are the knots.
3. ***Define Eases*** – Ease is the velocity of the movement along a path. There are two ways of defining the ease. Ease in is the slow movement at the beginning of the path and faster at a later stage. Ease out being the reverse.

In general, Modeling applications use spline. They are also used to pass the curve smoothly through the parameters. These parameters are known as sparse time.

Spline Representation

As we know spline is a flexible wooden strips tool that can be used to produce a smooth curve through a specified set of points. These points are represented mathematically with a cubic polynomial function called spline curves. The points in a spline have more than one coordinate. Splining of points together is to spline the entire x , y, and z coordinates together. So, it is expected to present a solution for one coordinate and apply the same on others during the splining process.

The following equation set can be used to describe a cubic polynomial that has to be fitted between every pair of control points:

$$\left.\begin{aligned} x(p)&=axp^3+bxp^2+cxp+dx \\ y(p)&=ayp^3+byp^2+cyp+dy \\ z(p)&=azp^3+bzp^2+czp+dz \quad \text{where } 0<=p<=1 \end{aligned}\right\} \qquad (1)$$

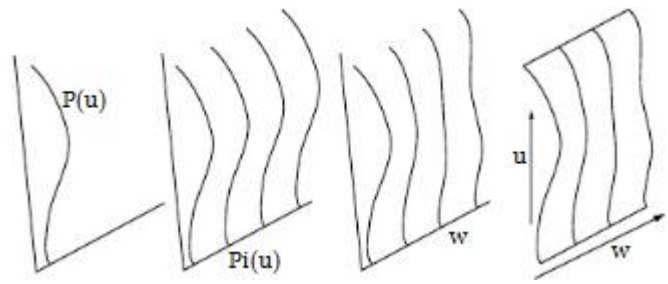
The values of all the four coefficients are determined for every n curve section within n+ 1 point. The numerical value for each of the coefficients is obtained by setting boundary conditions for all the joints between curve sections. Common methods that are implemented to set the boundary conditions are discussed in section 6.12 for cubic interpolation splines.

8.1.2 Surfaces

Surface in simple words is known as a family of curves. A solid surface is obtained by merging infinite number of curves without allowing any gap between any two curves.

The figure 8.2 shows a single curve being duplicated and placed as a set to form a surface.

Figure 8.2 : A Surface as a Family of Curves



If the curve is denoted by P (u), its duplicates can be denoted by P_i (u), i being the integer index. The integer index i can be replaced by 'w' for each curve. Here, w is the real index. The solid surface will then be denoted by P_w (u) here, u moves with the curve in small steps.

Polygon Surfaces

Also referred to as "standard graphics object", the set of polygons (where, a polygon is a plane figure bounded by a closed path) surfaces are the most commonly used 3-D graphics object representation. The graphics system stores all the descriptions of object as set of polygon surfaces. It helps in simplifying and speeding up the surface rendering and display of objects. A polygon representation defines the surface features of the polyhedron object in a very accurate manner. Other than polygon surface many packages also allow objects to be described with spline surfaces that are converted into polygon representations.

Polygon Mesh

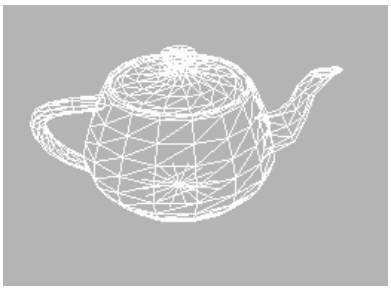
The polygon mesh representation is common in designing and solid modeling applications. The mesh outline for the surface can be displayed quickly to give an indication of the structure of the surface. To reduce the display of polygon edge boundaries on the surfaces, the realistic renderings are created. It is done by interpolating shading patterns across the surfaces.

Notes

Mesh outline – It is a data structure used as a modeling method in the field of computer graphics. An example of mesh outline image is as shown in figure 8.3.

The figure 8.3 shows mesh outline example.

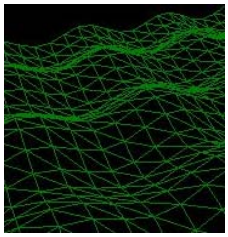
Figure 8.3 : Mesh Outline Example



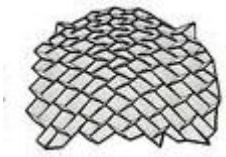
Source: http://www.siggraph.org/education/materials/HyperGraph/modeling/polymesh/polymesh.htm

Example: Some of the common polygon meshes are triangle strip and quadrilateral mesh as shown in figure 8.4. The renderers that implement hardware have the capacity to produce up to 1,000,000 or even more shaded triangles per second. They also include application of lighting effects and surface texture.

Figure 8.4 : Triangle and Quadrilateral Examples



Triangle strip



Quadrilateral mesh

Source: http://en.wikipedia.org/wiki/File:Spoon_fi.jpg

Polygon Tables

The polygon surfaces are usually specified with a set of vertex coordinates and associated attribute parameters which are listed below:

1. ***Geometric Data Table:*** It includes vertices, edges, and polygon surfaces.
2. ***Attribute Table*:** It includes degree of transparency and surface reflectivity, etc.

The best way to organize the storage of geometric data is to create three lists:

1 ***A Vertex Table*:** Here the values of each coordinate of the vertex are stored.

2. ***An Edge Table*:** It includes the pointers to indicate each of the vertices edge into the vertex table.
3. ***A Polygon Table*:** It contains pointers to the edge table to indicate edges for the polygons.

The separate listing of vertices, edges and polygons prove to be a very convenient method to refer to individual polygonal components. The object can be displayed efficiently by using the information extracted from the edge table and drawing the component lines. In some cases only tables are used, i.e. the vertex and polygon table.

The table 8.2 depicts the geometric data representation for the two adjacent polygon surfaces that are shown in the figure 8.5.

Figure 8.5 : Adjacent Polygon Surface

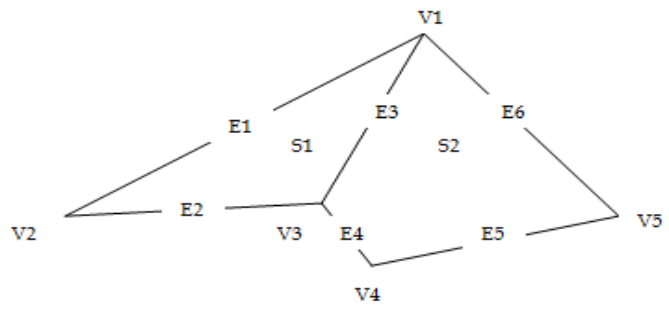


Table 8.2 : Vertex, Edge and Polygon Surface Tables

Vertex Table
V1: x1,y1,z1
V2: x2,y2,z2
V3: x3,y3,z3
V4: x4,y4,z4
V5: x5,y5,z5

Edge table
E1:V1,V2
E2:V2,V3
E3:V1,V3
E4:V3,V4
E5:V4,V5
E6:V1.,V5

Polygon surface table
S1: E1,E2,E3
S2:E3,E4,E5,E6

The data tables that are written for the polygons can be tested for their consistency by undergoing the following tests. The tests check whether,

1. Each vertex is listed as an endpoint for at least 2 edges.
2. Each edge is part of at least one polygon.
3. Each polygon is closed.
4. Each polygon has a minimum of one common edge.

5. Each edge table contains indicators to their respective polygons and every edge pointed by the polygon indicator has an indicator which is indicating back to its polygon or not.

Plane Equation and Visible Points

The input data that defines the representation of the 3-D object must be processed by the computer to display the object. The steps to be included in the processing are the following:

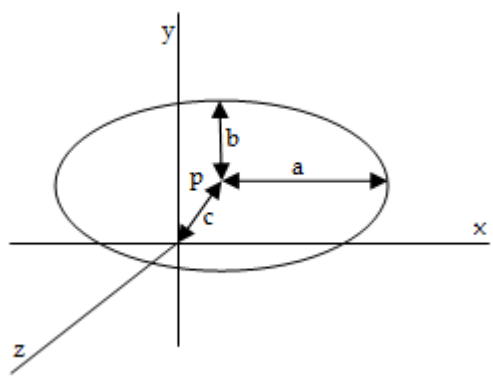
1. Transforming the model and descriptions of coordinates to device coordinates.
2. Identifying the surfaces that are visible.
3. Applying the procedures of surface rendering.

In some cases the information regarding the spatial orientation of individual surface components are required. The vertex coordinate values and the equations that describe a polygon plane provide the geometrical information.

Quadratic Surface

The quadratic polynomials define quadratic surfaces in 3-D. A quadratic object is created and rendering state is set with one or more state setting functions. The object is specified when drawing one of its surfaces, and its state determines the way the object is rendered. These surfaces include spheres, ellipsoids, etc.

Example:



The above ellipsoid can be mathematically written as,

$f(p)=(x-x_c)2/a^2+(y-y_c)^2/b^2+(z-z_c)^2/c^2-1=0$

Bezier Surface

The Bezier surfaces were first described by Pierre Bezier. The Bezier surface is defined by a set of control points. Here, the surface does not pass through the control points that are positioned in between the surface. They can be of any degree but the popularly used are the cubic Bezier surfaces. They are the most common parametric surfaces implemented for modeling.

Bezier equation

For a 2-D Bezier surface,

$$\mathbf{p}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} B_i^n(u)\ B_j^m(v)\ \mathbf{k}_{i,j}$$

Here, P= function of parametric coordinates u, v

Bezier surface is of the order (n, m)

Control points are (n+1) (m+1) =ki,j

$B_i^n(u) = \binom{n}{i} u^i (1-u)^{n-i}$ B is Bernstein polynomial and

$\binom{n}{i} = \frac{n!}{i!(n-i)!}$, this is a binomial coefficient.

For a 3-D Bezier surface,

$$s(\alpha, \beta) = \sum_{k=0}^{3} \sum_{j=0}^{3} B_j^3(\alpha) B_k^3(\beta) \bar{p}_{j,k} = \sum_k B_k^3(\beta) \bar{p}_k(\alpha)$$

Here, each $\bar{p}_k(\alpha)$ is a Bezier curve.

$$\bar{p}_k(\alpha) = \sum_j B_j^3(\alpha) \bar{p}_{j,k}$$

Source: http://en.wikipedia.org/wiki/B%C3%A9zier_curve

Caution The B here stands for basis. Not to be confused with beta or Bezier.

B-spline Surface

The B–spline surface is obtained by the Cartesian product resulting from the extension of B-spline curve. B-spline surface is sectioned by dividing the polygon grid lines in one or both parametric direction. Its flexibility is improved by raising the order of the basis function resulting in defining the polygon/grid lines. It allows the generation of curves for any degree of continuity. They are the preferred way to describe smooth curves.

Properties of B-spline

1. The highest order in each parametric direction depends on the number of defining polygon vertices in that direction.
2. The surface being continued in each parametric direction is k-2, l-2.
3. The surface does not change based on affine transformation.
4. The polygon net vertex influences the parametric direction in the range ±k/2, ±1/2.
5. The B spline surface reduces to a Bezier surface if the number of polygon net vertices is equal to the order of basis in that direction and if there exists no interior knot values.

8.1.3 Curves

A continuous map from a one-dimensional space to an n dimensional space is known as a curve in mathematical terms. It is made up of a number of continuous points. The main property of a curve is that every point in a curve has a neighbor. The infinite and closed curves do not have neighbors. (End points).

Curves can be described in three different ways.

Implicit –these representations define the set of points on a curve by providing them with a procedure to check if a point is positioned on the curve or not. It is defined by an implicit function of the form,

F(x, y) =0

It is a scalar function.

Explicit or Parametric Curve

It provides a mapping from a single parameter to a set of points on the curve. The parameter provides an index to curve points.

Generative or Procedural Curve

The procedural curve descriptions are very different from the first two types of curves.

Example: Subdivision schemes and fractals.

Hermit Curve

Hermit curves are described by the start points, end points and tangents at those points. To obtain a point on the hermit curve, each of the control points are multiplied by some function and added. The functions are called basis functions. The basis functions when applied to all the components of the hermit curve act in the similar way as Bezier and B spline curves.

Hermit curve equation

The equation for a hermit curve that interpolates on (x_k, x_{k+1}) implements the following equation:

$P(x) = h_{00}(t)p_k + h_{10}(t)(x_{k+1} - x_k)m_k + h_{01}(t)p_{k+1} + h_{11}(t)(x_{k+1} - x_k)m_{k+1}$

Where, $t=(x-x_k)/(x_{k+1}-x_k)$ and h is the basis functions.

These curves do not change due to transformations of rotation, translation and rotation. If these transformations are applied to the endpoints and tangent vectors, they will automatically get applied to the transformations of the entire curve.

Bezier Curve

In the case of Bezier curves the endpoints are defined by two control points. The tangents at the end points are controlled in a geometric manner.

Bezier curve equation:

A Bezier curve of degree **n** can be defined as

$$\begin{aligned}\mathbf{B}(t) &= \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-i} t^i \mathbf{P}_i \\ &= (1-t)^n \mathbf{P}_0 + \binom{n}{1}(1-t)^{n-1} t \mathbf{P}_1 + \cdots \\ &\quad \cdots + \binom{n}{n-1}(1-t) t^{n-1} \mathbf{P}_{n-1} + t^n \mathbf{P}_n, \quad t \in [0,1],\end{aligned}$$

Here, $\binom{n}{1}$ is a binomial coefficient.

For linear interpolation,

$\mathrm{B}(t) = \mathrm{P}_0 + t(\mathrm{P}_1 - \mathrm{P}_0) = (1-t)\mathrm{P}_0 + t\mathrm{P}_1$, $t \in [0,1]$

For quadratic interpolation,

$\mathbf{B}(t) = (1-t)^2\mathbf{P}_0 + 2(1-t)t\mathbf{P}_1 + t^2\mathbf{P}_2$, $t \in [0,1]$.

Properties of Bezier curves

1. The initial and final control points are interpolated.
2. The tangent to the curve at the first control point is along the line joining the first and second control points.
3. The tangent at the last control point is along the line joining the second last and last control points.
4. The curve lies entirely within the convex hull of its control points.

B-Spline Curve

A B-spline curve P (t) can be defined as

$$\mathbf{P}(t) = \sum_{i=0}^{n} \mathbf{P}_i N_{i,k}(t)$$

Here,

{Pi: i=0, 1, n} represent the control points.

K represents the order of the polynomial segments of the B-spline curve. The order k means that the curve is composed of piecewise polynomial segments of degree k-1.

$N_{i,k}$ (t) - this term is the normalized B-spline blending functions. They can be described by order k and by the sequence of real numbers that do not reduce.

Periodic B-spline

The B-spline that closes on itself is called the periodic B-spline. It requires that the first 'n' control points are identical to the last 'n', and the initial 'n' parameters in the knot vector are identical to the finishing intervals.

8.1.4 Add-ons: Polygon Rendering

Polygon rendering is used to create images that are 3-D in nature. In normal practice, objects that arise when their surface is modeled are triangular polygons and they are next rendered in a wire frame model. Rendering is the process that used to model the objects by specifying their surfaces using polygons.

It consists of three types of shading, namely:

1. Constant Shading
2. Gouraud Shading
3. Phong Shading

Now let us discuss each of them.

Constant Shading

This is the simplest shading model. It is also known as 'faceted shading' or 'flat shading'. A polygon here is shaded by applying an illumination model and determining the intensity value so that it can be used to shade an entire polygon model. The intensity value is held across polygon to create the polygon's shade. This approach is considered valid if the following assumptions are true:

1. Light source is placed at infinity, with the light source vector kept constant across polygon face.
2. Viewer at infinity, with normal vector kept constant across polygon face.
3. The actual surface being modeled is represented by the polygon and not its approximation.

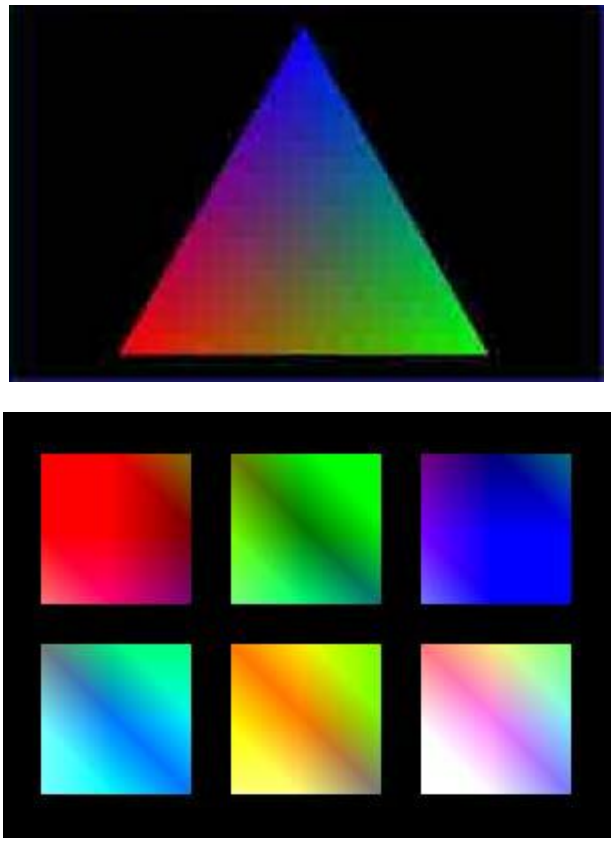
In case, any of the first two assumptions are not correct, some method must be used to determine the value of L and V.

Example: For the center of a polygon or the polygon's first vertex, the values may be calculated. Here, the variations in shade are not produced for this polygon that should actually occur in this case.

Gouraud Shading

Gouraud shading also known as intensity interpolation shading, does not support intensity discontinuity. The figure 8.6 is a triangle with all its vertices having different shades. The second figure displays all the shades that are displayed in vertices and the edges of the shaded triangle.

Figure 8.6 : Gouraud Shading



Source: http://www.dcemu.co.uk/vbulletin/threads/172575-Gouraud-Shading-Test-(Sega-Saturn-Techdemo)

Gouraud shading is based on the concept of another shading method known as interpolation shading. This method interpolates polygon vertex illumination values that take the approximate values of polygons. It requires that the normal is known for each vertex of the polygonal mesh.

Notes

Interpolation Shading

Here the shading information is interpolated in a linear manner across a triangle from the values that are determined for its vertices. This method is conveniently used in scan line algorithms.

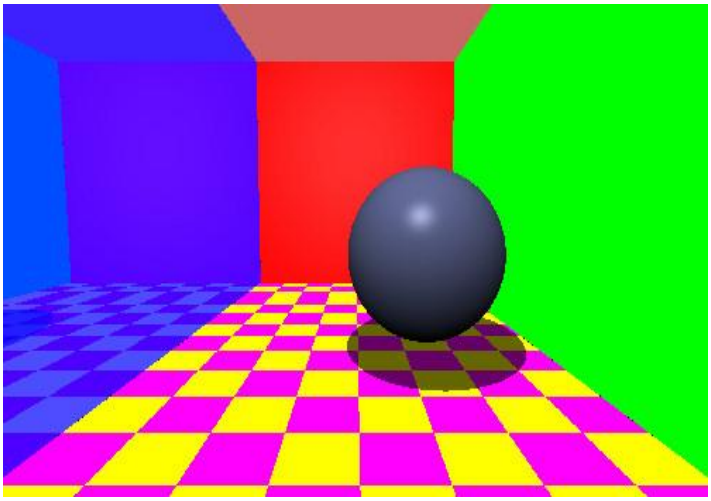
Phong Shading

The phong shading model was developed in 1973 by Bui-Tong Phong. The phong model consists of three components namely,

1. ***Diffuse Relection*:** Here the intensity of light is independent of the direction to the viewer.
2. ***Specular:*** There is simulation of relective surfaces and specular highlights based on the viewer direction.
3. ***Ambient:*** The light that is diffusely reflected from the surfaces results in an approximation of an illumination.

The figure 8.7 displays the diffuse component. The light is falling from one-direction as the light is in the normal to the surface, the shadow is bright and scattered equally.

Figure 8.7: Phong Shading

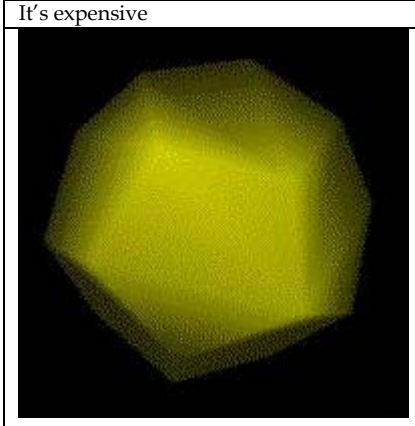
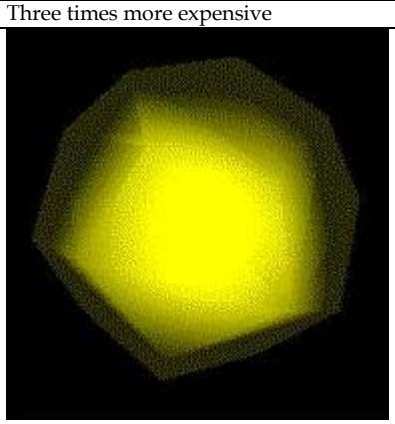


Source: http://people.rit.edu/pds2352/CGII/raytracer.htm

Phong model is more accurate when compared to other models. With incremental calculations the components of the normal vector can be incremented from scan line to scan line, the interpolation along edges can be done.

Table 8.3 illustrates the difference between gouraud and phong model.

Table 8.3 : Gouraud vs. Phong

Gouraud Shading	Phong Shading
Intensities are interpolated	Normal are interpolated
It's expensive	Three times more expensive
	
Restricted to Diffuse Component	**Specular Reflection**

Source: http://www.vrarchitect.net/anu/cg/Illumination/shadingSummary.en.html

Task

Browse the net and study in detail about the different shading models.

8.2 3-D Transformation

A transformation is the process of mapping points to other positions. The manipulation viewing and creation of 3-D images require the use of 3-D geometric and coordinate transformations. The 3-D transformations are implemented from 2-D methods. They are also formed by composing the basic transformations of translation, scaling and rotation concepts as discussed in the previous section. These transformations are represented in matrix transformation with homogenous coordinates. They are formed by combining matrices for individual transformations in sequential order.

Transformations are used in mapping spaces along the graphics pipeline. In the extensions from 2-D to 3-D, the following changes occur:

1. An additional vector is added, totally 2 vectors
2. Extra column and row gets added to the matrices
3. Singular Value Decomposition (SVD) remains working in the same way

Some of the properties of 3-D transformations:

1. Lines are preserved
2. Parallelism is preserved
3. Proportional distances are preserved

The various 3-D transformations that are discussed next are the following:

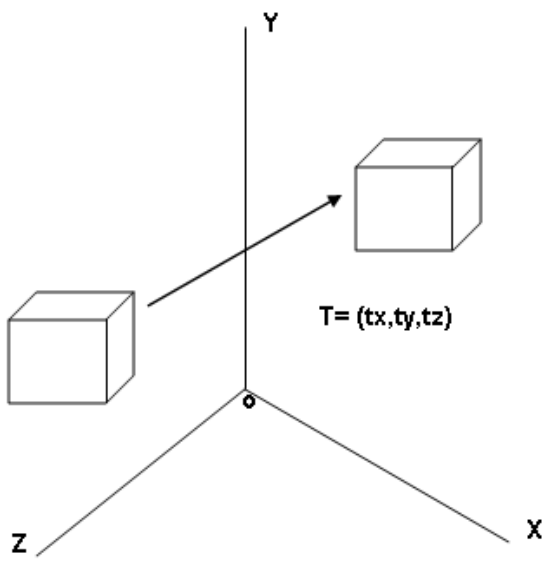
1. Translation
2. Scaling
3. Rotation

These are some of the basic 3-D transformations.

8.2.1 Translation

Translation is the method of changing the location of an object that is placed on a straight line from one coordinate position to another. Translation distances, D_x, D_y and D_Z are added in case of moving a three dimensional point from (x, y, z) to any point (x′, y′, z′). The same is depicted in figure 8.8.

Figure 8.8 : 3-D Translation



In order to achieve a translation transformation within a 3-D space we make use of the transformation matrix:

$$T(x, y, z) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ D_x & D_y & D_z & 1 \end{bmatrix}.$$

Notice the use of the symbol T followed by (x, y, z) to denote a matrix responsible for translation (in the x, y and z directions). This provides a concise way of referring to a matrix of this form.

Example: To move a point P that is located at (3, 4, 5) to a new location with distance (2, 4, 6) units.

Given, P= [3, 4, 5, 1]

Dx=2, Dy=4, Dy=6

We apply the translation transformation and obtain the result as

$$\begin{bmatrix} 3 & 4 & 5 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 2 & 4 & 6 & 1 \end{bmatrix} = \begin{bmatrix} 3+2 & 4+4 & 5+6 & 1 \end{bmatrix} = \begin{bmatrix} 5 & 8 & 11 & 1 \end{bmatrix}$$

Hence the result is the point P being shifted to a new position P'.

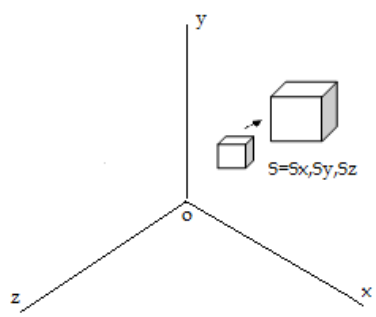
8.2.2 Scaling

The scaling in the x, y and z directions can be obtained by using the transformation matrix:

$$S[x,y,z] = \begin{pmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

In the x, y and z directions the scaling factors S_x, S_y and S_z are applied as shown in the figure 8.9. The letter S denotes the basic scaling matrix.

Figure 8.9 : 3-D Scaling



The following sequences of transformations occur in the scaling process for a fixed point:

1. The fixed point is translated to origin
2. The object is scaled
3. The fixed point is translated to its original position

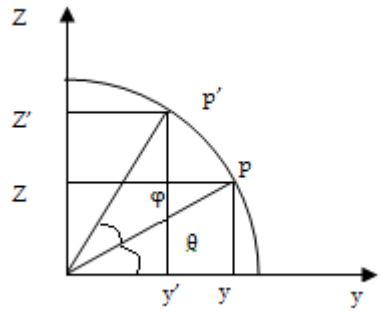
8.2.3 Rotation

The rotation transformations in 3-D plane can be designated around any line in space. This is not true in the case of 2-D rotations. So, an axis of rotation is specified for a 3-D rotation. The object rotates along the angle of rotation.

Assume that a point P is an arbitrary point with coordinates(x, y, z). It is rotated through an angle φ about the x-axis which does not alter. Now, we redraw the figure as a view from the y-z plane.

The figure 8.10 depicts 3-D rotation.

Figure 8.10 : 3-D Rotation



In the Figure 8.10 the point P is rotated through an angle θ about the x-axis to position P (the positive x-axis is assumed to be out of the page).

The coordinates of P are given by (x, y, z) and those of P′ are (x, y, z) as shown in figure 8.10.

Notes

The angle of rotation is assumed to be anti-clockwise (when considered from a perspective in which the rotation is viewed down the x-axis (from the positive end) looking towards the origin).

Using the standard trigonometric equations, we can express the transformed coordinates in terms of θ and φ as

$x' = p'\cos(\varphi+\theta) = p'\cos\varphi\cos\theta - p'\sin\varphi\sin\theta$…… (1)

$y' = p'\sin(\varphi+\theta) = p'\cos\varphi\sin\theta + p'\sin\varphi\cos\theta$…….. (2)

In the polar form the original coordinates are given in the form:

$$\left.\begin{aligned} x &= p'\cos\varphi \\ y &= p'\sin\varphi \end{aligned}\right\} \quad \text{…… (3)}$$

These equations are substituted in (1) and (2), we get the transformation equations for rotating a point (x, y) through an angle θ about the origin as:

$x' = x\cos\theta - y\sin\theta \quad y' = x\sin\theta + y\cos\theta$ ……. (4)

The equations given above can be written as in matrix form as:

P′= P.R

$$[x'\ y'] = [x\ y]\begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \text{…….. (5)}$$

where, R is a rotation matrix and is given as

$$R = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \text{…… (6)}$$

The matrices that are used to achieve rotation can be shown as,

$$R_x(\varphi) = \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\varphi & 0 \\ 0 & -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

$$Ry(\varphi) = \begin{matrix} \cos\varphi & 0 & -\sin\varphi & 0 \\ 0 & 1 & 0 & 0 \\ \sin\varphi & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

$$R_z(\varphi) = \begin{matrix} \cos\varphi & \sin\varphi & 0 & 0 \\ -\sin\varphi & \cos\varphi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

Here, φ is the angle of rotation. Above given are the matrices for achieving rotation about x, y or z coordinate axes.

Notes

Positive values for the rotation angle define counter clockwise equations about the rotation point and negative values rotate objects in the clockwise direction.

If the values of θ are negative, then for clockwise rotation the matrix becomes

$$R = \begin{pmatrix} \cos(-\theta) & \sin(-\theta) \\ -\sin(-\theta) & \cos(-\theta) \end{pmatrix}$$

Here, cos (-θ) =cos θ and sin (-θ) =-sin θ, Therefore,

$$R = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

8.2.4 3-D Clipping

Clipping is the process of removing objects that are not visible from the scene. The clipping process helps in reducing the computational effort. It can be obtained by following two steps:

1. Deleting objects that cannot be viewed. e.g., objects behind camera, outside the view field or that are too far from sight.
2. Also remove the object that intersects with the clipping plane.

Text clipping

Text can be represented through clipping. With respect to text, clipping can be divided into area and line clipping. Area clipping comes into picture when bitmaps are used to represent characters. When characters are represented by lines, line clipping is implemented. But in general cases, characters are a combination of lines and curves. Hence, they are called vectors.

Clipping of a text can be categorized in the following ways:

1. Precise clipping
2. Clipping to a character
3. Clipping a text by words

Each text is seen as an inseparable object in the case of clipping. Every character consists of a rectangle that is called as "rectangle envelope". The center of the envelope is tested in the display box while clipping two characters to each other.

8.3 3-D Viewing

The 3-D viewing process is much more complex than the 2-D viewing operations. Here projections are implemented to convert 3-D objects into 2-D projection planes. Here, the view volume is specified in the world coordinates using the modeling transformation. These transformations are next used in the conversion process.

In 3-D viewing, a viewing volume specifies a viewing volume with a projection method in the world coordinates, and a viewport on the display.

1. Viewing volume states what to appear
2. View port specifies where to display

For viewing an object, the camera and perspective projection is required. The camera properties are the following:

1. Eye at a point in space
2. View volume whose apex is at the eye
3. Field of view θ
4. Near and far clipping images
5. View plane and its aspect ratio

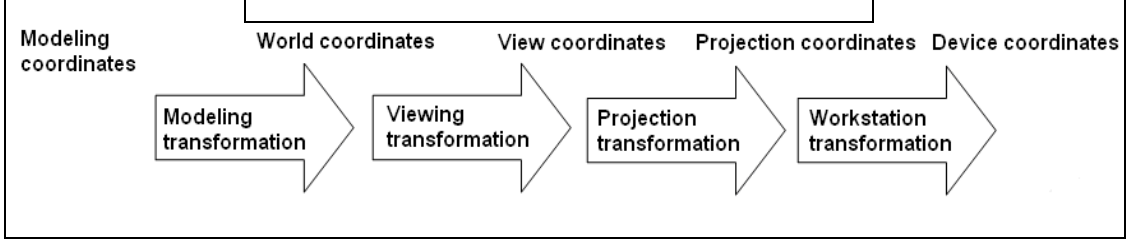
8.3.1 The Viewing Pipeline

The 3-D viewing volume is processed for projection methods. Objects are clipped against the 3-D viewing volume. The contents are then transformed into the viewport for display. The 3-D coordinates are preserved for lighting and hidden-surface removal.

Graphic Pipeline in 3-D Viewing

In the case of 3-D viewing, a view volume is described in terms of world coordinates. The modeling transformation is used for the same. The positions of world coordinates are transformed to viewing coordinates with the implementation of viewing transformation. The description in viewing coordinates is converted to 2-D projection coordinates using the projection transformation. In the final stage, the projection coordinates are transformed into device coordinates through the workstation transformation. The figure 8.11 depicts the pipeline.

Figure 8.11 : 3-D Viewing Pipeline



The stages in the graphic pipeline are responsible for information processing that are provided at initial stages as just properties at the vertices or control points. They specify what has to be rendered. The

basic objects related to 3-D graphics are lines and triangles. They include texture, reflectivity, RGB values, etc.

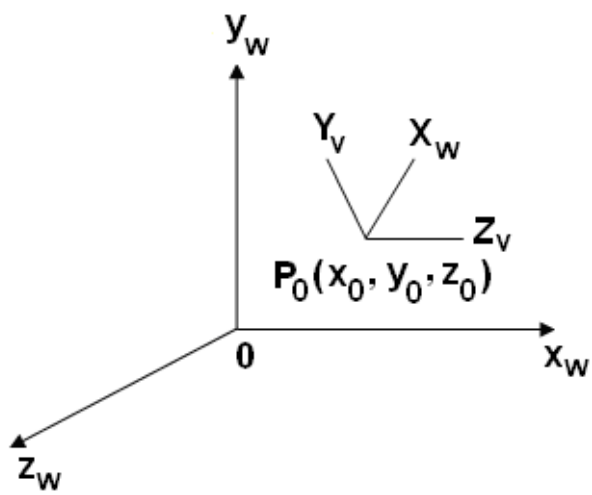
8.3.2 Camera Coordinates

A view plane is a film plane in a camera that is positioned for a particular shot of the scene. The world position coordinates are converted into viewing coordinates and projected into the view plane. The view plane can be established by viewing coordinate system or View Reference Coordinates (VRC).

Specifying an Arbitrary 3-D View

We know that an object can be viewed from any direction. It is also necessary to define a view plane based on a specified point of view. The figure 8.12 displays the view plane and the view coordinate system.

Figure 8.12 : Viewing Coordinate System



Here the first view parameter to be considered is the V**iew Reference Point** [VRP]. It is the center of view reference system. It is normally close to or on the surface of some object in a scene.

The second viewing parameter is the **View-Plane Normal vector** [VPN]. The normal vector is the direction that is perpendicular to the view plane.

The **view distance** is another parameter. The normal vector is a directed line segment from view plane to view reference point. The length of the directed line segment is called the view distance. It is illustrated in the figure 8.14. These parameters allow the use to select the necessary object view.

Transforming World Coordinates to Viewing Coordinates

The position of objects can be described in different ways namely, the camera coordinates and the world coordinates. The world coordinates can be converted to viewing coordinates by the following procedure:

1. To translate view reference point to origin of world coordinate system.
2. To align the xv, yv, zv axes with the world coordinates xw, yw and zw axes, apply the rotations.

For the conversion the following matrix transformation is used:

$$T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x_p & -y_p & -z_p & 1 \end{pmatrix}$$

P' =P.T

To align the three axes, the three coordinate-axis rotations are required. It depends on the direction that is chosen for N. Here, N is the normal vector.

8.3.3 Viewing Volume

An orthographic projection is a parallel projection if the projection lines are all normal to the view plane. The orthographic viewing volume constitutes the six plane equations, x=left, x=right, y=top, y=bottom, and z=-near or far.

The camera is always set in such a way that the eye is at the origin and the VPN is aligned with the z-axis. The look point defined by the programmer acts as a point of importance in the scene, and when combined with eye, the look and eye together defines the View Plane Normal (VPN) as eye-look.

Setting the View Volume

To view a scene the camera is moved and aimed in a particular direction. The viewing is done by performing the rotation and translation processes, it is part of model view matrix that stores the exact vector data that defines the three directions.

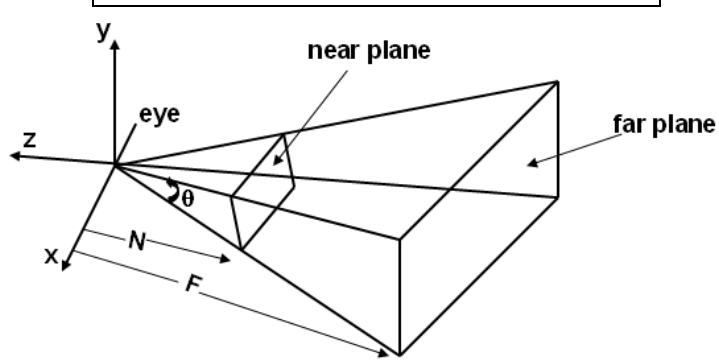
By setting the view volume, the camera is moved so that its eye is positioned at point eye. The position of look is towards the point of importance. The vector up suggests the upward direction, which is in normal case set to (0, 1, 0).

Hence, the view volume defines the volume in 3-D. When the object is projected, the view volume assures that the projection fits the viewport. The figure 8.13 clearly shows the two planes. The view volume can be divided into two parts namely,

View Plane Rectangle: It defines the boundaries of x and y plane that will be mapped into the viewport. Rendering is possible only to the objects that lie inside the rectangle view plane.

Near Far Clipping: The bounds for object are defined only for the z direction in the near far clipping planes. Rendering here is possible only for the objects that lie nearer to the eye than the near clipping plane and also for the objects that lie farther from the eye and far from the clipping plane.

Figure 8.13 : View Volume



8.4 Summary

- The three-dimensional representation of data is stored in the computer. It is used for calculation and rendering purposes.
- The 3-D graphics and 2-D graphics rely on the same algorithms.
- A spline is mathematically described with a piecewise cubic polynomial function.
- A Bezier curve is one of the approaches to construct a curve. It is determined by defining a polygon.
- The creation of 3-D graphics can be divided into content creation, scene layout and rendering.
- Spline is defined as a parametric curve that can be well-defined by control points. The control points were adjusted by the user to regulate the shape of curve.
- A solid surface is obtained by merging infinite number of curves without allowing any gap between any two curves.
- Polygon surface is also referred to as "standard graphics object." A polygon surface is specified with a set of vertex coordinates and associated attribute parameters.
- The polygon mesh representation is common in design and solid modeling applications.
- The Bezier surface is defined by a set of control points. A B–spline surface is obtained by the Cartesian product resulting from the extension of B-spline curve.
- Images that are 3-D in nature require the implementation of polygon rendering. It consists of three types of shading namely, flat shading, gouraud shading and interpolation shading.
- Transformations are used in mapping spaces along the graphics pipeline. The three major types of transformations are translation, scaling and rotation.
- The other transformations are shear, reflection and composite.
- Clipping is the process of removing objects that are not visible from the scene.
- In 3-D viewing, a viewing volume is specified with projection methods in the world coordinates and a view port on the display.

8.5 Keywords

Application Programming Interface (API): It is a particular set of rules and specifications that a software program can follow to access and make use of the services and resources provided by another particular software program that implements that API.

Cartesian Product: It is the product two number sets. Example, product of two points coordinates.

Modeling: It is the mathematical representation of any 3-D image.

Open GL: It is a cross platform API used to write applications that develop graphic images.

Perspective Projection: It is an approximate representation of an object seen through naked eye on a flat surface.

Raster: It represents grid of pixels on a display medium or screen.

Rendering: It refers to the creation of an image from a model with the help of computer programs.

Texturing: It is the addition of extra details or color to a 3-D graphic.

Wire Frame Model: It is the visual representation of 3-D model.

World Coordinates: They are real coordinates of an object.

8.6 Self Assessment

1. State whether the following statements are true or false:
   (a) 3-D is represented with the aid of digital computers and 3-D software.
   (b) Scene layout is the arrangement of virtual objects, cameras, lights and other units.
   (c) A set of control points define the quadratic surface.
   (d) The vector is a directed line segment from view plane to view normal point.
   (e) An orthographic projection is a parallel projection if the projection lines are all normal to the projection plane.
   (f) The perspective transformation changes graphics pipeline into the canonical cube.
2. Fill in the blanks:
   (a) Splines are represented mathematically with a ……. polynomial function.
   (b) A …… is a film plane in a camera that is positioned for a particular shot of the scene.
   (c) Clipping is the process of removing objects that are not……. from the scene.
   (d) A composite transformation matrix is the product of ……. matrices.
   (e) ……. are implemented to convert 3-D objects into 2-D projection planes.
3. Select the suitable choice for every question:
   (a) Surface in simple words is a family of:
      (i) Curves
      (ii) Lines
      (iii) Squares
      (iv) Spheres
   (b) The Bezier surface does not pass through the control points that are positioned:
      (i) Beside the surface
      (ii) Behind the surface
      (iii) Front the surface
      (iv) In between surface
   (c) Which process helps in reducing the computational effort?
      (i) Clipping
      (ii) Transforming
      (iii) Rendering
      (iv) Rotating
   (d) Which of the following representations are common in design and solid modeling applications?
      (i) Polygon tables
      (ii) Polygon rendering
      (iii) Polygon mesh
      (iv) Polygon mapping

8.7 Review Questions

1. "Spline is defined as a parametric curve that can be well-defined by control points." Do you agree? Justify.
2. "The set of polygon surfaces are the most commonly used 3-D graphics object representation." Discuss.
3. "Its flexibility is improved by raising the order of the basis function resulting in defining the polygon/grid lines." Describe.
4. "A continuous map from a one-dimensional space to an n dimensional space is known as a curve in mathematical terms." Elaborate.
5. "Polygon rendering is used to create images that are 3-D in nature." Discuss.
6. "Gourand and Phong shading are quite different." Explain.
7. Transformations are used in mapping spaces along the graphics pipeline. Explain the different transformations.
8. "Rotation for three-dimensional objects is possible." Justify.
9. "Shears distort the shape of an object." Explain.
10. "The world position coordinates are converted into viewing coordinates and projected into the view plane." Describe.
11. "An orthographic projection is a parallel projection if the projection lines are all normal to the view plane." Analyze.
12. "The view volume must be set to a particular direction to view an object in the correct manner." Explain.

Answers – Self Assessment

1. (a) True (b) True (c) False (d) false (e) False (f) False
2. (a) Cubic (b) View plane (c) Visible (d) Individual (e) Projections
3. (a) Curves (b) In between the space (c) Clipping (d) Polygon mesh

8.8 Further Readings

Books

Godse, A.P.(2005).Computer Graphics, 1st ed. India: Technical Publications

Xiang, Z., & Plastock, R.A.(2010). Computer Graphics, 2nd ed. USA: McGraw-Hill companies

Online link

http://www.slidefinder.net/c/computer_graphics_clipping/12155743

http://www.scribd.com/doc/7075120/Curves-and-Surfaces-for-Computer-Graphics

http://faculty.umf.maine.edu/daniel.jackson1/public.www/Computer%20Graphics.htm

Unit 9: Clipping I

CONTENTS

Objectives

Introduction

9.1 Overview of Clipping

9.2 Point Clipping

9.3 Line Clipping

9.3.1 Cohen Sutherland Algorithm

9.3.2 Midpoint Subdivision Algorithm

9.3.3 Liang-Barsky Algorithm

9.4 Summary

9.5 Keywords

9.6 Self Assessment

9.7 Review Questions

9.8 Further Readings

Objectives

After studying this unit, you will be able to:

- Explain point clipping
- Explain line clipping
- Explain midpoint subdivision

Introduction

A scene consists of collection of objects that are specified in world coordinates(x and y axis). When a scene is displayed, you can view only that objects which are inside the viewing window (clip window) and clip everything outside the v window. The clipping algorithms are developed to identify which points and lines should be clipped and which ones should be retained.

The clipping algorithm is the procedure that helps to identify those portions of a picture that are either inside or outside the specified window frame at the time of clipping. The picture region against which the object is supposed to be clipped is called as a **clip window**.

Clipping algorithms are important in computer graphics. They are used to view transformations and erase picture sections in drawing and painting packages. They are also used in many window manager systems such as GNOME.

Notes

GNOME is a graphical user interface that includes a variety of features such as file archiving tool, file search tool and games.

The window manager systems provide a function to create and manipulate the display of multiple processes.

In this unit, you will learn the two most common types of clipping, namely point clipping and line clipping. You will learn various clipping algorithms that are developed for extracting a part of a defined scene for viewing, displaying multi-window environment, drawing, painting, and so on.

9.1 Overview of Clipping

Clipping, in general, refers to the removal of a part of a scene. Imagine a person seeing the world through the window of a house. The part of the scene that cannot be viewed from the window is called as cut off portion or left out portion. In computer graphics, the portion that is left outside the clip window is called the clipped part and the process used to display the image within the window is called clipping. In general, a clipping algorithm is used to remove the image components that lie outside the viewing window.

According to Peter Comninos, Director of Bournemouth's National Centre for Computer Animation, "Clipping is a process that subdivides each element of a picture to be displayed into its visible and invisible parts, thus allowing us to discard the invisible part of the picture."

Clipping algorithms can be applied in world co-ordinates such that only the picture of the object inside the window is mapped to the device co-ordinates. Alternatively, the entire world co-ordinate picture is first mapped to the device co-ordinate picture and then clipped against the viewport boundaries.

World co-ordinate clipping does not consider pictures outside the clipping window. Thus, it eliminates the processing required to transform those pictures to device space. On the other hand, viewport clipping reduces calculations by linking the viewing and geometric transformation matrices. This unit discusses three primitive types of clipping. They are:

1. Point clipping
2. Line Clipping
3. Polygon Clipping (Will be discussed in Unit-10)

Notes

World co-ordinate is the reference frame that is used to place the object in the appropriate position. This co-ordinate system is linear along both axes.

Did you know?

Applications such as Word processors and Spreadsheets clip keyboard input to prevent the input from appearing in the margins of a page. Computer-Aided Design (CAD) programs clip graphics output to prevent the output from overwriting the edges of a drawing.

9.2 Point Clipping

How does point clipping work? Point clipping removes points that are outside the clipping window. Assume that the clip window is rectangular. The point p(x, y) is saved for display, if the following inequalities are satisfied:

$xw_{min} \le x \le xw_{max}$

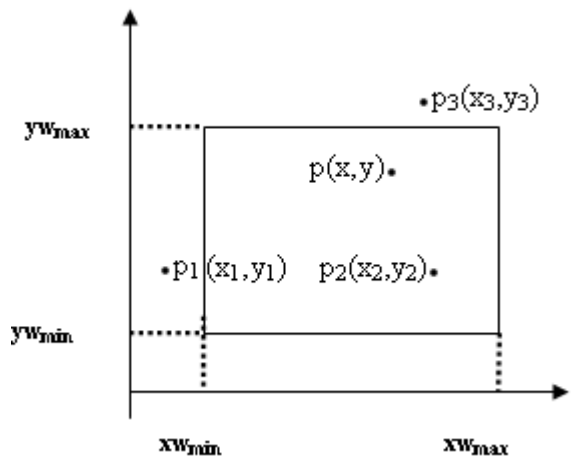
$yw_{min} \le y \le yw_{max}$

where, xw_{min} and xw_{max} are the edges of the clip window boundaries that are parallel to the y-axis and

yw_{min} and yw_{max} are the edges of the clip window boundaries that are parallel to the x-axis.

The figure 9.1 depicts point clipping.

Figure 9.1: Point Clipping



In figure 9.1, the edges of the clip window can either be world co-ordinate window boundaries or viewport boundaries. The points p(x, y) and $p_2(x_2, y_2)$ are not clipped as they satisfy the above inequalities. The points $p_1(x_1, y_1)$ and $p_3(x_3, y_3)$ that are outside the window are clipped and are not displayed.

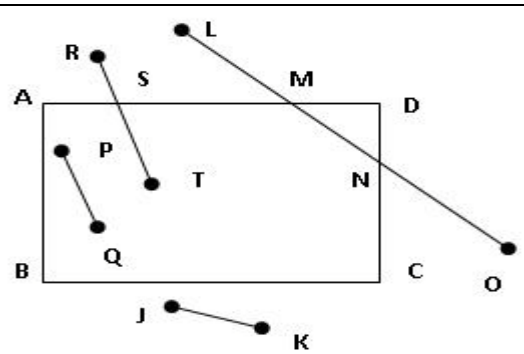
Point clipping is used less often compared to line clipping. Some applications prefer using point clipping.

Example: Scenes that have explosions use point clipping. This is because explosion scenes involve particles (points) that spread in various regions of the scene.

9.3 Line Clipping

Line clipping mainly determines whether points lie within or outside a clipping window and whether a line joining two points lies completely inside or outside the window. With reference to a window ABCD as indicated in figure 9.2, there are three possibilities for the line to be visible.

Figure 9.2: Line Clipping



The three possibilities shown in the figure 9.2 are:

(a) The line PQ lies within the window. Thus, the entire line is visible.

(b) The line JK does not cut the edge of the window and lies outside the window. Thus, the entire line will be invisible.

(c) The line RT has one end within the window and the other end outside the window. The line cuts the window edge at S. Thus, the portion ST will be visible and the other portion RS, which is outside the window, will be invisible. The line LO cuts two edges of the window. Thus, the portions LM and NO, which lie outside the window, will be invisible and the line segment MN within the window will be visible.

From figure 9.2, it is clear that the line clipping depends on the position of the endpoints with reference to the window edges. If the endpoints lie within the window, then the entire line is visible. If only one endpoint lies within the window, then the line cuts the window and the visible line segment has to be determined. If both the endpoints lie outside the window, then it has to be determined whether the line cuts the window edges. Some algorithms are used to determine the visibility of lines. They are:

(a) Cohen Sutherland Algorithm

(b) Midpoint Subdivision Algorithm

(c) Liang-Barsky Algorithm

Did you know? Evans & Sutherland (E&S) was established in 1968. The company created custom line drawing routines. The Line Drawing System (LDS-1) was the first hardware accelerated graphics system developed that introduced line clipping.

Now, let us discuss the algorithms.

9.3.1 Cohen Sutherland Algorithm

The Cohen Sutherland algorithm is the most commonly used algorithm for clipping lines against a window frame in 2-D space. It divides the 2-D space into nine regions. Each region is assigned a 4-bit code for identification. Cohen Sutherland algorithm does the line clipping in two phases. It first identifies the lines that intersect the clipping window and then performs the clipping process.

As discussed in figure 9.2, you know that the lines can be in any one of the following three categories:

1. ***Visible:*** The endpoints of a line lie within the window.

2. ***Invisible:*** The endpoints of a line lie outside the window.

Example: If a line with the endpoints (x_1, y_1) and (x_2, y_2) satisfies the below four inequalities, then the line lies outside the window.

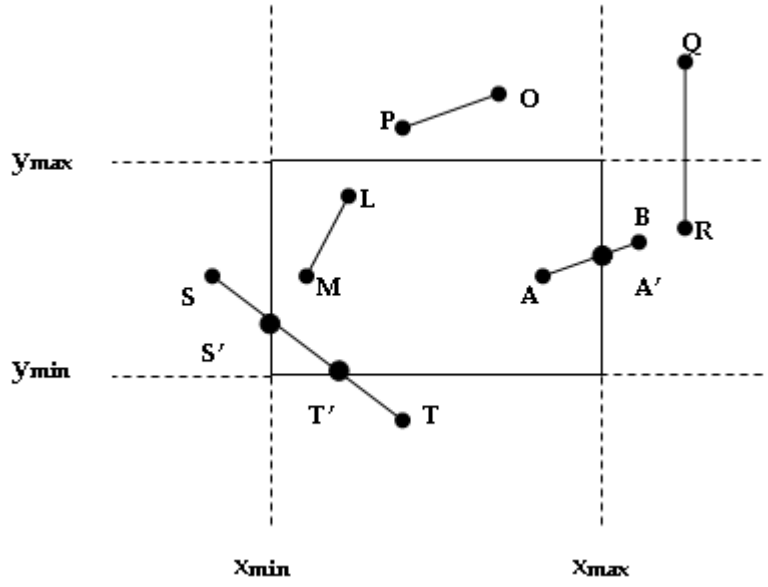
$(x_1, x_2) > x_{max}$ $(y_1, y_2) > y_{max}$

(a) $(x_1, x_2) < x_{min}$ $(y_1, y_2) < y_{min}$

3. ***Clipping Candidate:*** The line is neither in visible or invisible category. The line intersects the clipping window.

In figure 9.3, the above three categories are clearly differentiated.

Figure 9.3: Line Categories

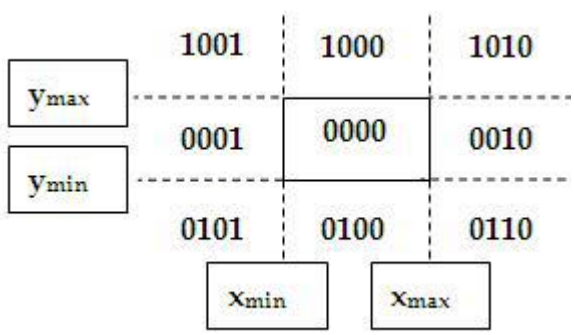


In figure 9.3, the line LM falls under category 1 (Visible). The lines OP and QR fall under category 2 (Invisible) and the lines ST and AB fall under category 3 (Clipping candidate).

The Cohen Sutherland algorithm provides an efficient way to determine the category of a line. The algorithm divides the procedure into two steps:

1. ***Step 1*:** A 4-bit binary code called region code is assigned to each endpoint of the line. The endpoints lie in any of the nine regions of the plane. Figure 9.4 shows the 4-bit region codes that are assigned to every endpoint of the line depending on its position in 2-D space.

Figure 9.4 : Region Codes



Each bit of the code is set either true (1) or false (0) according to the following scheme.

A region code is assigned to every point of a line depending on its position (x,y). Starting from the leftmost bit,

Bit 1: sign($y-y_{max}$) = Endpoint is above the window

Bit 2: sign($y_{min}-y$) = Endpoint is below the window

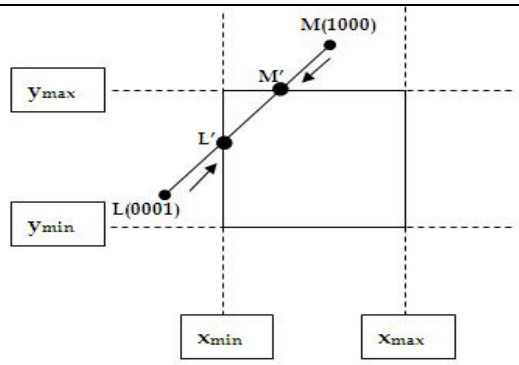
Bit 3: sign($x-x_{max}$) = Endpoint is to the right of the window

Bit 4: sign($x_{min}-x$) = Endpoint is to the left of the window

Here, sign(*a*) is set to 1 if *a* is positive otherwise *a* is set to 0. The point with code 0000 is always inside the window.

2. ***Step 2:*** If the endpoints of the line lie in the region with codes 0000 then the line is visible. The line is not visible if the bitwise logical AND of the codes is not 0000. Thus, this line is considered as clipping candidate.

Figure 9.5: Clipping Candidate



In case of a line in category 3 (Clipping candidate), you need to find the point that intersects with one of the boundaries of the clipping window. First, you need to select the endpoint of the line (x_1, y_1) that is outside the window whose region code is not 0000. Then, you need to select an extended boundary by observing those boundary lines that are candidates for intersection. Here, the selected endpoint is pushed across in order to change '1' in its code to '0'. This indicates that:

If Bit 1 is 1, then intersect with line $y = y_{max}$

If Bit 2 is 1, then intersect with line $y = y_{min}$

If Bit 3 is 1, then intersect with line $x = x_{max}$

If Bit 4 is 1, then intersect with line $x = x_{min}$

In figure 9.5, the line LM falls under the clipping candidate category. If endpoint L is selected, then the intersection point is computed by selecting the bottom boundary line $y = y_{min}$. If endpoint M is selected, then either the top boundary line $y = y_{max}$ or the right boundary line $x = x_{max}$ is selected for computing intersection. The co-ordinates of the intersecting points are:

$$\begin{cases} x_i = x_{min} \text{ or } x_{max} & \text{if the boundary line is vertical} \\ y_i = y_1 + m\,(x_i + x_1) & \end{cases}$$

or

$$\begin{cases} x_i = x_1 + (y_i - y_1)/m & \text{if the boundary line is horizontal} \\ y_i = y_{min} \text{ or } y_{max} & \end{cases}$$

where, m is the slope of the line. $\mathbf{m = (y_2 - y_1)/(x_2 - x_1)}$

Now, the endpoint (x_1, y_1) is replaced with the intersection point (x_i, y_i), thereby eliminating the line segment that lies outside the window boundary. The new endpoint is assigned a 4-bit region code and the clipped line is re-categorized. This process is performed until the clipped line falls under either category 1 (visible) or category 2 (invisible).

Task

Consider the point (50, 65) is at the top and to the right of the clip window. What is the 4-bit Sutherland code (region code) of this point? You can perform the bitwise logical OR of both top and right region code and obtain the result.

Caution

Only those lines that are partially inside and partially outside the clipping window are checked for intersection with the window boundaries. The clipping process starts by comparing an outside endpoint to a clipping boundary. This helps to determine which line segments need to be discarded. This process continues till the section of the line lies inside the clipping window.

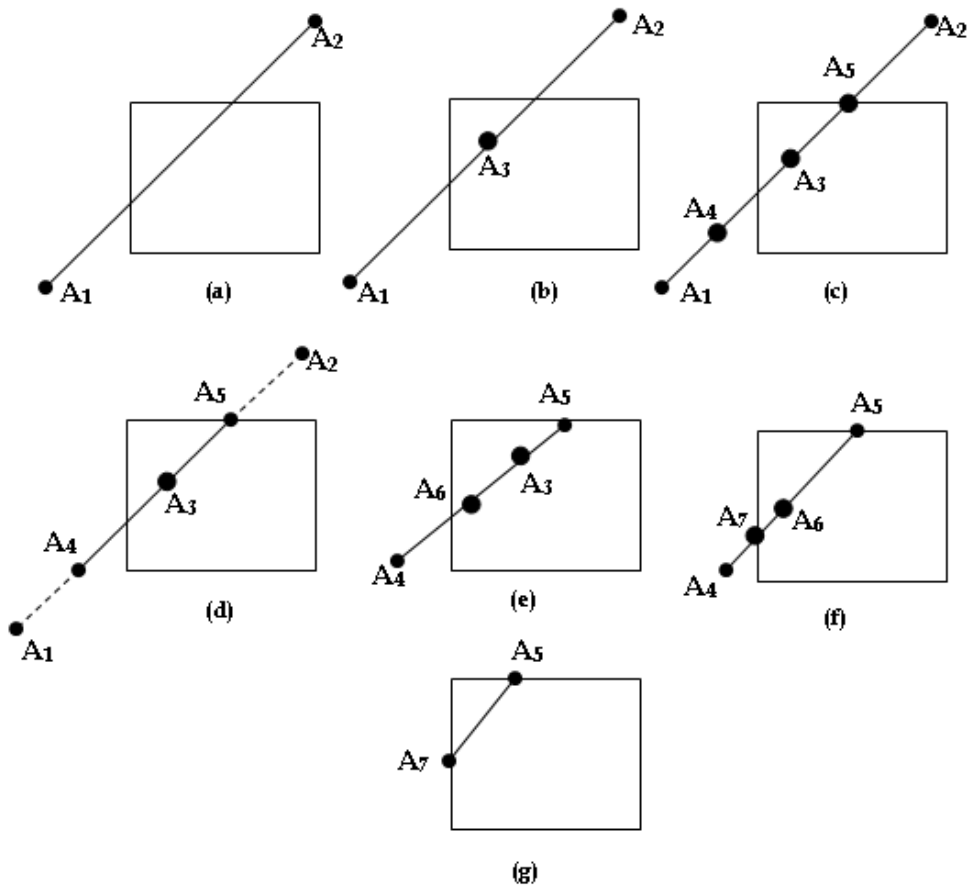
9.3.2 Midpoint Subdivision Algorithm

By now you know that the Cohen Sutherland algorithm involves calculating the intersection of the line with the window's boundaries. However, these calculations can be avoided by subdividing the line at its midpoint. This subdividing process is repeated until you get the line segment completely visible or completely invisible.

In the Midpoint Subdivision algorithm, the line is first tested for visibility. If the line is completely outside the window, then it is invisible and is rejected. If the line is visible, then the line is within the window and is drawn. On the other hand, if the line is partially visible, then it is subdivided into two parts. Each line segment is again tested for visibility. This subdivision process is repeated until you get a line segment that is completely visible (inside the window) or completely invisible (outside the window).

The figure 9.6 depicts Midpoint Subdivision process.

Figure 9.6: Midpoint Subdivision



In figure 9.6, the line A_1A_2 is partially visible. This line is subdivided into two equal parts A_1A_3 and A_3A_2 (figure 9.6 (b)). The visibility tests are then applied on these two line segments and found to be partially visible. Each of the line segments is further divided to obtain the midpoints A_4 and A_5 of A_1A_3 and A_3A_2 respectively (figure 9.6 (C)). You can observe from figure 9.6 (d) that A_1A_4 and A_5A_2 are completely invisible and hence they are rejected. The line segment A_3A_5 is completely visible while the line segment A_4 A_3 is partially visible. The line segment A_4A_3 is further subdivided into two equal parts A_3A_6 and A_6A_4. Now, the line segment A_6A_3 is completely visible whereas the line segment A_4A_6 is partially visible. The line segment A_4A_6 is further subdivided to obtain midpoint A_7. Now, the line segment A_4A_7 is completely invisible and is rejected, and the line segment A_7A_6 is completely visible. Thus, you get a line A_7A_5 which falls into category 1 (visible).

Notes

The midpoint co-ordinates (xm, ym) of line joining (x1, y1) and (x2, y2) are given by

xm = x1+ x2 / 2

ym = y1 + y2/2

The Midpoint subdivision algorithm is written as below:

1. Read two endpoints of the line: A(x1, y1) and A(x2, y2).
2. Read two regions of the window (left top and right bottom): (Wx1, Wy1 and Wx2 and Wy2).
3. Assign 4-bit region codes for the two endpoints. Initialize code with bits 0000.

   Set Bit 1 if ($x < Wx_1$)

   Set Bit 2 if ($x > Wx_2$)

   Set Bit 3 if ($y < Wy_1$)

   Set Bit 4 if ($y > Wy_2$)
4. Check for line visibility

   (a) If the region codes for both the endpoints of the line are 0000 then the line is within the window and completely visible.

   Go to step 6

   (b) If the region codes for endpoints of the line are not 0000 and the bitwise logical AND is also not 0000, then the line is outside the window and completely invisible.

   Go to step 6

   (c) If the region codes for endpoints do not satisfy any of the above two conditions then that line intersects any one of the window boundaries, and only a segment of the line inside the window frame is visible.
5. The partially visible line is divided into two equal parts. Repeat step 3 through 5 until the line is completely visible or completely invisible.
6. Stop.

As Midpoint Subdivision algorithm involves repeated subdivision of the line segments, it is slower compared to Cohen Sutherland algorithm, which directly performs the calculations to determine the co-ordinates where the line intersects the window boundaries.

9.3.3 Liang-Barsky Algorithm

Many line clipping algorithms have been developed that perform more line testing in an effective way before proceeding to the intersection calculations. Cyrus and Beck developed a line clipping algorithm that performed a quick line testing and then proceeded to the intersection calculations. The algorithm was based on the analysis of the parametric line equations. Later in 1984, Liang and Barsky independently devised an even faster algorithm that used inequalities derived from basic parametric line equations. This algorithm determines and differentiates the clipping region and the visible region of the window.

> *Notes* Parametric equations are the equations where co-ordinates of points appear dependent on the parameters.

Consider a line segment with endpoints (x_0, y_0) and (x_{end}, y_{end}). The parametric form of the line is given as:

$x = x_0 + u\Delta x$

$y = y_0 + u\Delta y \qquad 0 \leq u \leq 1$

where, $\Delta x = x_{end} - x_0$ and $\Delta y = y_{end} - y_0$.

The Liang-Barsky algorithm combines the above parametric line equations with the point clipping conditions to obtain the inequalities as below:

$xw_{min} \leq x_0 + u \Delta x \leq xw_{max}$

$yw_{min} \leq y_0 + u \Delta y \leq yw_{max}$

The above equation can be split into four separate equations:

$xw_{min} \leq x_0 + u \Delta x$

$xw_{max} \geq x_0 + u \Delta x$

$yw_{min} \leq y_0 + u \Delta y$

$yw_{max} \geq y_0 + u \Delta y$

Each of these inequalities specifies the limits set for each clipping boundary. The first equation corresponds to the left hand boundary, the second equation corresponds to the right hand boundary, the third equation corresponds to the lower boundary, and the fourth equation corresponds to the upper boundary. These four inequalities can be expressed as:

$u.p_k \leq q_k$ where k = 1, 2, 3, 4

The parameters p and q can be defined as:

$k = 1 \quad p_1 = -\Delta x \quad q_1 = x_0 - xw_{min}$

$k = 2 \quad p_2 = \Delta x \quad q_2 = xw_{max} - x_0$

$k = 3 \quad p_3 = -\Delta y \quad q_3 = y_0 - yw_{min}$

$k = 4 \quad p_4 = \Delta y \quad q_4 = yw_{max} - y_0$

Consider k =1, the parameters defined for this value are $p_1 = -\Delta x$ and $q_1 = x_0 - xw_{min}$. Now insert these into the equation $u.p_k \leq q_k$. The result obtained is:

$-\Delta x \leq x_0 - xw_{min}$

Now rearranging the equation you will get:

$xw_{min} \leq x_0 + u \Delta x$

This indicates that k = 1 corresponds to the limits set by the left hand clipping boundary.

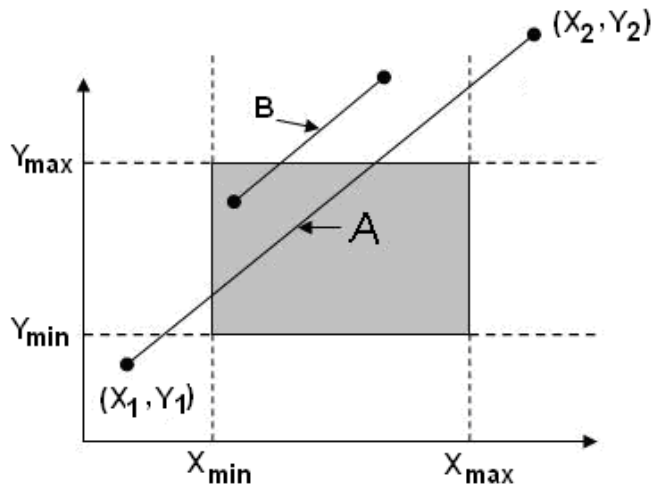
The line parallel to one of the clipping window edges has $p_k = 0$. Here, the value of k corresponds to that boundary where k = 1, 2, 3, and 4 corresponding to left, right, bottom and upper boundaries respectively. For that value of k, if $q_k < 0$ then the line lies outside the clipping boundary. If $q_k > 0$, then the line is inside the clipping boundary.

If $p_k < 0$, then the line extends from outside to inside the infinite extension of the clipping window edge. If $p_k > 0$, then the line extends from inside to outside. If p_k value is nonzero, then the value of u, which corresponds to the point where the extended line intersects the extension of the clipping window edge, can be calculated using the below equation:

$u = q_k / p_k$

To illustrate the above idea, consider figure 9.7 in which line **A** and a rectangular clipping window is defined. The clipping window boundaries are extended. The line A is not parallel to any of these boundaries. Then a check is performed to know whether the line passes each boundary from inside to outside or from outside to inside.

Figure 9.7: Liang-Barsky Algorithm



For each value of k, p_k is computed. As shown in figure 9.7, the first line crosses the lower boundary (k = 3). When k = 3, the parameters p_3 is - Δy. As p_3 is negative, the line passes from outside to inside of the boundary. The line then crosses the left hand boundary (k = 1) for which p1 is -Δx. As p_1 is negative, the line again passes from outside to inside of the boundary. Next, the line crosses the top boundary (k = 4) for which the parameters p_4 is Δy. As p_4 is positive, the line passes from inside to outside of the boundary. Finally, the line crosses the right hand boundary (k = 2) for which p_2 is Δx.

Thus, for each intersection compute parameter **u** with the help of the equation $u = q_k / p_k$. In this case, there are two values of **u** for outside to inside intersections and two values of **u** for inside to outside intersections.

Thus, for each line the value of parameters $\mathbf{u_1}$ and $\mathbf{u_2}$ are calculated. The value of $\mathbf{u_1}$ is determined by examining the rectangle edges where the line passes from the outside to the inside (p<0). For these edges, compute $r_k = q_k / p_k$. Similarly, the value of $\mathbf{u_2}$ is determined by examining the boundaries where the line passes from inside to outside (p>0). A value of r_k is computed for each of these boundaries. The value of $\mathbf{u_1}$ is considered as the largest of the set which consists of 0 and calculated **r** values. The value of $\mathbf{u_2}$ is the minimum of the set which consists of 1 and the various values of **r**. If $u_1 > u_2$, then the line is outside the clipping window and thus can be rejected. Otherwise, the values of the parameters are used to calculate the endpoints of the clipped line.

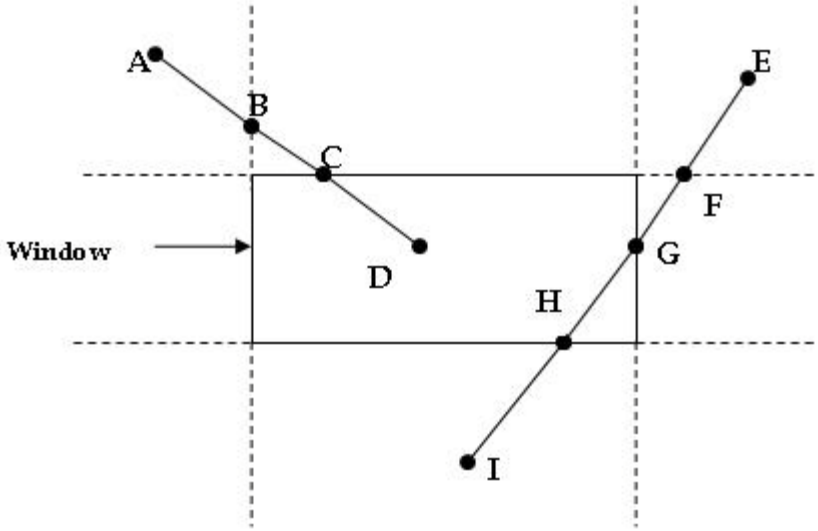
The Liang-Barsky algorithm uses the parametric line equations to speed up the intersection computations. In this algorithm, the intersections along a line path are computed only when the final value of parameters $\mathbf{u_1}$ and $\mathbf{u_2}$ is computed and hence there is no repetitive process. However, in Cohen Southerland algorithm the window intersections of the line are repeatedly calculated even though the line is outside the clipping window. Thus, the Liang-Barsky algorithm is considered more efficient than Cohen Sutherland algorithm.

Task

Consider a line B, as shown in figure 9.7, which is partially inside the clipping window. Show how this line segment can be covered within the clipping window. You can refer the above discussion relating to line segment A.

Lab Exercise

Figure 9.8: Line Clipping



Consider the above two lines AD and EI. Clip the above two lines such that the line segment DC and HG are visible. You need to show the clipping process using (a) Cohen Sutherland algorithm and (b) Midpoint Subdivision algorithm.

9.4 Summary

- Clipping is used to display the images within the window frame.
- Clipping algorithm is the procedure used to determine whether the portion of the object lies within or outside the window frame.
- Point clipping and line clipping are the two most common types of clipping.
- Point clipping removes the points that lie outside the window.
- Line clipping determines whether the line lies within or outside the clipping window and removes the line segments that lie outside the window.
- Line clipping includes three algorithms used to determine the line visibility, namely Cohen Sutherland Algorithm, Midpoint Subdivision Algorithm and Liang-Barsky Algorithm.
- Cohen Sutherland Algorithm determines whether the line is visible, invisible or clipping candidate.
- If the line is clipping candidates then Cohen Sutherland algorithm performs calculation of the intersection of the line with the window edge.
- Midpoint Subdivision Algorithm avoids the calculation involved Cohen Sutherland algorithm by simply subdividing the line at its midpoint until the line segment is completely visible or completely invisible.
- Liang-Barsky Algorithm based on analysis of the parametric line equations.

9.5 Keywords

Device Co-ordinate: A device co-ordinate is a device-dependent co-ordinate system. The origin is at the upper left corner, X increases to the left, Y increases down. As its units are pixels it is also known as pixel co-ordinates.

GNOME: GNU Object Model Environment.

Viewport: It is a window manager window in computer graphics. It is basically a rectangular viewing region.

Window Manager System: It controls the appearance of the windows within a windowing system in a graphical user interface.

9.6 Self Assessment

1. State whether the following are true or false:
   (a) When you input and transform the picture of the object in the window on the screen viewport, the picture regions surrounding the window frame also disappears.
   (b) In point clipping, the edges of the clip window can either be world-co-ordinate window boundaries or viewport boundaries.
   (c) If the line is either completely visible or completely invisible, then the line is clipping candidate.
   (d) The point whose region code is 0000 is inside the clipping window.
   (e) In Midpoint Subdivision Algorithm, the subdividing process is repeated until you get the completely visible or completely invisible line segment.
   (f) The equation xwmin $\leq$ x0 + u Δx corresponds to the right hand clipping boundary.
2. Fill in the blanks:
   (a) In computer graphics, the portion that is left outside the window frame is called as ……..
   (b) The …….. reduces the calculations by linking the viewing and geometric transformation matrices.
   (c) The line clipping depends on the position of the endpoints with reference to the ……..
   (d) A 4-bit binary code, called as …….., is assigned to each endpoint of the line.
   (e) Cyrus and Beck developed a line-clipping algorithm which was based on ……..
3. Select the suitable choice for every question:
   (a) The picture regions against which the object is to clip is called as a …….
      (i) Clip window
      (ii) World-co-ordinate
      (iii) Clipping candidate
      (iv) Clipped part

(b) The sign(x-xmax) indicates that the:

(i) Endpoint is above the window (ii) Endpoint is to the left of the window

(iii) Endpoint is to the right of the window (iv) Endpoint is below the window

(c) The endpoints of the line which lies within the window fall under …….. category.

(i) Visible (ii) Invisible

(iii) Clipping candidate (iv) Parametric

(d) Clipping algorithm can be applied in world co-ordinates such that the picture of the object inside the window is mapped to……...

(i) Clip window (ii) Transformation matrices

(iii) Viewport (iv) Device co-ordinates

9.7 Review Questions

1. What is clipping? Mention the types of clipping.
2. Explain how point clipping works.
3. "Line clipping depends on the position of the endpoints with reference to the window edges". Explain with diagram.
4. What are the two steps followed in Cohen Sutherland algorithm to determine the category of a line?
5. In Midpoint Subdivision algorithm, when do you start subdividing a line?
6. Explain with diagram the Midpoint Subdivision algorithm.
7. Explain Liang-Barsky Algorithm.

Answers: Self Assessment

1. (a) False (b) True (c) False (d) True (e) True (f) False
2. (a) Clipped part (b) Viewport clipping (c) Window edges
   (d) Region code (e) Parametric line equations
3. (a) Clip window (b) Endpoint is to the right of the window
   (c) Visible (d) Device co-ordinates

9.8 Further Readings

Books

Godse A.P, Godse D.A. *Computer Graphics And Multimedia,* Technical Publications, Pune

Sinha and Udai, *Computer Graphics,* Tata McGraw-Hill, New Delhi

Zhigang Xiang, *Computer Graphics,* Tata McGraw-Hill Education

Online link

http://i.thiyagaraaj.com/tutorials/computer-graphics/clipping-techniques/1-point-clipping

www.skytopia.com/project/articles/compsci/clipping.html

Unit 10: Clipping II

CONTENTS

Objectives

Introduction

10.1 Polygon Clipping

10.2 Projection

10.3 Summary

10.4 Keywords

10.5 Self Assessment

10.6 Review Questions

10.7 Further Readings

Objectives

After studying this unit, you will be able to:

- Explain polygon clipping
- Discuss projections

Introduction

The projection is transforming of points and lines from one plane to the other plane. The projections are done by connecting the corresponding points on two planes.

Did you know? In projection it is important to control:

1. Projection type that can be perspective or orthographic.
2. Field of view and aspect ratio of image.
3. Near and far clipping panes.

In this unit, you will learn the third most common types of clipping, that is, polygon clipping. You will learn various clipping algorithms that are developed for extracting a part of a defined scene for viewing, displaying multi-window environment, drawing, painting, and so on. At the end of this unit you will learn about projection, which helps transform 3-dimensional (3-D) objects on to a 2-dimensional (2-D) projection plane. The projection is considered as the solution to the mismatch between 3-D objects and 2-D displays.

10.1 Polygon Clipping

A polygon comprises a set of lines. A polygon is formed by placing the line segments on the plane where in each line segment shares its endpoints with other line segment and thus forming a closed region on the plane. Here, you need to modify the line clipping algorithm to clip the polygon. The polygon is tested to see if it is within the display window. The polygon edges are clipped that lie outside the window boundary. The edge segments that are outside the window are clipped. The polygon edges must be handled in sequence. The clipping polygon is of two types, namely concave polygon and convex polygon.

The polygon is said to be concave when a line is obtained after joining the two interior points of the polygon which lies totally inside the polygon. Where as polygon is said to be convex when there is at least one pair of point such that the line joining the points inside the polygon does not lie completely inside the polygon. The figure 10.1 provides a clear idea of concave polygon and figure 10.2 provides a clear idea of convex polygon.

Figure 10.1: Concave Polygon

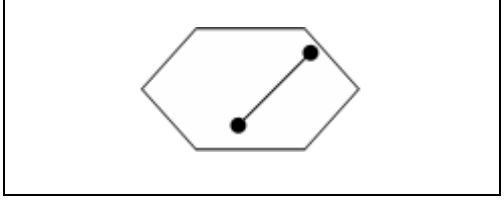
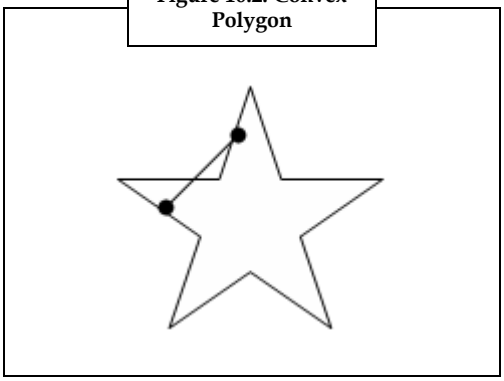


Figure 10.2: Convex Polygon



The polygon clipping can be further explained using the following two algorithms.

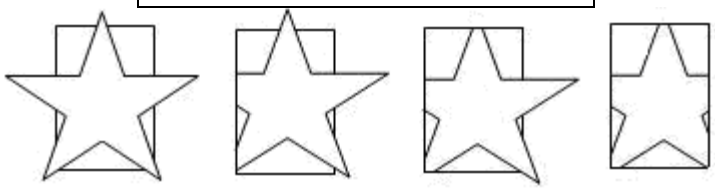
1. Sutherland-Hodgeman Algorithm
2. Weiler-Atherton Algorithm

Sutherland-Hodgeman Algorithm

Polygon clipping is relatively straightforward and can be easily implemented. Sutherland-Hodgeman algorithm uses the **divide and conquer** strategy to solve the problem related to polygon clipping. First, this algorithm clips the part of the polygon that is adjacent to the right or left clipping boundary. Then, this partly clipped polygon is clipped against the top boundary and the same process is repeated for the two remaining boundaries.

The figure 10.3 below depicts a general idea about Sutherland-Hodgeman clipping algorithm.

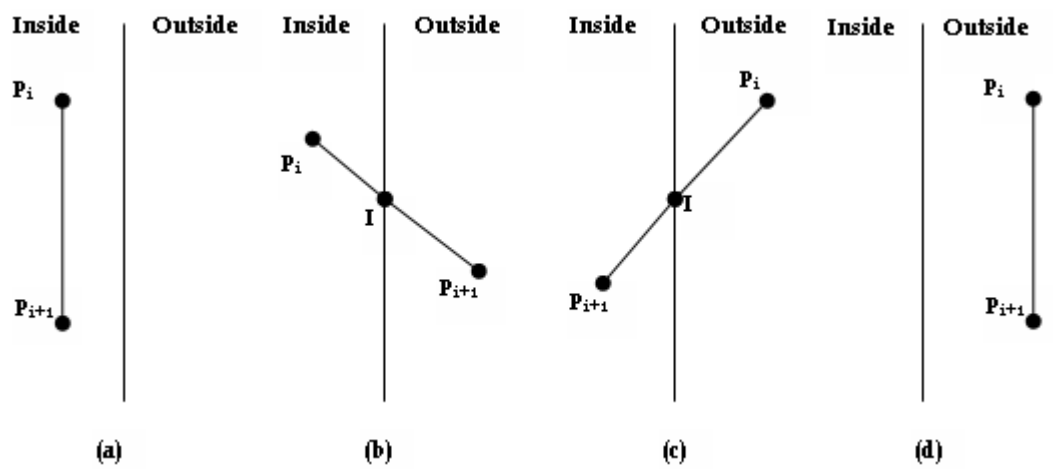
Figure 10.3: Sutherland-Hodgeman Algorithm



A single polygon can be clipped against each edge of the window at a time. In the above figure 10.3, the Southerland-Hodgeman algorithm clips around the edges of the clipping rectangle (Clipping window). The polygon is clipped against the four clipping edges, one after another. This algorithm does clipping by traversing all the edges of the polygon and creates the clipped polygon by following the below four rules:

1. If both points are inside the clipping window, as in figure 10.4 (a), then add the second one (Pi+1) to the result polygon.
2. If the current point is inside the clipping window and the next one lies outside, as in figure 10.4 (b), then add the intersection point 'I' to the result polygon.
3. If the current point is outside the clipping window and the next one lies inside, as in figure 10.4 (c), then add the intersection point 'I' and the next point Pi+1to the result polygon.
4. If both points are outside the clipping window, as in figure 10.4 (d), then insert none of them to the result polygon.

Figure 10.4: Divide and Conquer Approach



Thus, you get a new polygon (result polygon), which is obtained by clipping 'around' the edges of the clipping window.

Polygons clip against convex clipping windows in the Sutherland-Hodgeman polygon clipping algorithm. It does so by clipping the subject polygon against each clip edge, thereby creating intermediate subject polygons. The Sutherland-Hodgeman algorithm can easily extend to three dimensions.

Disadvantages of Sutherland-Hodgeman Algorithm

Some of the disadvantages of Sutherland-Hodgeman algorithm are as follows:

1. Sutherland-Hodgeman algorithm constantly generates a single output polygon even though the clipped polygon is concave and is expected to produce multiple polygons.
2. The polygons are linked with overlapping segments along the edge of the clipping rectangle. The result may not be desired because it depends on the application.

Weiler-Atherton Algorithm

The Weiler-Atherton algorithm can clip a concave polygon. The clipping region is called as clip polygon and the polygon to be clipped is referred to as subject polygon. In this algorithm, you have an arbitrary vertex of the subject polygon. You need to trace around its border in the clockwise direction until you find an intersection with the clip polygon. While tracing, you need to:

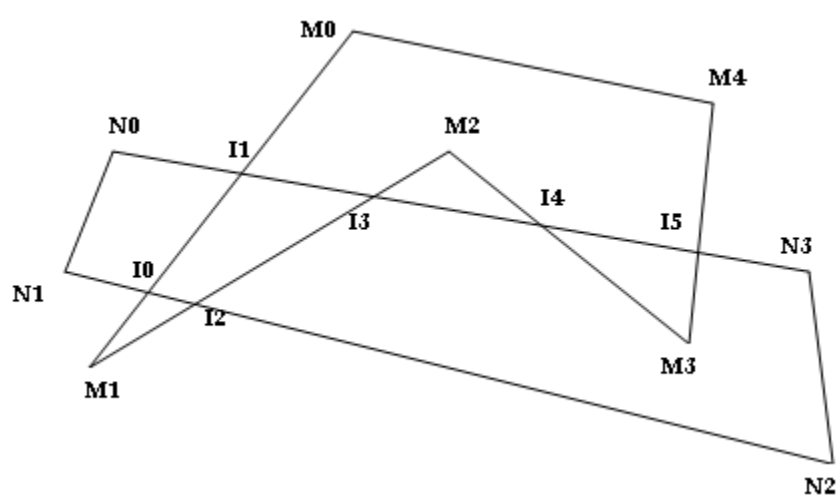
1. Record the intersection point when the edge enters the clip polygon. Continue to trace the subject polygon.

2. Record the intersection point when the edge leaves the clip polygon. Make a right turn to follow the clip polygon. You need to consider the clip polygon as subject polygon and vice versa and then proceed as before.

When the path of traversal forms a sub-polygon, you output the sub-polygon as a part of the overall result. You need to continue tracing the rest of the subject polygon from the intersection point that marks the beginning of an untraced edge. When the entire border of the subject polygon can be traced once then the algorithm terminates.

The Weiler-Atherton algorithm is mainly used in modeling to find the union, intersection and difference of the polygons. Consider two polygons M and N with the vertices as shown in the figure 10.5.

Figure 10.5: Weiler-Atherton Algorithm



The following steps are followed to obtain the sub-polygon:

1. The intersections of sides are calculated and placed in two lists:

   For polygon M: M0, I1, I0, M1, I2, I3, M2, I4, M3, I5, I4

   For polygon N: N0, N1, I0, I2, N2, N3, I5, I4, I3, I1

   The list of entering points for M is I1, I2, and I4

2. Begin with the first point of entering list I1

   (a) Take the next point on the list for M

   The next point I0 is the exiting point for M

   (b) You get the list N. Here, you take the next point I2

   I2 is the exiting point for N and thus you enter list M
   I3 is the exiting point for M and thus you enter list N

   (c) Retrieve point for beginning I1

   The first common area of polygon I1 I0 I2 I3 is obtained

3. Delete the entering points in this area I1 and I2

4. Take last point in entering list I4

   You find M3 in M list and then you get I4
   I5 is exiting point for M and thus you enter list N and find I4

   The second common area of polygon I4 M3 I5

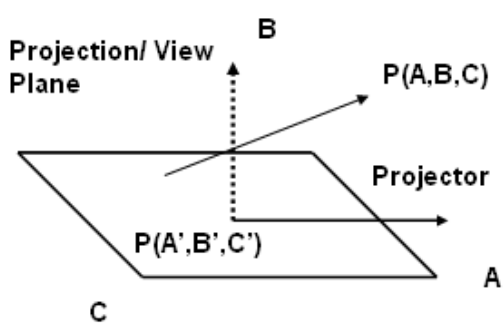
The entering list of points is now empty and you obtain all common areas

This algorithm gives false result if the polygons are crossed. This is because, with crossed polygons you cannot decide whether a point is entering or existing the polygon. These common areas form the sub-polygon, which is the final result of the clipping process.

10.2 Projection

Projection can be defined as a mapping of point P (A, B, C) to its image P′ (A′, B′, C′) on the projection plane that represents the display surface. The mapping is ascertained by a projection line called the projector that passes through P and intersects the view plane. The point of intersection is P′. The outcome of projecting an object mainly depends on the spatial relationship among the projectors that project the points on the object and the spatial relationship between the projectors and the view plane.

Figure 10.6 Projection



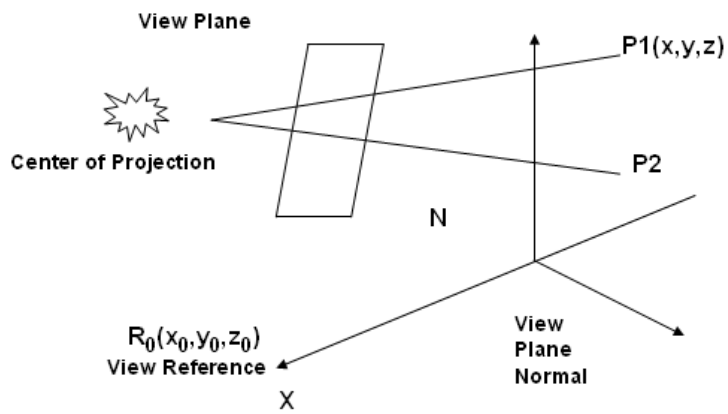
There are two commonly used methods of projection. They are:

1. Perspective Projection
2. Parallel Projection

Perspective Projection

Basic Principle: The techniques of perspective projection are an overview of the principles used by artists in arranging the perspective drawings of three–dimensional objects and scenes. The eye of the artist is positioned at the center of projection and the plane holding the canvas becomes the view plane as shown in figure 10.7. With the help of a projector, an image point that goes from an object point to the center of projection is determined.

Figure 10.7 Perspective Projection



Perspective drawings are distinguished by perspective foreshortening and vanishing points. Perspective foreshortening is the illusion of the objects and lengths that appear to be small as their distance from the center of projection increases. The illusion is nothing but sets of parallel lines that appear to meet at some point. This point is called as vanishing point. The major vanishing points are formed by intersecting lines parallel to x, y, or z axes. The number of principal axes that are intersected by the view plane determines the number of principal vanishing points.

Perspective Anomalies

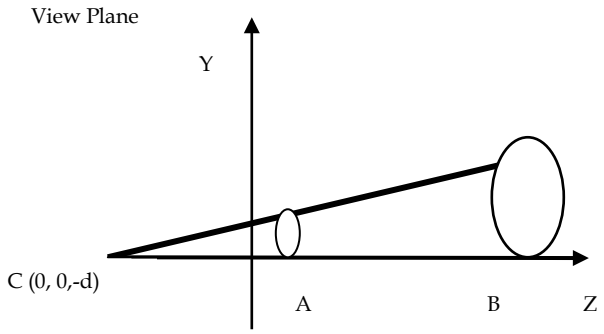
The process of constructing a perspective view introduces certain inconsistencies that improve practicality in terms of depth cues but alter the actual sizes and shapes.

The following are the perspective anomalies:

1. ***Perspective Foreshortening:*** The farther the object is from the center of projection, the smaller it appears (its projected size becomes smaller). As shown in figure 10.8, object B is twice the size of object A. However, both the objects appear to be of same size when projected on the view plane.

The figure 10.8 depicts perspective foreshortening.

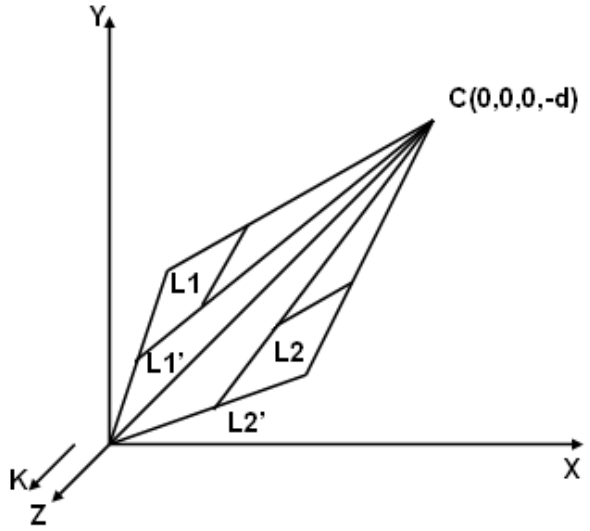
Figure 10.8 Perspective Foreshortening



2. ***Vanishing Points:*** Vanishing points are the projections of lines that are not perpendicular to the view plane but appear to meet at a point on the view plane. As in figure 10.9, the projections L1′ and L2′ of lines L1 and L2 respectively appear to meet at the origin 'O'.

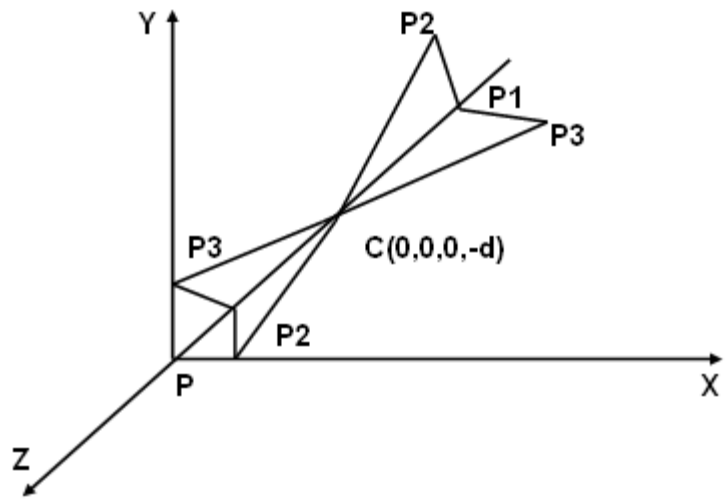
Example: An illusion that railroad tracks meet at a point on the horizon forms a good example of vanishing points perspective anomaly.

Figure 10.9 Vanishing Points



3. ***View Confusion:*** Objects that are behind the center of projection are projected in an inverted manner. As shown in figure 10.10.

Figure 10.10 View Confusion



4. ***Topological Distortion*:** Consider a plane that is parallel to the view plane and passes through the center of projection. The perspective transformation projects the points of the plane to infinity. A finite line segment that joins a point that lies in front of the viewer to a point at the back of the viewer is actually projected to a broken line of infinite extent.

Parallel Projection

Parallel projections are the three-dimensional objects that are placed on a two-dimensional projection plane such as screen, paper, or film. The main rule of all parallel projections is to choose a direction and construct a ray that starts at a general point on the object and goes in the direction of the plane. The point where the ray intercepts the projection plane becomes the projection. The process is repeated for all the points on the object, creating a set of parallel rays. Hence, this is known as parallel projections.

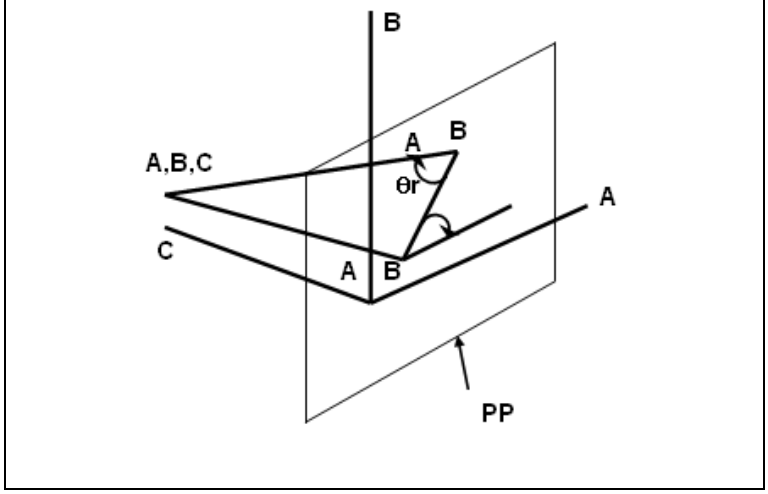
There are two types of parallel projections. They are:

1. ***Orthographic Projections:*** This is the simpler of the two parallel projections. In orthographic projections, the direction of projection is perpendicular to the projection plane.
2. ***Oblique Projections****:* If the direction of projection is not perpendicular to the projection plane, then it is called as oblique projection.

Example: In figure 10.11, consider a parallel projection of a point (A, B, C). (Note the left-handed co-ordinate system). The projection plane is at C = 0. A and B are the orthographic projection values, and AP and BP are the oblique projection values (at an angle with the projection plane).

The figure 10.11 depicts parallel projections.

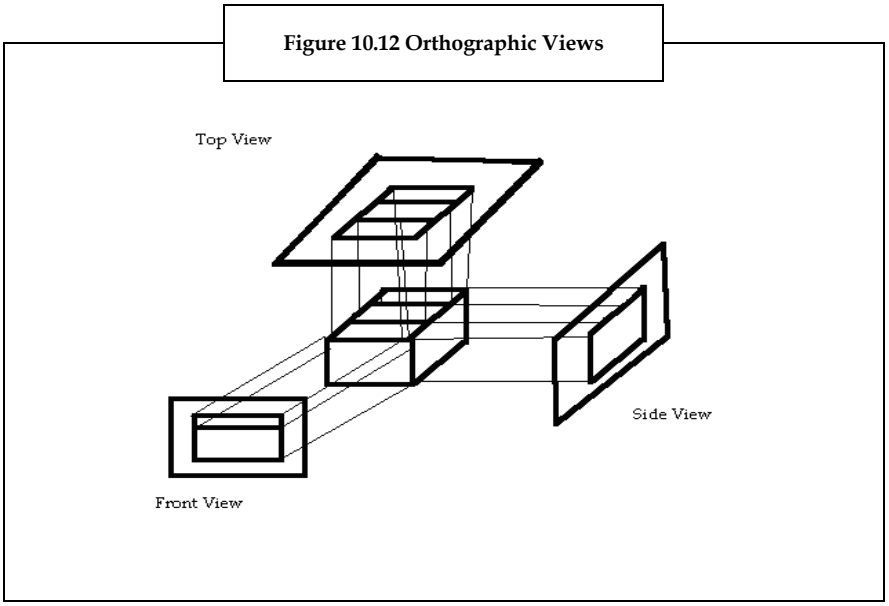
Figure 10.11 Parallel Projections



Orthographic Projections

Engineering drawings often use front, side, and top orthographic views of an object. Figure 10.12 illustrates the three orthographic views of an object.

Figure 10.12 Orthographic Views

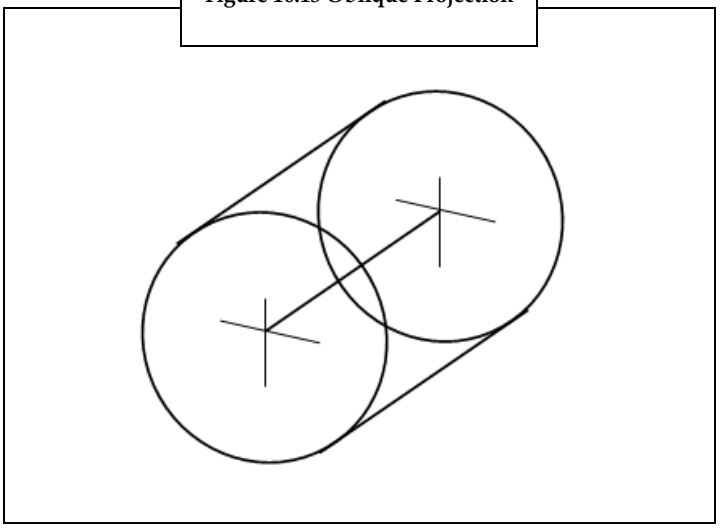


Orthographic projections that show more than one side of an object are called axonometric orthographic projections. An isometric projection is the most basic type of axonometric projection, where the projection plane intersects each co-ordinate axis in the model co-ordinate system at an equal distance. This projection is a method used to show an object in all three dimensions in a single view.

Oblique Projections

Oblique projection projects an image by intersecting the parallel rays from the three-dimensional source object with the drawing surface. In oblique projection, the parallel lines of the source object produce parallel lines in the projected image. The projectors produce the projected image by intersecting the projection plane at an oblique angle.

Figure 10.13 Oblique Projection



As shown in figure 10.13, oblique projection is not really a 3-D system, it is a 2-D view of an object with forced depth. One way to draw using an oblique view is to draw one side of the object in two-dimensions that is flat. Then, draw the other sides at an angle of 45 degrees. Instead of drawing the sides full size they are only drawn with half the depth creating forced depth - adding an element of realism to the object. Even with this forced depth, oblique drawings look very unconvincing to the eye. For this reason, oblique projection is not often used by professional designers and engineers.

10.3 Summary

- The line clipping is modified to clip the polygon. The clipping polygon is of two types: concave polygon and convex polygon.
- The Sutherland-Hodgeman polygon clipping algorithm applies divide-and-conquer strategy to clip a polygon.
- Weiler-Atherton Algorithm illustrates both the subject polygon and the clip polygon with the help of list of vertices.
- The projection maps a point to its image on the display surface. The two basic method of projection are perspective projection and parallel projection.

10.4 Keywords

***Anomaly*:** The deviation from the general order, type or rule.

***Isometric*:** This is the act of presenting equality in dimension and measurements.

***Orthographic*:** In mathematical terms, this means perpendicular.

***Polygon*:** A polygon is a closed, geometric, two dimensional plane figure formed by straight lines.

10.5 Self Assessment

1. State whether the following are true or false:
   (a) The polygon edges are clipped that lie outside the window boundary.
   (b) Parallel projections are the two-dimensional objects that are placed on a three-dimensional projection plane.
   (c) The main rule of all parallel projections is to choose a direction and construct a ray that starts at a general point on the object and goes in the direction of the plane.
2. Fill in the blanks:
   (a) Sutherland-Hodgeman algorithm uses a …….. strategy to solve the problem related to the polygon clipping.
   (b) The number of …….. that are intersected by the view plane determines the number of principal vanishing points.
   (c) Orthographic projections that show more than one side of an object are called …….orthographic projections.
3. Select the suitable choice for every question:
   (a) In Weiler-Atherton Algorithm, when the path of traversal forms a……you output the sub-polygon as part of the overall result.
      (i) Subject polygon
      (ii) Clip polygon
      (iii) Sub polygon
      (iv) Convex polygon
   (b) The …… is one of the perspective anomalies, where the far the object is from the center of projection, the smaller it appears.
      (i) Perspective foreshortening
      (ii) Vanishing points
      (iii) View confusion
      (iv) Topological distortion

10.6 Review Questions

1. What is polygon clipping? Explain Sutherland-Hodgeman polygon clipping algorithm.
2. Mention the important points that need to be considered in Weiler-Atherton Algorithm.
3. What is projection? Mention the types of projection in computer graphics.
4. What are perspective anomalies?
5. Explain parallel projections.

Answers: Self Assessment

1. (a) True (b) False (c) True
2. (a) Divide-and-conquer (b) Principal axes (c) axonometric
3. (a) Sub polygon (b) Perspective foreshortening

10.7 Further Readings

Books

Godse A.P, Godse D.A. *Computer Graphics And Multimedia,* Technical Publications, Pune

Sinha and Udai, *Computer Graphics,* Tata McGraw-Hill, New Delhi

Zhigang Xiang, *Computer Graphics,* Tata McGraw-Hill Education

Online link

http://i.thiyagaraaj.com/tutorials/computer-graphics/clipping-techniques/1-point-clipping

www.skytopia.com/project/articles/compsci/clipping.html

Unit 11: Hidden Surfaces

CONTENTS

Objectives

Introduction

11.1 Z-Buffer

11.2 Binary Space Partitioning

11.3 Painter's Algorithm

11.4 Warnock Algorithm

11.5 Ray Tracing vs. Rasterization

11.6 Determination

11.7 Removal

11.8 Summary

11.9 Keywords

11.10 Self Assessment

11.11 Review Questions

11.12 Further Readings

Objectives

After studying this unit, you will be able to:

- Discuss the different types of buffers - Z-Buffer, C-Buffer, S-Buffer
- Explain the concept of binary space partitioning
- Discuss Painter's algorithm
- Provide overview about Warnock algorithm
- Comprehend Ray tracing vs. Rasterization
- Explain the concept of determination
- Define removal

Introduction

In a 3-dimensional (3-D) objects, only the front surfaces and edges are visible to the viewer. In other words, the surfaces and edges which are at the rear end are not visible. Moreover, this can also be observed in an overlapping 2-dimensional (2-D) object. These surfaces and edges which are not visible to the viewer are known as hidden surfaces and hidden edges respectively. The viewing parameters such as rear view, side view, top view and bottom view decide the visibility of an object. In case an object hides the other object or a part of an object hides another part of the same object, you will find that only the object in front is visible.

The most important question here is to determine which lines or surfaces of the objects should be visible. This process is termed as hidden surfaces, hidden line elimination, or visible surface determination. The hidden line or hidden surface algorithm determines the lines, edges, surfaces that are visible or invisible to a viewer. These algorithms are classified into two categories. One category is the one that deals with the object definitions directly and the other category is the one that deals with the object definitions through projected images. These two algorithms are called object space and image space methods respectively.

Object space method compares objects and parts of objects to each other in order to decide the visibility. The different types of object space methods are back face and Painter's algorithm.

Image space method decides visibility point by point at each pixel position on the view plane. The different types of image space methods are area subdivision, Z-buffer, and scan line.

Did you know? Image space method requires more work than object space method. Therefore, most algorithms should theoretically be applied in object space. This is not the case in general. Image space methods are considered to be more efficient because it is easier to take advantage of consistency in raster scan implementation of an image space method.

11.1 Z-Buffer

Z-buffer is one of the simplest and commonly used image space approaches to eliminate hidden surfaces. This is referred to as the z-buffer, since depth of an object is mainly calculated from the view plane along the z axis of a coordinate system.

A 3-D shape is a collection of 3-D surfaces or planar surface patches. A planar equation can be used to estimate each of these surfaces. The planar equation helps to evaluate the **Z** co-ordinate value of each of the planar surface points in world co-ordinates.

Notes

Once you have defined the Z buffer, Direct3-D will automatically stop rendering triangles that are hidden by other triangles. The planar equation helps to evaluate the Z co-ordinate value of each of the planar surface points in world co-ordinates.

When 3-D shapes are projected onto the 2-D computer screen, the position of each pixel on the view plane is represented by x and y coordinates, whereas the z-value holds the information about depth. For every (X, Y) screen location, there is one or more 3-D surface points. The different **Z** co-ordinate values of these 3-D surface points are defined by the planar equations of the surfaces. These **Z** values are then compared to determine the visible surface points. A surface is considered visible if it has the minimum **Z** value. It is important to know that in the corresponding (X, Y) location of the screen, the color of the visible surface point is painted.

Did you know? The Z-buffer algorithm was invented by E. Catmull around the 1970s. It is very simple and can handle just about any 3-D primitive. Since then, it has become the most preferred choice for hardware accelerated 3-D boards.

The equation of a planar surface is given by, $Ax + By + Cz + D = 0$, where vectors (A, B, C) are normal to the plane and D is constant.

As mentioned earlier, Z-buffer algorithm operates on planar surfaces of a 3-D shape in order to determine visible surfaces. Therefore, the **Z** value of the plane at location (X, Y) is given by:

$$Z = - Ax - By - D/ C \quad \text{.... (Eq.11.1)}$$

Consider a 3-D viewpoint looking at a planar surface, where the **Z** value of the surface gives the depth of the surface from the viewpoint. Therefore, this algorithm is also known as **depth buffer algorithm**. For planes with known equations, the visibility of planes can be determined by comparing **Z** values of these planar points.

Let us assume that **Z** values are written in a 2-D matrix of the same size as a computer screen and let us henceforth refer to it as Z-buffer. Initially, the Z-buffer values for all the elements of the entire 2-D matrix are set to a maximum possible number. Also, initialize a 2-D screen matrix of same size as Z-

buffer with background color. The Z-value of each (X, Y) point for a plane of a 3-D object to be projected on the computer screen is evaluated with the help of equation 8.1. This value is then placed in the (X, Y) location of the Z-buffer if it is less than the initial **Z** value set at (X, Y) location. For the next plane of the 3-D object, the **Z** values are calculated in the similar manner and compared with the existing Z-buffer values.

The plane with the minimum Z-buffer value is visible. It is assumed that the visible plane is closer to the viewer. This visible plane obstructs the planes having higher **Z** values at corresponding pixel locations. After the visibility test using Z-buffer for a particular screen location (X, Y), the color of the plane having the least Z-buffer value at that point is painted. If the **Z** value is not updated for some pixel position i.e. if screen location is not painted new then the background color exists in that position. This happens since the background color is already initialized.

Z-Buffer Algorithm

The Z-buffer algorithm is among the most preferred routines in present environment. It is easy to implement and simple to use and is more often found in hardware. The idea behind its implementation is very simple. You just have to assign a Z value to each polygon and then identify the one (pixel by pixel) having the smallest value.

Algorithm

1. Initialize the Z-buffer values to maximum possible integer.
2. Initialize a 2-D screen matrix of same size as Z-buffer along with background color.

```
For each plane
{
  For each point on the plane
  {
     Using equation 11.1 calculate the Z value of the planar point
     If the obtained Z value < existing Z-buffer value at the corresponding frame buffer location
     {
      Store Z value in Z-buffer and paint the pixel position of the screen with the color of the plane.
     }
  }
}
```

Since, equation 11.1 is linear in (X, Y), notice that the Z value of a frame buffer location along the row of the frame buffer at location (X+1, Y) can be easily calculated by adding (-A/C) with the Z-buffer values at (X, Y) as shown below.

$$Z(x+1, y) - Z(x, y) = (-A(x+1) - By - D)/C - (-Ax - By - D)/C = -A/C$$

Similarly, z-values at location (X, Y+1) can be calculated by adding (-B/C) with the Z-buffer values at (X, Y) as shown below.

$$Z(x, y+1) - Z(x, y) = (-Ax-B(y+1)-D)/C - (-Ax -By-D)/C = -B/C$$

Z-buffer values can be calculated from the linear interpolation of the Z-buffer values which are calculated at the vertices of the plane with the help of equation 11.1.

Some of the advantages of Z-buffer are that:

1. It can be used easily.
2. It can be implemented easily in object or image space.
3. It can be executed quickly, even with multiple polygons.

With advantages the Z-buffer also has some disadvantages as

1. It requires lot of memory.
2. It requires additional code to do transparent surfaces.

Now let us look at an example of Z-buffer.

Example: Z-Buffer Algorithm for Cube

```
Initialize Z-buffer values to the maximum possible integer.
Initialize a 2-D screen matrix of same size as Z-buffer with background color
For each surface P having color 1 and for plane equation
{
For each pixel (x1,y1) in the 2-D projection of P
{
Compute Z value at (x1,y1)
If z value < z-buffer value at (x1,y1)
{
On Pixel ( x1 , y1 ,1);
Z-buffer value at (x1, y1) = z value
}
}
}
```

Let us study about other available buffers such as C-buffer and S-buffer.

C-buffer is also known as the coverage buffer. For the C-buffer the spans are drawn directly onto the screen. Whenever a span gets clipped on to the left side, the clipping is done and then the span that caused the clip is extended along the length of the clipped span. At the end of the rendering process each scan line on screen is represented by a single entry which is called as the C-buffer.

S-buffer is also known as the surface buffer. It is basically a structure in which a track is maintained of the parts of each scan line that are filled with pixel data already. It can be filled with the spans that make up the polygon when we rasterize it. The S-buffer thus consists of either a linked list or a tree that holds the spans that are sent to a specific scan line in the correct order. When a new span is inserted on an already existing span then the new span is either clipped or split or rejected.

Example: *The 'x' line is the new span, '=' means spans already in the tree:*

```
Xxxxxxxxxxxx          (New span)
============          (Span already exists)
```

Here, portion of the right side of this span is clipped of so that the new S-buffer for this screen line becomes:

```
xxxxxxx============
```

11.2 Binary Space Partitioning

Algorithms that are used to determine the proper obstruction between objects should be able to quickly determine the relative depths of objects or parts of objects. If the viewpoint and view direction are changed in an image space algorithm then the process has to be repeated from scratch. As an alternative to the above process a Binary Space Partitioning (BSP) tree is constructed for the scene that is independent of the viewpoint location. The BSP tree breaks up the scene into regions with the spatial relationship of the same regions built into the tree. The space is subdivided until the object in each of the regions is definite in its relationship with the objects in the other regions.

The leaf node of the BSP tree where the viewpoint is located holds the closest object. As we move up one node in the tree and down into the other leaf, the second closest object is identified. This way the child-node of other nodes in the tree pointing towards the collection of the objects can be identified as the next closest and farthest objects.

BSP trees are also used to speed up ray tracing (which will be discussed in the unit further) and in planning the motion of robots. Let us now focus on what happens in the BSP.

Firstly, the polygons are sorted for display in back to front order. For this sorting **Controlling Object Transparency Test** which compares all the vertices of one polygon against the plane of another polygon is conducted. If the plane intersects the polygon, divide the polygon along the plane and extend the above test. Take one polygon as the reference polygon and compare all other polygons to it in order to check whether it lies in front of the reference polygon or behind it. According to the result group then divide into two groups. Select a polygon for each of these subgroups and use it to separate the subgroups. Firstly, the polygons are sorted for display in back- to- front order.

Repeat the above process until all polygons have been sorted. The resultant outcome is obtained as a binary tree. At each node of the tree lies a polygon. The root of the tree is the reference polygon and where as at each node of the tree, it is a polygon. All the polygons that are in front of the reference polygon lies in one sub tree or branch. The polygons which lie behind it are in the other branch. Perform an in-order traversal, so that the polygons can be achieved in back to front order.

A lot of information and storage for sorting of the polygons are required in this algorithm. For constructing and processing BSP trees we may use hardware implementations.

11.3 Painter's Algorithm

The Painter's algorithm works similarly in the manner as an artist who paints his/her canvas with background and foreground colors. According to the painter's desire, the canvas is initially painted with the background color. Then the painter creates objects using foreground colors over the background color. It is assumed that the background color does not affect the drawing of foreground objects. The same thing happens even in the frame buffer. Initially, the frame buffer is filled with the background color. One by one the polygons representing object surfaces are drawn keeping the farthest from the viewer first and nearest from the viewer last.

This algorithm requires enough computing effort to determine the exact order of the polygons to be drawn. Each polygon is compared to see which precedes the other.

Order of Polygons

Finding the exact order of all the polygons is a complicated and time consuming task. To simplify it, different sort of tests have to be applied before applying the sorting process itself. The minimax test or boxing test can be done to determine whether any two polygons are overlapping. To do this test, let us consider enclosing two boxes around specific polygons. If these two boxes are not overlapping then this is true for the polygons also. These boxes in general are defined by the co-ordinates x-max, y-max, x-min, and y-min. If the minimax test result is true then the polygons do not overlap. However, if the test result is false, you cannot ascertain whether the polygons overlap or not. Therefore, other tests are required.

If the minimax test is applied on the Z co-ordinate then the relative ordering of two polygons which overlap on X-Y plane can be determined. In addition, if the smallest Z value for a polygon is greater than the highest Z value on the other polygon then the former polygon is in front of the latter.

Algorithm

1. Start.
2. Make a polygon list sorted according to their maximum z-value.
3. Set priorities to the polygons starting from the end of the list.
4. While the list is not empty do
   (a) start
   (b) identify all polygons P2 whose Z range overlaps with the preceding polygon P1
   (c) for each P2 do
   obscurity test for P2

if any member of P2 does not obscure P1, draw P1 and remove it

if any member of P2, say Q, obscures P1, swap P1 with Q

tag Q

if Q is already tagged, use the plane containing P1 to separate Q into polygons Q1 and Q2

eliminate Q from the polygon list

include Q1 and Q2 in the polygon list in z-sorted manner

(d) end

5. end

Advantages of Painter's Algorithm

The advantages of painter's algorithm are that

1. It is based on the divide and conquer strategy. Therefore, parallel computers can be used to speed up the process.
2. It does not use extra memory buffer.

11.4 Warnock Algorithm

The Warnock algorithm also known as the area subdivision method is an image space method but it makes use of the object space method for depth comparison. It also makes use of area coherence. This method tries to get the display right instead of deciding exactly what is happening in the scene. It is a resolution dependent method. As the resolution of the display increases, the amount of computation also increases.

According to this method, the viewport area is divided into smaller areas. This division is done until every small area contains a polygon projected on viewport area. As soon as the area (pixel) becomes small, the dividing process stops.

This algorithm helps to decide which projected polygon surface overlaps a given area on the screen and is therefore a possible candidate for visibility in this area. In second part of the algorithm all those potential candidates are again tested to find out which one will be completely visible and can be displayed. The given rectangular area or samples are further subdivided if entire polygonal surface is not visible.

Notes

As the resolution of the display increases, the amount of computation also increases.

Algorithm

1. Initialize the area to be subdivided as the given viewport.
2. Sort the polygon according to the minimum depth value.
3. Create a visible polygon list.
4. Apply the following tests for the visibility of the polygon, if it satisfies the following condition:

   (a) If all the polygons forming an object on screen fall outside, then set -

   The sampling area with background color.

   (b) If a single polygon is present in the sampling area, then draw-

   The polygon with desired intensity values provided for each internal pixel.

   (c) If a single polygon is present in the probable visible list, then color-

   The entire sample area by intensity values of each pixel on the polygon.

(d) If the surrounding polygon is the closest to the sample area then draw -

The polygon along with its color value.

(e) If the area under consideration is a pixel –

Compute the z value of all the polygons in the polygon visible list.

Set the color of the polygon possessing optimum z value.

5. If no condition in step 3 is true, then subdivide the area into four equal parts.
6. Repeat steps 2 and 3 for each area.

11.5 Ray Tracing vs. Rasterization

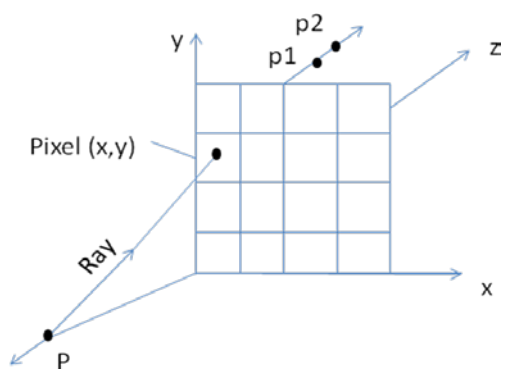
It is important to understand the differences between ray tracing and rasterization when studying hidden surfaces.

Ray Tracing

Ray tracing algorithm provides the flexibility to handle both flat and curved surfaces. This algorithm can be easily modified to provide realistic shading. It is based on the principles of light and optics. Let us assume opaque surfaces of given colors and deal directly with perspective projection transformation without applying perspective to parallel transformation.

The basic idea here is to trace light rays and determine which one arrives back at the eye or viewpoint. This involves an infinite number of light rays. Hence, we trace a ray from the viewpoint through a pixel until it reaches a surface. This identifies the first surface seen at the given pixel. Set this pixel to the color of the surface at the point where the light ray strikes it.

Figure 11.1: Ray Tracing Phenomenon

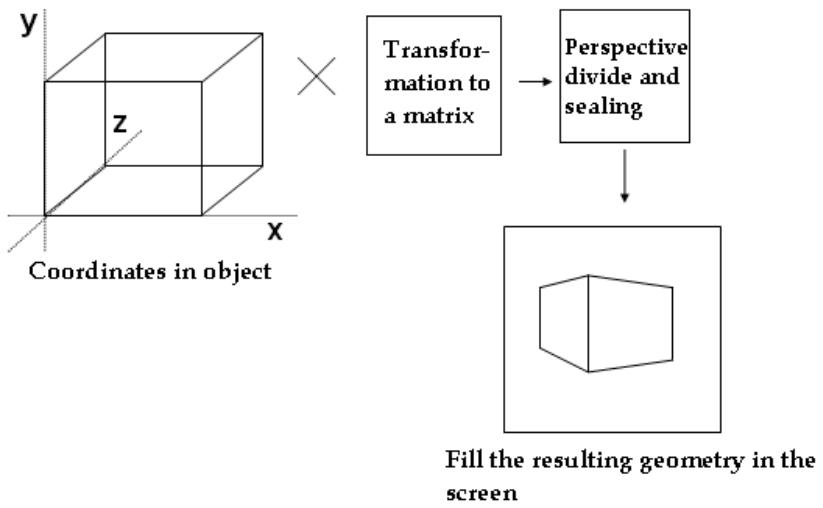


Each ray is tested for intersection with each object in the picture including the non-clipping plane. Since each ray can intersect many objects, we find the intersection point 'P' is closest to the viewpoint. Set the pixel belonging to the given ray to the color of the surface on which this point 'P' lies. This becomes the first surface intersected by the ray. Repeat this process for every pixel. There is no need to check for intersections with enclosed surface if a ray does not intersect a bounding box. The most important reason for using ray-tracing method is to create realistic renderings of pictures with the help of laws of optics for reflecting and transmitting light rays.

Rasterization

Rasterization, also known as **Scan line rendering,** is a process where the outline representation of a 3-D object is captured, which is usually a triangular mesh. The captured representation is then transformed into screen space, and projected on the screen. The resulting view is obtained in 2-D geometry as shown in figure 11.2. This technique is employed in various computer games and other interactive visualizations since these modern 3-D graphics hardware can rasterize at a fast speed.

Figure 11.2: Rasterization Effect



Now, let us discuss some of the prime differences between ray tracing and rasterization techniques.

Table 11.1 Ray tracing vs. Rasterization

Ray tracing	Rasterization
The speed of ray tracing with proper acceleration structures is O (log n) to the number of primitives in the scene. Thus, it is a little bit slower.	Rasterization is usually O (n). Thus, it is faster than Ray tracing.
Effects such as reflections, shadows, and global illumination can be easily added by simply shooting more rays in smart ways.	These effects find it much harder or even impossible to take effect appropriately through rasterization.
Usually very low level of memory coherency is achieved through ray tracing.	Extremely high level of memory coherency can be achieved through rasterization.
While ray tracing, the entire picture has to be kept in memory.	While rasterizing, the entire picture need not to be kept in memory as it can be drawn piece by piece.

Did you know? The traditional way of rendering in most of the 3-D programs and games is not ray tracing, but rather **rasterization** that takes effort to do the same thing with a much faster speed but with a little less perfection.

11.6 Determination

Determination is the process used to determine the surfaces and parts of an object's surfaces that are not visible from a definite viewpoint. The process of hidden surface determination is often referred to as hiding and such an algorithm is sometimes called as a hider. Hidden surface determination is very much essential to render an image properly so that it could not be possible to look through walls in virtual reality. Under determination, it is important to learn about the Octree method to understand the concept of hidden surfaces. Before, discussing Octree method, let us understand the meaning of Octree. Octree is also called as Octagon tree. Octrees are efficient data structures which can store three-dimensional data of any form. It provides the advantage to search and insert data at a very fast rate but the process of deleting and moving a data entry is very slow.

Octrees are just like binary trees except for having eight sub-nodes instead of two. The data is stored in a hierarchical manner which allows you to search an element at a faster rate. Every Octree has a root-node to which all the sub-nodes are anchored. A root-node has pointers to all the eight sub-nodes. A C++ structure of a root-node would look like:

```
struct octreeroot
{
  void *data;
  struct octreeroot subnodes[8];
};
```

Here, data points to some other data and the child-node points to other child-nodes. If there are no child-nodes, it points to null. The sub-node having no child-nodes is called as a leaf. Now, let us discuss about Octree method.

Octree Method

Octree method is the most practical method employed to eliminate the hidden surfaces of an image. In an Octree method, the hidden- surface elimination is accomplished by projecting Octree nodes onto the viewing surface in a front to back order. When we come across a color value in an Octree node, the pixel area in the frame buffer corresponding to this node is provided the color value only if no values have already been stored into the buffer. If an area is invalid then nothing is loaded. If a node is found to be completely obscured then it is eliminated from further processing so that there is no access to its subtrees.

To display an Octree, one should

1. Map the Octree onto an Ouadtree of visible areas by traversing Octree nodes from front to back in a recursive procedure.
2. Load the Quadtree representation for the visible surfaces into the frame buffer.

Task Search web and find out why the Octrees offer up an easier implementation and better structure than a BSP tree?

Use of Tree for Color Quantization

If there is a need for color quantization, we would require extending our node-structure by a reference counter and three color markers. Now, there is no need for the data pointer anymore as all the information has been stored directly into the Octree node structure. The structure would look like:

```
struct octreeroot
{
  unsigned long references;
  unsigned long red;
  unsigned long green;
  unsigned long blue;
  struct octreeroot childnodes[8];
};
```

Now, we will insert these colors into the tree. Whenever a leaf is encountered while moving up in the tree, we would have to increment the reference count and add the color value to the red, green, and blue counters respectively. This can help us in identifying the colors that are most important. After inserting all the colors we should start reducing them. Since, we know that only leaves can have a reference so we will count all the nodes that have a reference greater than zero. Now, we will move into the following code-fragment:

WHILE (numb of leaves>256)

search the node, where the sum of the child references is minimum and then reduce it.

ENDWHILE

The range 0 – 256 represents the range of colors from white to black, where 0 represents white and 256 represents black. Thus, we get the desired color composition at the end.

11.7 Removal

Surface of objects closer to the camera obscure the more far-off ones. Now, we are left to decide upon which surfaces are visible and which are not and remove them accordingly. This process of removing hidden surfaces is termed as removal.

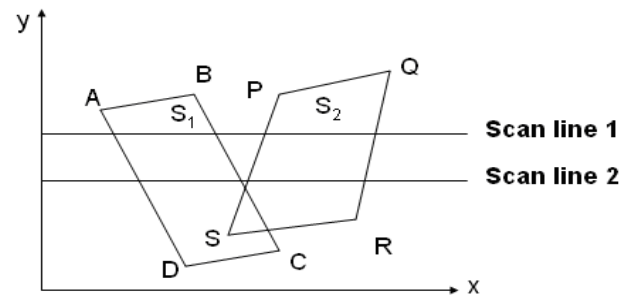
Scan Line

This is another approach of image space method. This algorithm deals with more than one surface. When each scan line is processed, the polygon surfaces intersecting that line are examined to determine which are visible. The depth calculation is then done to find which polygon is nearest to the view plane. Finally, it takes the intensity value of the nearest polygon at that position into the frame buffer.

Scan line algorithm maintains the edge list in the Edge Table (ET). The Active Edge Table (AET) contains edges which cross the current scan line. These are sorted in order of increasing value of x. The scan line algorithm also stores a flag for each surface. This flag is set on or off to indicate whether a position along a scan line is inside or outside of the surface. The processing of scan lines is done from left to right. The surface flag is turned ON at the leftmost boundary of a surface and it is turned off at the rightmost boundary.

Figure 11.3 depicts the scan line method.

Figure 11.3: Scan Line Method



As shown in the figure 11.3, for scan line 1, the active edge list contains the information for edges AD, BC, PS, and QR. Between edges AD and BC only the flag for surface S_1 is **on**. Hence, there is no depth calculations required. The intensity information for surface S_1 is entered into the frame buffer. Between edges PS and QR, only the flag for surface S_2 is **on**. The information of intensity is entered for surface S_2 into the frame buffer.

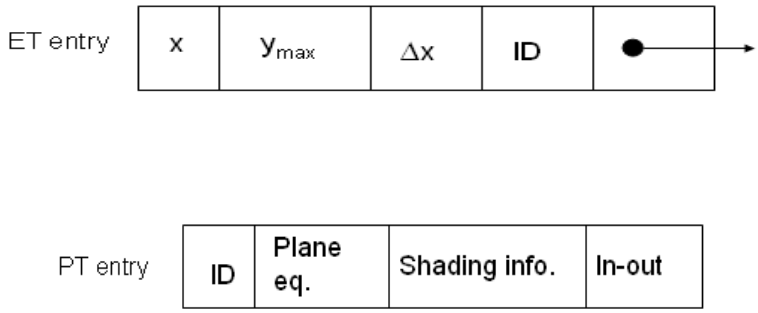
As shown in the figure 11.3, for scan line 2 the active edge list contains the information for edges AD, PS, BC, and QR. Between edges AD and PS, only the flag for surface S_1 is **on**. Between edges PS and BC, the flags for both surfaces are **on**. In this portion of scan line 2, depth calculations are required. Here we assume that the depth of surface S_2 is more than the depth of surface S_1 and hence the intensities of S_1 are loaded into the frame buffer. Between edges BC and QR of scan line 2, intensities of surface S_2 are entered into the frame buffer because during that portion only flag for surface S_2 is **on**.

To implement this algorithm along with Active Edge Table (AET) it is required to maintain a Polygon Table (PT). This PT contains at least the following information for each polygon.

1. Coefficient of the plane equation
2. Color information for the polygon
3. In-out Boolean flag

Following figure 11.4 shows the edge table (ET), polygon table (PT) for the scan line algorithm.

Figure 11.4: ET, PT for the Scan line algorithm



Back face

This algorithm is based on the inside test. Let us consider an object say a cube. Any object in computer graphics is made of polygons or surfaces. A cube is made of six different polygons.

Let us make an assumption that if a surface is in the front of the viewer then it is drawn using edge in anticlockwise pen motions whereas the surfaces that are not visible to the viewer are drawn using edge in the clockwise pen motions. A polygon has two surfaces such as the front surface and the rear surface. The front surface of a polygon is drawn using the edge in anticlockwise direction and the rear surface is drawn using the edge in clockwise direction.

This brings us to a conclusion that only those surfaces or portions of the object are visible which are drawn in anticlockwise direction. There is a very simple test for checking whether a surface is in front of the user's view or not. Take a normal vector passing through a surface and if that vector comes to the viewer's eye then the surface is visible to the viewer which otherwise is hidden. That is whether it is in the front side or in the rear side. However, the most important question that arises here is how to find out the direction of any vector.

Let us find the solution to this problem. Take two vectors A and B in a right-handed system. The dot product of these two vectors can be written as:

$$\boldsymbol{A \cdot B = |A|\,|B| \cos\theta} \qquad \text{…. (Equation 11.2)}$$

where,

Both $|A|$ and $|B|$ are length of vectors A and B respectively. Therefore they are non-negative values. Therefore, the sign of dot product is based on cos θ.

If the two vectors are in the same direction, then

$$\boldsymbol{0 \le \theta \le \prod/2}$$

If the two vectors are in the opposite direction, then

$$\boldsymbol{\prod/2 < \theta \le \prod}$$

We know that

$\cos\theta > 0$ if $0 \le \theta < \prod/2$

$\cos\theta < 0$ if $\prod/2 < \theta \le \prod$

This means,

$A \cdot B = |A|\,|B| \cos\theta > 0$ if $0 \le \theta < \prod/2$

and $A \cdot B = |A|\,|B| \cos\theta < 0$ if $\prod/2 < \theta \le \prod$

Also, if $A \cdot B > 0$ A and B are in the same direction

and if $A \cdot B < 0$ A and B are in the opposite directions

On the basis of this, if we take two vectors such as vector N which is normal to the surface in test and vector V which is the direction vector, then

If $N \cdot V > 0$ surface in test is visible

and $N \cdot V < 0$ surface is invisible …. (Equation 11.3)

If it is considered that the vector V is in the direction of depth, that is, in the direction of Z-axis then V = [0 0 V3].

Therefore,

$$V \cdot N = [0\ 0\ V3] \cdot [N1\ N2\ N3]$$

$$= V3N3 \qquad \text{… (11.4)}$$

From Equations 11.2 and 11.3, we can say that in order to check the direction of two vectors V and N, it is enough to check the sign of the Z co-ordinate of the normal vector N. If the sign of the Z co-ordinate of the normal vector is positive then the surface in test is visible, otherwise it is invisible.

Similarly for a left-handed system, the conditions will be reversed.

11.8 Summary

- The surfaces and edges which are not visible to the viewer are known as hidden surfaces and hidden edges. The process of determining which lines or surfaces of the objects are visible is known as hidden surfaces or hidden line elimination, or visible surface determination.
- When 3-D shapes are projected onto 2-D computer screen, then for every (X, Y) screen location, there is one or more 3-D surface points.
- Binary Space Partitioning (BSP) tree is an algorithm independent of the viewpoint location.
- The painter's algorithm works similarly in the manner as an artist paints his canvas with background and foreground colors.
- The Warnock algorithm also known as the area subdivision method is an image space method but it makes use of the object space method for depth comparison.
- The ray tracing algorithm provides the flexibility to handle both flat and curved surfaces. The back face algorithm is based on the inside test.
- Rasterization, often referred to as Scan line rendering, is the process that captures an outline representation of 3-D objects, transforms it into screen space, and projects the resulting 2-D geometry on the screen.
- The hidden surface determination process is often referred to as hiding, and such an algorithm is sometimes called a hider.
- Octrees provide the facility to search and insert data at a fast rate. However, the process of deleting and moving a data entry is very slow.

11.9 Keywords

Algorithm: This is a step by step procedure to perform a task.

Leaf node: The node having no child-node.

Rasterize: A process used to convert a digital image into a format suitable to be displayed on a computer monitor or printout.

Tracing: This is the exact replication made by superimposing the transparent sheet on original sheet and drawing the lines.

11.10 Self Assessment

1. State whether the following statements are true or false:
   (a) Errors are usually an Object space method that differentiates objects and parts of objects from each other to decide the visibility.
   (b) A 3-D shape is a collection of 3-D surfaces or planar surface patches in 3-D world.
   (c) BSP trees are also used to slow down ray tracing and in planning the motion of robots.
   (d) The process of hidden surface determination is often referred to as hiding.
   (e) The Warnock algorithm also known as the area subdivision method is an image space method.
   (f) Any object in computer graphics is made of polygons and surfaces.

2. Fill in the blanks:

   (a) The …… or hidden surface algorithm determines the lines, edges, surfaces that are visible or invisible to a viewer.

   (b) A surface is considered visible if it has the …….Z value.

   (c) At the end of the rendering process each scan line on screen is represented by a single entry in the S-buffer which is called as the…….

   (d) The …….. breaks up the scene into regions with the spatial relationship of these regions built into the tree.

   (e) The most important reason for using ……. method is to create realistic renderings of pictures with the help of laws of optics for reflecting and transmitting light rays.

3. Select a suitable option for every question:

   (a) Which of the following is not an image space method?

      (i) Back space

      (ii) Area Subdivision

      (iii) Z-Buffer

      (iv) Scan Line

   (b) Which of the following is a disadvantage of using Z-buffer algorithm?

      (i) It can be used easily

      (ii) It can be implemented easily in object or image space

      (iii) It can be executed quickly, even with multiple polygons

      (iv) It requires additional code to do transparent surfaces

   (c) The Warnock algorithm uses …… for depth comparison.

      (i) S-buffer

      (ii) C-buffer

      (iii) Z-buffer

      (iv) Object space method

   (d) The Scan line algorithm maintains the edge list in the……...

      (i) Edge table

      (ii) Depth tree

      (iii) Octree

      (iv) Scan line

   (e) The Warnock algorithm is also known as the……..

      (i) Painter's algorithm

      (ii) Area subdivision method

      (iii) Ray tracing algorithm

      (iv) Hiding

11.11 Review Questions

1. "A surface is considered visible if it has the minimum Z value." Comment.
2. "Binary Space Partitioning tree is an algorithm independent of the viewpoint location." Justify.
3. "The visible plane occludes the planes having higher Z-values at corresponding pixel locations." Discuss?
4. "The Warnock algorithm is a resolution dependent method. " Discuss?
5. "Hidden surface determination is very much essential to render an image properly." Give reason behind it?
6. "The most important reason for using ray-tracing method is to create realistic renderings of pictures." Comment.
7. "The depth calculation is then done to find which polygon is nearest to the view plane." Justify.
8. "Only those surfaces or portions of the object are visible which are drawn in counter-clockwise direction." Justify.
9. "The processing of scan lines is done from left to right." Explain why?
10. "A root-node has pointers to all the eight sub-nodes." Justify.
11. "If the smallest z-value for a polygon is greater than the highest z-value on the other polygon, then the former polygon is in front of the latter." Comment.
12. "If a node is found to be completely obscured it is eliminated from further processing." Discuss.

Answers: Self Assessment

1. (a) False
   (b) True
   (c) False
   (d) True
   (e) True
   (f) False
2. (a) Hidden line
   (b) Minimum
   (c) C-buffer
   (d) BSP tree
   (e) Ray-tracing
3. (a) Back space
   (b) It requires additional code to do transparent surfaces
   (c) Object space method
   (d) Edge tree
   (e) Area subdivision method

11.12 Further Readings

Books

Raghavachary S. (2005). Rendering for beginners: image synthesis using RenderMan. British Library Cataloguing in Publication Data.

Online link

zNCA&sa=X&oi=book_result&ct=result&resnum=11&ved=0CGUQ6AEwCg#v=onepage&q=computer%20graphics%20best%20books&f=false

http://books.google.co.in/books?id=0VOEjFmPB-0C&dq=computer+graphics+best+books&printsec=frontcover&source=in&hl=en&ei=93KDTb3WFYe0rAf4p-zNCA&sa=X&oi=book_result&ct=result&resnum=13&ved=0CGkQ6AEwDA#v=onepage&q=computer%20graphics%20best%20books&f=false

Unit 12: Color and Shading Model

CONTENTS

Objectives

Introduction

12.1 Light and Color

12.2 Phong Model

12.3 Interpolative Shading Methods

12.4 Summary

12.5 Keywords

12.6 Self Assessment

12.7 Review Questions

12.8 Further Readings

Objectives

After studying this unit, you will be able to:

- Discuss about light and color
- Describe the Phong model
- Analyze interpolative shading methods

Introduction

A color in a computer graphic enhances the image by clearly depicting the information and identifying the features that are unclear in black and white pictures. Color helps you to distinguish between various elements of a diagram. Color also helps you to read and understand the information in a quick and accurate manner.

Color is nothing but light. Light is made up of many colors.

Example: We can see the colors of the visual spectrum.

The visual spectrum comprises red, orange, yellow, green, blue, and violet colors. When light falls on objects, certain wavelengths of light are absorbed by the objects and rest of the wavelengths of light are reflected back to the viewer. The reflected back wavelengths are perceived as color by the viewers.

In computer graphics, colors should be specified in such a way that the colors are compatible with the hardware used and also user friendly. These two requirements can also be known as hardware and user oriented requirements respectively. In general, it is difficult to find a model that meets both the requirements.

Color can also be used to add more information into the visual message or picture, and thus convey clear information. It is necessary to examine and define the role of color in various applications to make use of color in the visualization of ideas, information, or concepts.

Various mathematical models can be used to find out the shading. Each shading model helps to determine the relation between the surface and the light source. The identification of the relation between the surface and the light source helps to create a particular shading effect.

This unit discusses about light, different color models, and how colors can be used in graphics applications. RGB-color model is also discussed which will help you to understand shading models.

The study of shading models helps to determine how an object's surface can be visualized under various lighting conditions. In addition, the Phong model and interpolative shading model is also discussed.

12.1 Light and Color

Light is a term used for a range of electromagnetic radiation which a human eye can easily detect. Color is the visual perceptual property corresponding to the types called red, green, blue and others in humans. The study of relationship between light and the color should follow an international standard for the specification and measurement of colors. The study of relationship between light and color also provides a better understanding of the widely used red, green, and blue (RGB) color model.

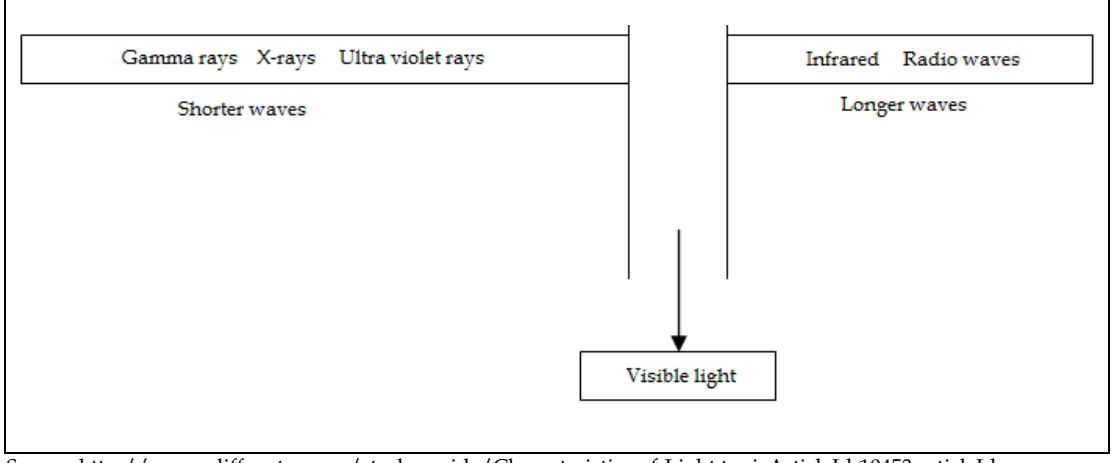
Light

Light exhibits dual nature. The reason is that, sometimes light exhibits the characteristics of a particle and sometimes it exhibits the characteristics of a wave. Perhaps, it is most precise to say that the concepts of both waves and particles can be used in simplified models that explain the nature of light. Light is such a complex phenomenon where a single model cannot explain its nature.

Basic Characteristics of Light

As we have mentioned above that the light has the characteristics of both wave and a particle, light is considered as the narrow frequency band of waves to which human eyes are sensitive. Figure 12.1 depicts spectrum of visible light.

Figure 12.1: Spectrum of Visible Light



Source: http://www.cliffsnotes.com/study_guide/Characteristics-of-Light.topicArticleId-10453,articleId-10440.html#huetinck3831c05-fig-0001

Did you know? Wavelength of light can be measured in units of nanometers (nm) where 1 nm = 10^{-9} meter or it can be measured in terms of angstroms (Å) where 1Å = 10^{-10} m).

Light travels at very high speed of about 3 × 108 m/sec. Light can be characterized into following three terms:

1. Brightness
2. Hue
3. Saturation

Brightness is similar to physical property of light called luminance. The total energy in the light can be measured using its luminance. The second character, hue can be used to differentiate a white light from a red or green light. The third character is the saturation which describes the degree of brightness.

Example: When two red lights differ in luminance and degree of brightness they might become pure/saturated red or pale/unsaturated red.

In reality, light scatters to infinite distance. Objects can be seen only when light falls on the object. In computer graphics we do not want the light to be scattered because the distance of light and its effect on objects, and so on needs to be calculated optimally to create proper shading of an object. If you have more light on a particular object then the Graphics Processor Unit (GPU) will take more time for calculation.

In 3-D computer simulations, use a model that has a final range of effect. Modern graphic engines or some custom projects should contain effective optimization methods for the calculation of light on an object. If we use all the lights with an infinite range and all the geometry together we may find it difficult to calculate the effect of each light. It may be difficult, even if the effect of light is undetectable .i.e. too small effect to an object.

Lights, scene, and shades provide a proper way to create rich and fast 3-D graphics. It is a form of art to create colorful, precise, rich, and fast 3-D graphics.

In 2-D, light is used to simulate depth and shade, and to simulate movement.

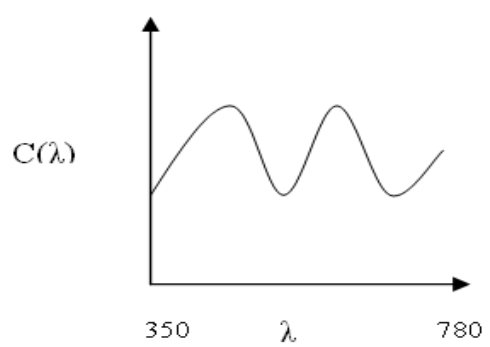
Example: Consider a ship that is moving towards a star. When you create a 2-D image for this, it will be very attractive if the ship appears as if it has been lit by the star. The simple way to do this is to represent the pixels on the side of the ship which is facing the star brighter, and the pixels on the other side should be made darker.

The impact of light in 2-D is not automated but 2-D graphics designer needs to add the impact of light on the objects.

Color

The most exciting aspect of both human perception and computer graphics is color. A visible color can be characterized by a function C(λ) that takes wavelengths of about 400 to 700 nano meter (nm) as shown in figure 12.2.

Figure 12.2: Typical Color Distribution



Source: Xiang. Z., Plstock. R. (2006). Computer Graphics. 2nd Edition. Tata McGraw Hill. Pg no. 290.

In the visible spectrum, the value λ represents the intensity of the wavelength of color light. Human visual system has three types of cones responsible for color vision. Human brains are not capable of

receiving the entire distribution C (λ) for a given color, however human brains receive only three values (the tri-stimulus values).

> Notes
>
> Tri-stimulus values are the three values that are used to define a color in a specific tri-chromaticity color model. For example the red color can be specified by the values r(1, 0, 0), g(0, 1, 0), and b(0, 0, 1).

Those three values are the responses of the three types of cones to the color. This reduction of a color to three values leads to the basic principle of tri-chromaticity theory.

> Notes
>
> If two colors have the same tri-stimulus values, then they are visually identical.

The Tri-Chromaticity Theory

The tri-chromaticity theory is a contribution of two renowned researchers by name Thomas Young and Hermann Von Helmholtz.

In this theory, let us consider **S** as a given light or color stimulus. The effect of **S** can be expressed by combining light from three primary colors (red, green, blue) in right proportions:

S= r (red) +g (green) +b (blue)

The above formula uses only red, green, and blue because they are considered as the standard choice in color matching experiments. Moreover, these three primary colors coincide with the wavelength values that cause peak response from three types of cones of human visual system. The wavelength of the three primary colors varies from 350 to 780. The three types of cones that are there in human visual system are named as β, δ, and ρ cones and these cones are most sensitive to the light in the wavelength range of 350 to 780.

An important aspect of tri-chromaticity theory is to match all visible colors and the weight values of three primary colors have to be negative.

> Example:
>
> When red, green, and blue are used as primary colors, the value of b which is the weight of the blue component may be negative. A negative value of b indicates that the effect of the given stimulus **S** can not be matched normally through the additive process. However, if **S** is mixed with some blue light then the effect of the mix can be matched by a linear combination of red and green.

Some important points about tri-chromaticity theory are that:

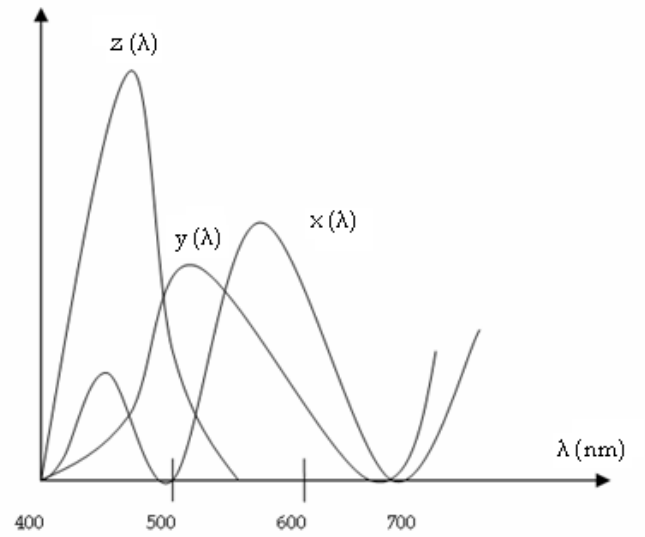
1. It follows simple model for interaction of stimulus and three different types of receptors in the human eye.
2. It can be described in terms of linear vector spaces.
3. It forms basis for colorimetry and design of many aspects of color imaging systems. This theory does not predict all visual phenomena.

Commission Internationale de l'Eclairage (CIE) XYZ Color Model

Based on the fact that no naturally occurring primaries can be used to match all visible colors, the International Commission on Illumination defined three imaginary (non-realizable) primaries such as X, Y, and Z in 1931. The three primaries X, Y, and Z were the result of an affine transformation applied to three natural primaries to ensure the expression of every single wavelength light or spectral color using a linear combination of the CIE primaries with no negative weight.

The relative wavelengths of X, Y, and Z that are required to describe each spectral color are shown in figure 12.3 in the form of three color matching functions x(λ), y(λ,), and z(λ).

Figure 12.3: Relative Wavelengths of X, Y, Z



Source: Xiang. Z., Plstock. R. (2006). Computer Graphics. 2nd Edition. Tata McGraw Hill. Pg no. 291

To match a color light of wavelength λ_0, the proper proportion is found by the line $\lambda = \lambda_0$ that intersects the curves at three positions to denote the three functions. In addition, y(λ) matches the luminous efficiency and corresponds to the human eye's response to light with constant luminance.

An arbitrary light **S** with spectral distribution P(λ) can be described by "adding together" the respective amounts of the CIE primaries that are essential to match all the spectral components of **S**. This is done with

$X = k\int_{\lambda} P(\lambda)\, x(\lambda)\, d(\lambda)$,

$Y = k\int_{\lambda} P(\lambda)\, y(\lambda)\, d(\lambda)$,

$Z = k\int_{\lambda} P(\lambda)\, z(\lambda)\, d(\lambda)$

where, k is a constant, which is based on light source. The final values of X, Y, and Z are used as weights to express S as follows:

S=X.X + Y.Y+Z.Z

CIE Chromaticity Diagram

The x, y, and z can be defined by normalizing the above weights against X+Y+Z

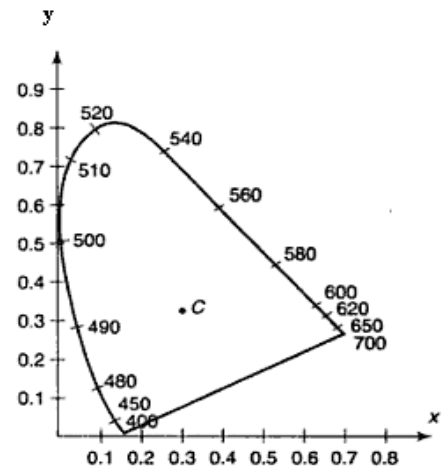
$$x = \frac{X}{X+Y+Z} \qquad y = \frac{Y}{X+Y+Z} \qquad z = \frac{Z}{X+Y+Z}$$

From the above equation; x + y + z =1, and z =1 – x - y. The variables x and y represents colors.

The CIE chromaticity graph as shown in figure 12.4 can be drawn by plotting x and y as the horizontal and vertical axes. The curved triangular figure includes all perceivable colors by ignoring luminance.

The figure 12.4 depicts CIE Chromaticity graph.

Figure 12.4: CIE Chromaticity Diagram



Source: Xiang. Z., Plstock. R. (2006). Computer Graphics. 2nd Edition. Tata McGraw Hill. Pg no. 292

When chromaticity of two colors varies then they are represented by two different points in the diagram to indicate that the two colors have different chromaticity co-ordinates.

The chromaticity co-ordinates (x, y) can be converted into a specific color in the XYZ color space and to do this an additional piece of information is required, typically Y:

$$X = \frac{x}{y}Y, \qquad Y = Y, \qquad Z = \frac{1-x-y}{y}Y$$

We can find all spectral colors along the upper border of the figure. Point C is referred as Illuminant C. The point C corresponds to a reference white that is obtained from a spectral distribution close to daylight.

The CIE chromaticity diagram can be used as a universal reference for measuring and comparing visible colors. A typical color monitor will have the following chromaticity co-ordinates: R (0.62, 0.34), G (0.29, 0.59), and B (0. l5, 0.06). As a result of these co-ordinates, a triangular region is defined within the diagram. The region is known as the color gamut of the monitor which represents all the colors which a color monitor is able to display.

Color Gamut Mapping

Let us consider that the phosphors of an RGB color monitor have the following co-ordinates: (x_r, y_r), (x_g, y_g), and (x_b, y_b). We will now introduce auxiliary variables as shown below.

$C_r = X_r + Y_r + Z_r$

$C_g = X_g + Y_g + Z_g$

$C_b = X_b + Y_b + Z_b$

where, (X_r, Y_r, Z_r), (X_g, Y_g, Z_b), and (X_b, Y_b, Z_b) represent the respective XYZ co-ordinates of the red, green, and blue colors the monitor can display. We have

$X_r = x_r C_r$, $Y = y_r C_r$, $Z_r = z_r C_r$ where $z_r = 1 - x_r - y_r$

$X_g = x_g C_g$, $Y_g = y_g C_g$, $Z_g = z_g C_g$ where $z_g = 1 - x_g - y_g$

$X_b = X_b C_b$, $Y_b = y_b C_b$, $Z_b = Z_b C_b$ where $Z_b = 1 - X_b - y_b$

Thus, the XYZ co-ordinates of a combined RGB color can be represented in terms of a transformation matrix M:

$$[X\ Y\ Z] = [R\ G\ B] \begin{pmatrix} X_r & Y_r & Z_r \\ X_g & Y_g & Z_g \\ X_b & Y_b & Z_b \end{pmatrix} = [R\ G\ B] \begin{pmatrix} X_r & Y_r & Z_r \\ X_g & Y_g & Z_b \\ X_b & Y_b & Z_b \end{pmatrix} \begin{pmatrix} C_r & 0 & 0 \\ 0 & C_g & 0 \\ 0 & 0 & C_b \end{pmatrix}$$

C_r, C_g, and C_b can be identified in two different ways. The first way is to find the luminance levels Y_r, Y_g, and Y_b of the red, green, and blue colors at their maximum intensity/brightness respectively with the help of a photometer. Then C_r, C_g, and C_b becomes as follows:

$C_r = Y_r/y_r$ $\quad C_g = Y_g/y_g$ $\quad C_b = Y_b/y_b$

The second way is to measure the XYZ co-ordinates (X_w, Y_w, Z_w) of the monitor's white color (R = G= B = l). Using these we have

$$[X_w\ \ Y_w\ \ Z_w] = [C_r\ \ C_g\ \ C_b] \begin{pmatrix} x_r & y_r & z_r \\ x_g & y_g & z_b \\ x_b & y_b & z_b \end{pmatrix}$$

Now C_r, C_g, and C_b, can be solved.

If M_1 is the transformation for monitor 1, and M_2 is the transformation for monitor 2, then a color (R_1, G_1, B_1) on monitor 1 can be matched by a corresponding color (R_2, G_2, B_2) on monitor 2:

$$[R_2\ G_2\ B_2] = [R_1\ G_1\ B_1]\ \ M_1 M_2^{-1}$$

with the condition that (R_2, G_2, B_2) is within the color gamut of monitor 2.

CMYK Color Model

CMYK color model is a subtractive color model that uses cyan, magenta, yellow, and black colors. Each of these colors comprises ink that can be measured with a percent from 0 to 100. A value of 100 percent signifies that the color can be applied at full saturation for any image.

In CMYK color model, many new colors can be formed by combining cyan, magenta, yellow, and black colors. When the colors cyan, magenta, yellow, and black are combined together and if each color has a saturation of 100% then the resulting color should be black. When the saturation percent of each color is 0 then it will result in pure white color.

CMYK color model is used to produce hardcopies of images. CMYK model can be used to generate millions of colors but as a result of the limitations of printing inks and the printing process, we can only produce few thousands of colors in print. Even though only few thousands of colors can be generated, computers can display millions of CMYK colors.

RGB Color Model

Now, we will discuss how color can be handled in a graphics system. In a graphics system, a color can be handled using two approaches. The two approaches are:

1. Indexed color model
2. RGB color model

When a computer graphics system uses 8 bits of memory to store each pixel of an image then the most effective way of generating the widest range of colors is to use indexed color model.

In the indexed color model, the computer graphics designer will have control over the method by which colors are loaded into a color lookup table. The control over color loading can be achieved by specifying

a palette. A palette can be used to map color index values into RGB values for the destination object. When the contents of the destination object are submitted to the physical device then the RGB values from the palette are either:

1. Passed directly to the physical device (that is by calling Draw() function).
2. Loaded into the lookup table of the physical device.

Notes Lookup table of indexed color model is used to store and index the colors of an image.

The indexed color model requires less memory and has limited color available on displays. Therefore, it was easier to support in hardware. However, it has become a norm to use RGB color model in modern systems.

The RGB color model is an additive color model in which the primary colors red, green, and blue light are combined together in different ways to reproduce a wide range of colors. The name RGB color model has originated from starting letters of the three additive primary colors, red, green, and blue.

Did you know? When red color is mixed with green color we will get yellow color. Same way when all the three primary colors are added, it produces white color.

The basic principle of RGB color model is that, a display requires only three primary colors to generate the three tri-stimulus values essential for a human visual system. Various colors can be produced by varying the intensity of each primary color.

Did you know? Cathode Ray Tube (CRT) is a technology used in modern computer monitors to produce a perceived color by adding primary colors.

The CRT uses additive color model. With the usage of primary colors, a CRT can add light to an initially black display and produce the desired color.

The three primary color additive system includes separate buffers for three primary colored images. Corresponding to memory location, each pixel maintains individual buffer for red, green, and blue components. A typical system may include an array of 1280 x1024 pixels and each pixel might consist of 3 bytes with one byte for each of red, green, and blue.

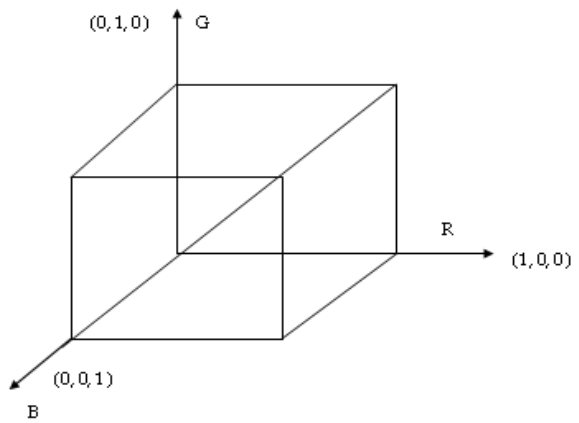
A programmer would like to specify any color that a frame buffer can store.

Example: If there is any image that takes 24-bits then there will be 2^{24} possible colors, sometimes referred to as 16M colors, where M =l024.

Other systems may have maximum of 12 bits per color or minimum of 4 bits per color. Therefore, it is a good practice to specify a color independently of the number of bits in the frame buffer and to let the drivers and hardware match the specification as closely as possible to the available display. A color can be specified by using a color cube as shown in figure 12.5, where the color components are specified as numbers between 0.0 and l.0. A value of l.0 represents the maximum (or saturated value) of the respective primary color and 0.0 denotes a zero value of that primary color.

The figure 12.5 depicts a color cube.

Figure 12.5: Color Cube



Now let us discuss how a color cube can be used to specify any color. Consider the example of Open Graphics Library (OpenGL). OpenGL is an Application Programming Interface (API) that helps in defining 2-D and 3-D computer graphics.

OpenGL uses the color cube as shown below:

To draw-in red, OpenGL issues the following function call:

glCo1or3f (l.0, 0.0, 0.0);

The execution of glColor3f (1.0, 0.0, 0.0) function will set the current image color to red. The reason is that the color is part of the state where red color remains the same until color is changed. The 3f is used to tell a program that we are using a tri-color (RGB) model.

Task

Make a list of colors that can be generated by mixing blue and green, and blue and red.

Hue, Saturation, and Intensity (HIS) Model

Similar to light, even the color can be differentiated by three quantities such as hue, saturation, and intensity. The Hue, Saturation, and Intensity (HSI) color model is an effective tool for developing image processing algorithms depending on color descriptions that are natural and intuitive to the users who use those algorithms.

Conversion from RGB to HIS can be done using the following equations:

H=arctan (L/M)

$S= (L^2+M^2)^{1/2}$

$I=N\sqrt{3}$

Where

$$[L\ M\ N]=[R\ G\ B]\begin{pmatrix} 2\sqrt{6} & 0 & 1/\sqrt{3} \\ -1/\sqrt{6} & 1/\sqrt{2} & 1/\sqrt{3} \\ -1/\sqrt{6} & -1/\sqrt{2} & 1/\sqrt{3} \end{pmatrix}$$

Conversion from HIS to RGB is as follows:

L=S sin H,

M=S cos H,

N=1/√3,

Where,

$$[L\ M\ N]=[R\ G\ B]\begin{pmatrix} 2\sqrt{6} & -1/\sqrt{6} & -1/\sqrt{6} \\ 0 & 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} \end{pmatrix}$$

National Television System Committee (NTSC) YIQ Color Model

NTSC YIQ Color model is used to represent the colors in systems that use color screens where Y stands for the luminosity, which is also the black and white signal, I stands for red minus the luminosity, and Q stands for blue minus the luminosity.

NTSC YIQ model is considered as a linear transformation of the RGB color model. The Y component of YIQ model is purposely made same as the Y component of the CIE XYZ color model. Since, Y is meant to carry luminance information, black and white television systems can use it to display color images as gray scale pictures. The formula for NTSC YIQ model is as follows:

$$[Y\ I\ Q]=[R\ G\ B]\begin{pmatrix} 0.299 & 0.596 & 0.212 \\ 0.587 & -0.275 & -0.523 \\ 0.114 & -0.321 & 0.311 \end{pmatrix}$$

The elements in the transformation matrix are obtained through the standard NTSC RGB phosphors whose chromaticity co-ordinates are R (0.67, 0.33), G (0.21, 0.71), and B (0.14, 0.08). In YIQ model, it is assumed that the white point is at x_w=0.31, y_w=0.316, and Y_w=1.0.

12.2 Phong Model

Phong model is generally used to imitate the reflection of light from object surfaces to the viewer's eye. Phong model is known as an observation approach because it relies on some basic principle of physics, and is mainly based on our observation of the phenomenon. It is also called as a local illumination model because the Phong model gives much importance to the direct impact of the light coming from the light source. On the other hand, a global illumination model may include such secondary effects as light passing through transparent/translucent material and light passing from one object to another.

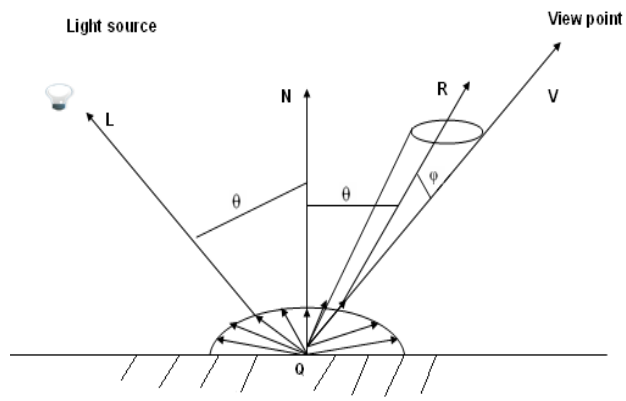
Now, let us consider a point light source as shown in figure 12.6, which is an idealized light with a single point in space as the source of energy. Our eye is looking at point Q that is present on the surface of an object. The point Q can appear in different colors depending on the type of reflection.

A light can reflect in two extreme cases. They are:

1. Diffuse reflection
2. Specular reflection

In diffuse reflection, light energy from the light source gets reflected uniformly in all directions. In figure 12.6, the small arrows that form a hemisphere represents diffuse reflection. The energy level of the reflection can be expressed as a function of the incident angle θ between L and surface normal N. Higher reflection can be achieved through smaller incident angle. The incident angle can be made smaller by having the reflection proportional to cos(θ).

Figure 12.6: Phong Model



Source: Xiang. Z., Plstock. R. (2006). Computer Graphics. 2nd Edition. Tata McGraw Hill. Pg no. 295

The specular reflection occurs when the surface has the characteristic of a shiny or mirror like surface. If the surface in figure 12.6 is a perfect mirror then the light rays from the point light source would be reflected exactly in one direction. A perfect mirror does not exist in reality. So, the Phong model distributes the reflected energy across a small cone shaped space centered around R. While doing so, the Phong model distributes more reflection along the direction of R and decreases the reflection energy quickly as φ increases. The mathematical means for distributing reflected light across a small cone shaped space centered around R is $\cos^k(\varphi)$, where k can be used to vary the degree of shininess.

Did you know? The parameter k in $\cos^k(\varphi)$ is 1 for a dull surface and 100 for a shiny surface.

Furthermore, the complex inter-object reflection should be considered in some way because many surfaces we see do not reflect directly by the light source. They are reflected by light that is passing through the environment. For these kinds of objects there is a need to introduce a directionless ambient light which illuminates all surfaces in a particular area and gets reflected uniformly in all directions by each surface.

Thus, according to Phong model, object surfaces might generate a blend of ambient light reflection and light source dependent diffuse/specular reflection. Mathematically the total reflected energy intensity ***I*** can be written as follows:

$$I = Iaka + Ip\,(kd\cos(\theta) + ks\cos^k(\varphi))$$

where, I_a -- the intensity of the ambient light

I_p -- the intensity of the point light

$0 <= k_a, k_d, k_s <= 1.0$ are reflection co-efficients that represent the ability of a surface to reflect ambient light in order to create diffuse reflection and to create specular reflection respectively.

If L, N, R, and V are all unit vectors then L * N = cos(θ) and R * V= cos (φ). We can re-write the above formula as

I=Iaka+Ip (kd L * N + ks(R * V)^k)

Notes

The value of R can be calculated with the term L * N.

The above mentioned formulae can be generally used along with the RGB color model. Thus, the intensity of a light can be written in the form of an RGB color vector which can be represented as I = (I_r, I_g, I_b). The reflection co-efficients can also be expressed as 3-D vectors.

Example: The reflection coefficient k_d = (0.7, 0.7, 0.3) defines a surface that appears yellowish when illuminated with white light.

The ambient reflection coefficient I_c, can simply be similar to the reflection co-efficient k_d. The three components of I_c are often made equal since the color of the reflection of a light source is usually the same as the color of the light source itself. When white light is used the formula for a single point source can be written as

Ir=Iakar + Ip (kdr L * N + ks (R *V)^k)

$$I_g = I_a k_{ag} + I_p (k_{dg} L * N + k_s (R * V)^k)$$

$$Ib, = I_a k_{ab} + I_p, (k_{dp} L * N + k_s (R * V)^k)$$

12.3 Interpolative Shading Methods

Interpolation is nothing but the identification of new values that exist between known values determined for the vertices of a polygon. Interpolative shading methods use interpolation at the vertices of the faces of polygons.

Surface color can be calculated using an illumination model such as the usage of a formula at any point of the surface. It can be more costly and it is even unnecessary when the image is used to preview the scene. To avoid this problem, it is suggested to apply the formula at full scale only at selected surface points. Then other surface points can be shaded with the help of any other techniques such as color interpolation and surface normal interpolation.

In the following sub sections we will discuss about some interpolative shading methods that are used to shade a polygon.

Constant Shading

Constant shading is the easiest shading model for a polygon.

Did you know? Faceted shading or flat shading is the other name for constant shading.

Constant shading method applies an illumination model once to produce a single intensity value. The single intensity value can be used to shade the polygon completely and to regenerate the polygon's shade. This method is useful only when the following assumptions are true.

1. The light source should be at infinite distance so that L * N becomes constant through out the polygon face.
2. The view point should be at infinite distance so that N * V becomes constant throughout the polygon face.
3. The polygon should signify the actual surface being considered and it should not be an approximation to a curved surface.

Among the above mentioned assumptions, if any of the first two assumptions goes wrong and if we carry on with constant shading then we need to determine a single value for each L and V with the help of some other methods.

Some of the advantages of constant shading method are that:

1. It is fast because it uses only one shading computation to shade a polygon.
2. A polygon can be filled with a single color.

Some of the disadvantages of constant shading method are that it may produce inaccurate results when it is used to shade polygons that share a common boundary and each shaded differently.

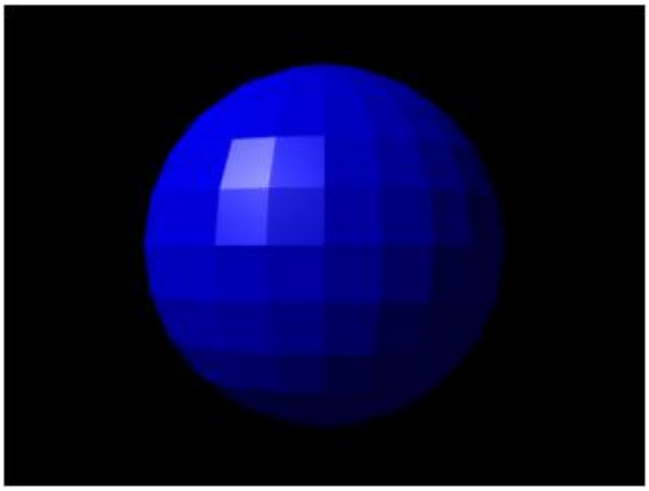
Example: The values for L and V may be determined for the center of the polygon or for the polygon's first vertex.

Notes When constant shading is used to shade a polygon, it results in uniformly shaded polygon surface.

Constant shading can produce good results for dull polyhedrons reflected by light sources that are located far away from the polygon. The main disadvantage of this method is that there may be wrong lines appearing in between adjacent polygons that approximate a curved surface. Suppose if a selected surface point is at the point of the strongest specular reflection of the light source then constant shading may modify the color of the corresponding polygon.

Constant shading method uses a single color (as shown in figure 12.7) to shade a given polygon.

Figure 12.7: Constant Shading of an Image



Source: http://images.yourdictionary.com/images/computer/_SHADING.GIF

Gouraud Shading

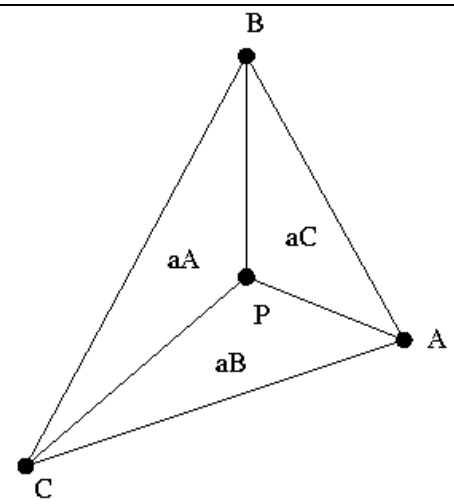
Gouraud shading is also known as intensity interpolation and it is used to display smooth-shaded polygons by defining the RGB color combination of each polygon vertex. The first thing that the Gouraud shading does is to estimate the RGB values between the vertical vertices along each edge. Using this method, we can obtain the RGB value for the left and right edges of each scan line. Then,

each row of pixels can be displayed by horizontally interpolating the RGB values between left and right edges of that row. This often results in a remarkably smooth-shaded polygon.

Gouraud shading calculates the illumination formula at the vertices of the polygon mesh. We then estimate the obtained color components to get complete shading of the polygon.

The figure 12.8 depicts the calculation of the illumination formula at each vertices of the polygon and then the interpolation of the obtained color components to get complete shading of the polygon.

Figure 12.8: Calculation of Illumination Formula using Gouraud Shading



When the illumination formula is calculated at the vertices A, B, and C, interpolation can be done as follows:

aA=PBC/ABC

aB=APC/ABC

aC=ABP/ABC

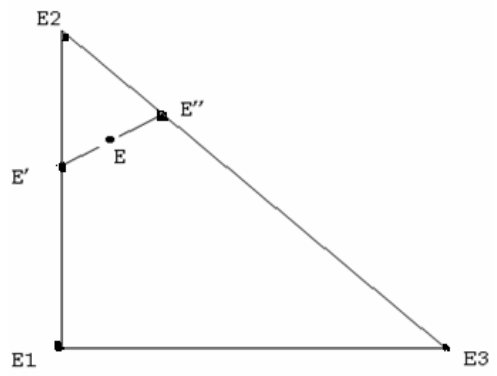
aA+aB+aC=1

P=aAA+aBB+aCC

Example: Bilinear interpolation can be used to find the color of point E inside a triangle whose vertices are E1, E2, and E3 as shown in figure 12.9. The scan line containing E intersects the two edges at points E' and E". We estimate the color of E1, and the color of E2 to get the color of P'. We then estimate the color of E2 and the color of E3, to get the color of P". Finally, we estimate the color of E' and the color of E" to get the color of E.

The usage of bilinear interpolation is shown in figure 12.9.

Figure 12.9: Usage of Bilinear Interpolation

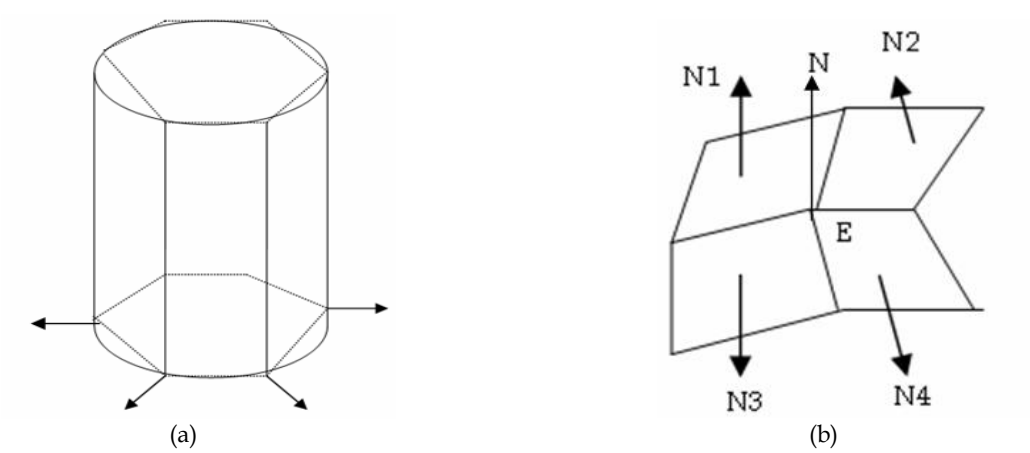


Source: Xiang. Z., Plstock. R. (2006). Computer Graphics. 2nd Edition. Tata McGraw Hill. Pg no. 298

A curved surface can be shaded with the help of polygons. The normal vectors at the polygon's vertices can be identified in two methods to obtain smooth-looking transition between adjacent polygons. The first method is to determine the exact surface normal at each vertex location with the help of underlying curved surface.

Example: The vertices of the polygon mesh in figure 12.10 are on the surface of an underlying cylinder. The normal vector N at a point on the underlying cylinder is perpendicular to the axis of the cylinder.

Figure 12.10: Finding Normals for Curved Surfaces



Source: Xiang. Z., Plstock. R. (2006). Computer Graphics. 2nd Edition. Tata McGraw Hill. Pg no. 298

The second method for determining the normal vectors involves calculating the average of normal vectors of adjacent polygons' vertices.

Example: The vertices of the polygon mesh in figure 12.10 (a) are on the surface of an underlying cylinder. The normal vector N at a point on the underlying cylinder is perpendicular to the axis of the cylinder.

To determine normal vector N at vertex E in figure 12.10 (b), we use the average of the normal vectors of the polygons that meet at E:

N=N1 + N2 + N3 + N4

The value of N can then be normalized by dividing it by |N|.

Common normal vectors at the vertices of a polygon mesh that are used to shade a curved surface result in color values that are shared by adjacent polygons and also eliminate abrupt modifications in shading characteristics across polygon edges. Even when the polygon mesh is relatively coarse, Gouraud shading is very useful in eliminating abrupt modifications.

Task Consider a hexagonal polygon, draw a bilinear interpolation diagram and analyze how you can find out normals at each vertex.

Some of the advantages of Gouraud shading are:

1. Unlike a constant shaded polygon, Gouraud shading allows to use different shade for each vertex of a polygon.
2. Gouraud shading provides much better image than constant shading.
3. Gouraud shading is not too computationally expensive as it calculates only one illumination formula for each vertex.

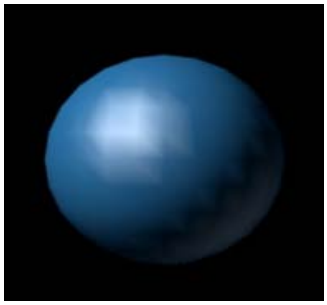
Some of the disadvantages of Gouraud shading are:

Gouraud shading eliminates folds that you may want to retain.

Caution In some cases, it is not appropriate to use Gouraud shading for polygons with three vertices. Polygons may appear quite different when they have more than three vertices with different shades for each.

The following figure 12.11 depicts the Gouraud shading of an image.

Figure 12.11: Gouraud Shading of an Image

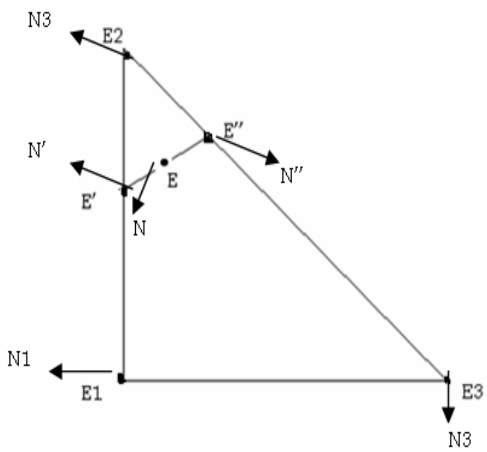


Phong Shading

Phong shading is also called as **normal vector interpolation shading**. Phong shading can be used to interpolate vector N that is normal to the surface, instead of interpolating the color values.

Example: In figure 12.12, to determine the normal vector N at point E in a triangle, we first interpolate N1, and N2, to get N', then N'' can be calculated by interpolating N2 and N3. Finally, normal to the surface can be calculated through the interpolation of N' and N".

Figure 12.12: Finding Normals for Phong Shading



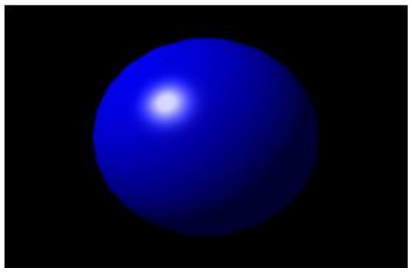
Source: Xiang. Z., Plstock. R. (2006). Computer Graphics. 2nd Edition. Tata McGraw Hill. Pg no. 299

Phong shading method consumes relatively more time since the illumination model is evaluated at every point of interest using the interpolated normal vectors. However, Phong shading is helpful when we need to deal with specular highlights.

Phong shading results in specular highlights which are much less dependent on the underlying polygons. The drawback of Phong shading is that it requires much calculation as it determines the interpolation and intensity of each pixel.

Consider the following figure 12.13 which depicts the Phong shading of an image.

Figure 12.13: Phong Shading of an Image



Source: http://images.yourdictionary.com/images/computer/_SHADING.GIF

12.4 Summary

- Usage of color in all computer graphics applications helps to clearly depict the images, the information presented, and also to clearly distinguish the features of images.
- The study of shading models helps to analyze object's surface under different lighting conditions.
- Light has dual nature and color is a mixture of all lights. Light is characterized by brightness, hue, and saturation.
- Various colors can be obtained by varying the wavelengths of primary colors.
- In computer graphics, color can be handled using indexed color model or RGB color model.
- RGB color model is an additive color model. In RGB color model, a color cube can be used to specify the colors.
- NTSC YIQ color model is also known as linear transformation of RGB color model and it can be used to represent colors in color screens.
- According to Phong model, object surfaces might produce a mix of ambient-light reflection and light-source-dependent diffuse/specular reflection.
- Constant shading of polygon produces uniformly shaded polygon surface.
- Gouraud shading calculates illumination formula for each vertices of a polygon.
- Phong shading involves more calculations.

12.5 Keywords

Additive Color Model: An additive color model uses light directly coming from a source or some sort of thing that serves as light source. The additive color model usually generates new colors by using red, green and blue light.

API: An API is a defined set of rules and regulations that a software program needs to follow to access and utilize the services and resources of other software programs that implements API.

Graphics Processor Unit (GPU): A GPU is a specialized processor that is used to process graphics in a quick manner than a central processor.

Luminance: Luminance is nothing but the brightness of a color. Luminance is considered as the total amount of light emitting from a given surface area.

OpenGL: It is an application programming interface that helps to define 2-Dimensional and 3-Dimensional computer graphics.

Tri-stimulus: It is considered as the total effect of three standard stimuli needed to match a given sample of light.

12.6 Self Assessment

1. State whether the following statements are true or false:
   (a) When the chromaticity of two colors varies, they are represented by two different co-ordinates.
   (b) The CIE chromacy diagram can be used as a universal reference for measuring and comparing visible colors.
   (c) RGB color model is a subtractive color model.
   (d) Light has dual nature.
   (e) The total energy in the light can be measured using its hue.
   (f) Even when the polygon mesh is relatively coarse, Gouraud shading is very useful in eliminating abrupt modifications.

2. Fill in the blanks:
   (a) The hue of light can be used to differentiate a ..... from a red or green light.
   (b) The tri-chromaticity theory is a contribution of …… and ……. researchers.
   (c) Brightness is similar to physical property of light called ……...
   (d) The …….. model requires less memory and has limited color available on displays.
   (e) In …… reflection, light energy from the light source gets reflected uniformly in all directions.
3. Select a suitable option for every question:
   (a) In NTSC YIQ color model, what does Y indicate?
      (i) Luminosity
      (ii) Intensity
      (iii) Saturation
      (iv) Hue
   (b) According to Phong model, what is the mathematical equation for the total reflected energy, Intensity I?
      (i) I=Iaka+Ip (kscos(θ) + kd cosk(φ))
      (ii) I=Iaka+Ip (kd cos(φ) + kscosk(θ))
      (iii) I=Iaka+Ip (kd cos(θ) + kscosk(φ))
      (iv) I=Iaks+Ip (kd cos(θ) + kacosk(φ))
   (c) What is the other name for constant shading?
      (i) Faceted shading
      (ii) Gouraud shading
      (iii) Phong shading
      (iv) Intensity interpolation
   (d) What is the correct expression of tri-stimulus generalization theory?
      (i) S=r+g+b
      (ii) r=S+g+b
      (iii) b=S-r+g
      (iv) g=S-r-b
   (e) In CIE XYZ color model, one of the primaries can be calculated as $Y=\int\lambda k\, P(\lambda)\, y(\lambda)\, d(\lambda)$. What is k?
      (i) Variable based on wavelength of the light
      (ii) Constant based on yellow color of the light
      (iii) Variable based on light source
      (iv) Constant based on light source

12.7 Review Questions

1. "Light exhibits dual nature". Justify this sentence.
2. According to tri-chromaticity theory of light, S= r (red) +g (green) +b (blue). Explain.
3. "An arbitrary light S with spectral distribution P(λ) can be described by adding together the respective amounts of the CIE primaries". Explain with equations.
4. "A color cube can be used to specify the colors". Explain.
5. RGB color values can be converted to HIS values. Explain the conversion method with the equations.
6. "NTSC YIQ model is considered as a linear transformation of the RGB color model." Explain.
7. "Constant shading is the easiest shading model for a polygon." Explain.
8. "Constant shading is useful only when some assumptions are true". Explain those assumptions.
9. "Gouraud shading provides much better image than constant shading." Explain.
10. "Phong shading method consumes relatively more time." Explain.
11. "The chromaticity co-ordinates (x, y) can be converted into a specific color in the XYZ color space." Briefly explain with equations.
12. "The XYZ co-ordinates of a composite RGB color can be expressed in terms of a transformation matrix." Explain.

Answers: Self Assessment

1. (a) True (b) True (c) False
   (d) True (e) False (f) True
2. (a) White light (b) Thomas Young, Hermann Von Helmholtz
   (c) Luminance (d) Indexed-color (e) Diffuse
3. (a) Luminosity (b) I=Iaka+Ip (kd cos(θ) + kscosk(φ)) (c) Faceted shading
   (d) S=r+g+b (e) Constant based on light source

12.8 Further Readings

Books

Gutierrez,M., Vexo,F., & Thalmann, D.(2008). Stepping into Virtual Reality. Springer-Verlag London Limited.
Angel, E. Interactive Computer Graphics: A Top-Down Approach Using Open-GL. 5th Edition. Pearson Education.
Shirley, P., & Marchner, S. (2009). Fundamentals of Computer Graphics. 3rd Edition. A.K. Peters, Ltd.
Xiang. Z., Plstock. R. (2006). Computer Graphics. 2nd Edition. Tata McGraw Hill.

Online link

http://portal.acm.org/citation.cfm?id=806793
http://www-graphics.stanford.edu/courses/cs448b-02-spring/04cdrom.pdf
http://www.articlesbase.com/science-articles/characteristics-of-light-3289311.html
http://www.csit.parkland.edu/~dbock/Portfolio/Content/Lights.html

Unit 13: Advanced Computer Graphics

CONTENTS

Objectives

Introduction

13.1 Texturing

13.2 Ray Tracing

13.3 Morphing

13.4 Summary

13.5 Keywords

13.6 Self Assessment

13.7 Review Questions

13.8 Further Readings

Objectives

After studying this unit, you will be able to:

- Define texturing
- Analyze ray tracing
- Visualize morphing

Introduction

The advent of computers led to the emergence of computer graphics. Initially, computer graphics were displayed on hardcopy plotters and cathode ray tube (CRT) screens. However; today, computer graphics is more interactive in nature when compared to the past.

Computer graphics is also defined as a process of creating and manipulating images with the use of computers. To work with computer graphics, it would be advantageous if you have knowledge and understanding of specific hardware, file formats, and Application Programming Interfaces (APIs).

Computer graphics involves the procedure of creation and storage of manipulated models of images. These manipulated models are generated from diverse fields of physics, mathematics, engineering and conceptual architectural structures.

In advanced computer graphics, the introduction of raster graphics as a built in feature in the personal computers popularized the use of bitmap graphics for user computer interactions. A bitmap is represented by 'zeros' and 'ones' of a rectangular array of points known as pixel or pels.

Did you know? Pixel is the short form of picture elements.

13.1 Texturing

Texture is the way how things look. One can see many types of textures of various things around them. Textures can be rough, smooth, soft, and uneven. A cotton ball has a soft texture while sand has a rough texture. Generally, textures are used by artists. They make smooth and flat pictures using water paints. By using thick layers of paint, an uneven texture can be obtained.

Did you know? Uneven or bump mapping is a technique in computer graphics developed by Jim Blinn.

Texture can be defined as the property that is held by the surface of any object and whose external surface can be felt by the sense of touch. Sometimes, texture is also termed as a pattern where the individual elements are non-distinguishable.

In music, you can find a musical texture. In painting, texture is supposed to be the feeling of an individual about the painting and the application method employed to paint the surface.

In computer graphics, a texture is a bitmap image used to apply a design onto the surface of a 3-D computer model.

The texturing method is used to handle the variations of reflectance and involves storing of reflectance as a pixel image or function and mapping reflectance onto the required surface. This image or function is known as **texture map** and the process is called **texture mapping**. Therefore, texture is the image used to modify the appearance of an object.

The flight simulators are the famous systems known for performing texturing at interactive rates. Texturing can also be simulated with the help of many small triangles with interpolated shading effects between the vertices.

Notes

Flight simulation is an artificial re-creation of a flight with the various aspects of the flight environment. The different equations governing how aircrafts fly, how aircrafts react to controls of the flight system and the external environment like turbulence, cloud and air density are taken into consideration.

Texture can either be two-dimensional or three-dimensional and is distinguished by visual and physical textures.

1. ***Physical Texture*:** Physical texture is known as actual texture or tactile texture. It depicts the variations on a particular surface. Physical texture can be felt by touch. Fur, wood, grains, statues, sand, glass, and so on can be included in this type of texture.

The following figure 13.1 depicts a typical texture image of a physical texture.

Figure 13.1: Physical Texture

Source: http://upload.wikimedia.org/wikipedia/commons/3/32/GreenHead01-AltesMuseum-Berlin.png

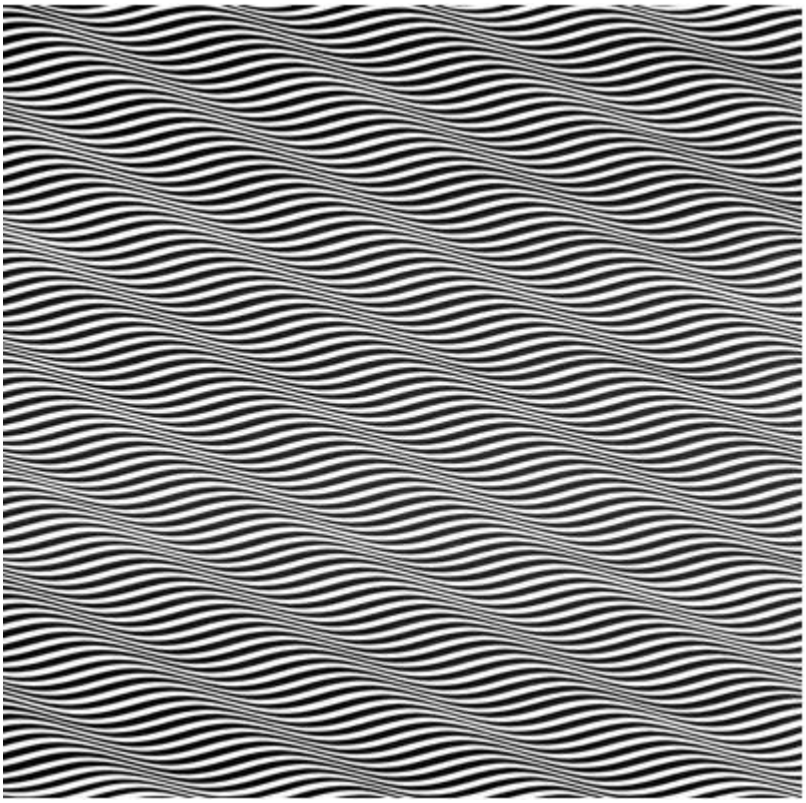
Example: Rough surfaces can appear visually active while smooth surfaces can be visually restful due to their smooth texture.

It is important to know that light is an important factor as it can affect the way in which surfaces are viewed. When a strong light is directed on a smooth surface, the readability of a drawing or a text is affected. However, the same light can create strong contrasts on a high textured surface.

2. ***Visual Texture*:** Visual texture is the perception of having a physical texture. Every material has its own visual texture which needs to be considered before creating a composition. Textures are created by repetitive use of shapes and lines.

The following figure 13.2 depicts a typical texture image of visual texture.

Figure 13.2: Visual Texture



Source: http://upload.wikimedia.org/wikipedia/en/6/67/Riley%2C_Cataract_3.jpg

Example: Photography, drawing, and paintings use visual textures.

Now let us understand the different types of texture mapping. There are three types of texturing, they are:

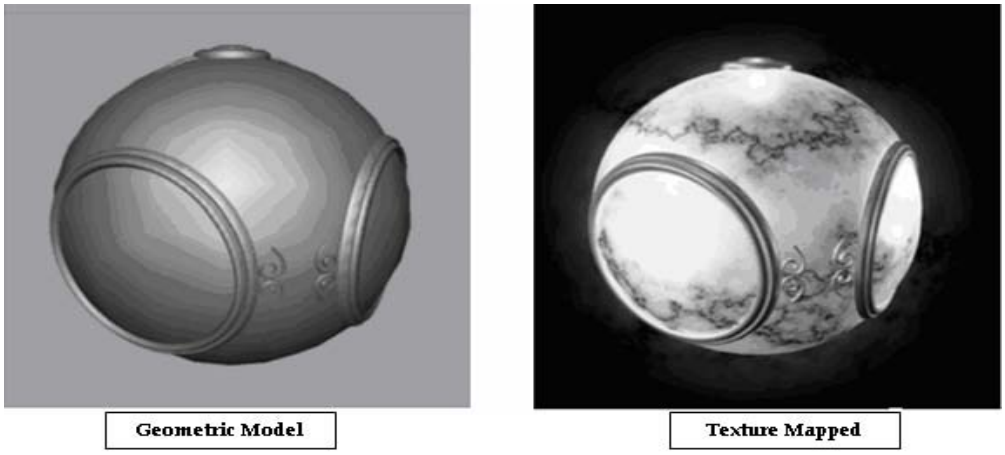
1. ***Texture Mapping:*** Texture mapping makes use of photographical or procedural generated textures. These procedural generated patterns are made to appear very natural by subjecting them to noise or random variation. The texture mapping method involves scanning or digitizing the picture into another space, 2-D texture space. The sections of these spaces are mapped onto the surfaces which are depicted by maintaining continuity. The 2-D co-ordinates use the exact object depiction method to map onto the polygons or into the 2-D surfaces.

   Realistic results are obtained by texture mapping the digitized photograph similar to wallpapering of a cylinder. To view the pixel structures, the texture is enlarged for a clear depiction. These pixels from the texture space are mapped on to the surfaces which are similar to sections of faces. As there is no similarity between the pixels of texture spaces and pixels of image spaces, these pixels tend to overlap each other.

   To calculate the average of the pixels in the texture space and the average of the pixels occupied by each of the pixel in image space is a tedious process. Texture mapping finds its application in animation and virtual reality systems as it produces realistic images and scenes that have less rendering time.

The following figure 13.3 depicts the view of a texture mapping a geometric model.

Figure 13.3: Texture Mapping



Source: http://www.cs.kent.edu/~farrell/cg05/lectures/cg22.pdf

2. ***Environmental or Reflection Mapping*:** Environmental or reflection mapping uses a picture from the environment in order to texture the maps. It also enables you to simulate high specific surfaces.

Figure 13.4: Environment Mapping

Source: http://www.cs.kent.edu/~farrell/cg05/lectures/cg22.pdf

Environmental mapping is a technique used to approximate the appearance of reflective surfaces by a pre-computed texture image. There are several methods used to store the surrounding environment. They are:

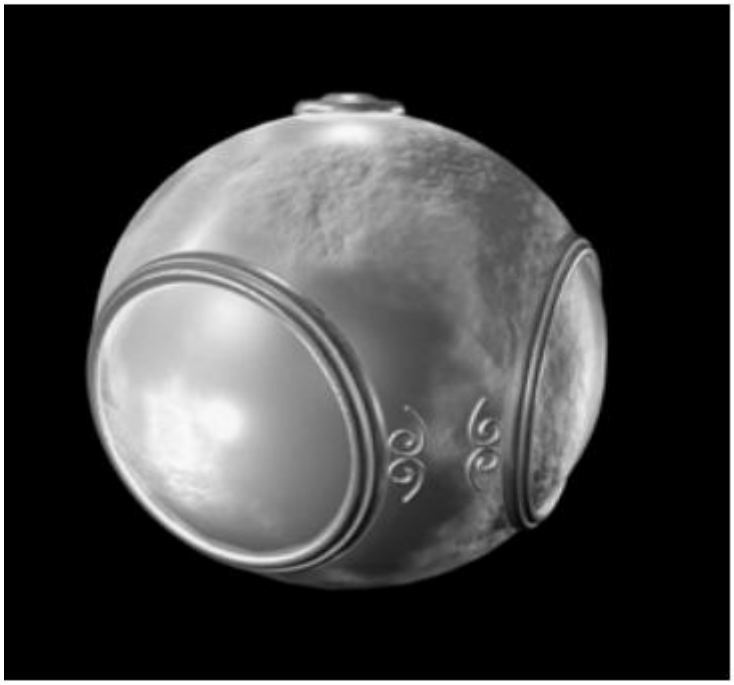
(a) ***Sphere Mapping Technique:*** In sphere mapping technique the single texture contains the image of the surroundings as reflected on a mirror ball. This method is surpassed by the cube mapping technique.

(b) ***Cube Mapping Technique:*** In cube mapping technique the environment is projected onto six faces of a cube. These projections are then stored as six square textures and are unfolded into

six square regions of a single texture. Some of the higher mapping techniques like pyramid mapping and octahedron mapping are also used in the environmental mapping technique.

3. ***Bump Mapping:*** Bump mapping alters the normal vectors during the process of rendering through emulation. The faces of various objects do not have the same color always. Random texturing can be depicted by accumulating the random deviations and calculating the surface normal as done in Phong shading. In the bump mapping method, the location of the surface is not changed, but the artificial change of orientation scatters light in all possible directions making the object appear as though it is from a series of bumpy surface. This method is known as bump mapping. The following figure 13.5 depicts the bumpy surface, pertaining to bump mapping.

Figure 13.5: Bump Mapping



Source: http://www.cs.kent.edu/~farrell/cg05/lectures/cg22.pdf

Unwrapping

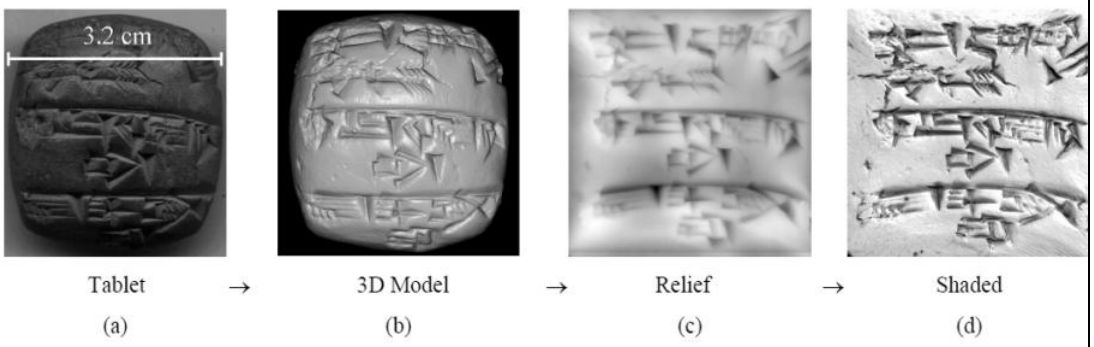
Before texturing any model, it is required to be mapped and unwrapped so that the lines and points of the image are clearly viewable without overlapping. Thus, unwrapping is the method of preparing a model for texturing.

Let us consider the cuneiform clay tablets of Mesopotamia. They are famous for the inscriptions which were made millions of years ago. To understand these writings, it is required to visualize the cuneiform writings by deciphering. This is done to ensure proper reproduction of the tablets on paper. Photographs were not adequate to visualize the tablets due to the following two reasons:

1. Insufficient single view point due to text wraps around the sides of the tablet.
2. Raking lights illuminating only a part of the textual information (this leaves the other texts either shadowed or invisible due to the obscured features on the tablet or when the tablets are nearly aligned with the lighting direction).

The following figure 13.6 depicts the various stages (a, b, c and d) of unwrapping the cuneiform tablets.

Figure 13.6: Cuneiform Unwrapping



Source: http://graphics.stanford.edu/papers/cuneiform/cuneiform-300dpi-images.pdf

Now, let us discuss about various stages found in above figure 13.6:

(a) ***Tablet:*** A photograph of an Ur III dynasty cuneiform tablet (from 2100 BC).

(b) ***3-D Model*:** A Phong-shaded rendering of a 50-micron resolution 3-D model containing 3.2 million triangles.

(c) ***Relief:*** The writing is unwrapped and shown as a displacement map.

(d) ***Shading:*** The unwrapped inscriptions have been accessibility colored, curvature colored, and rendered using phong shading to enhance readability.

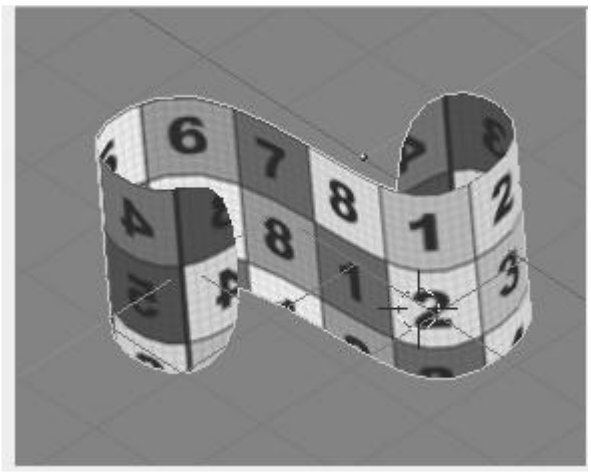
The issue with limitations of photography was solved with the help of a high resolution 3-D computer model from laser range data. The image is unwrapped by flattening the inscriptions found on the model to a plane. This is represented as a scalar displacement map as shown in figure 10.6 (c). Finally, the scalar displacement map is rendered non-photo realistically by using accessibility and curvature coloring. The output from this semi-automatic process enables to perceive the tablet's text in a single concise image.

The unwrapping techniques are also applied to other types of inscribed surfaces such as bas-reliefs.

The cuneiform inscriptions are believed to be the first written inscriptions and were made in moist clay tablets. These tablets were made before 3,500 BC. It has been found that more than 1, 00,000 tablets exist to this day in disintegrated state. The size of these cuneiform tablets ranges from 2 to 12 centimeters. These cuneiform tablets contain data relating to historic events and transactions of the past. The texts on these tablets are pictograms and hence are difficult to translate to texts.

The following figure 13.7 provides a clear idea about the way in which a cube is unwrapped using the Least Squares Conformal Maps.

Figure 13.7: Unwrapping - Least Squares Conformal Map (LSCM)



Source: http://www.blender.org/development/release-logs/blender-234/uv-unwrapping/

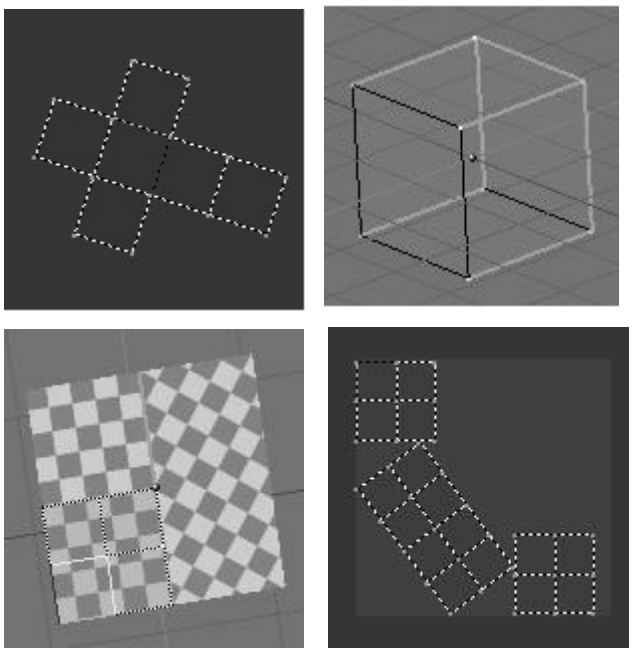
The Least Squares Conformal Map (LCSM) is an Ultra violet tool used to unwrap the arbitrary surfaces. It follows a method where the local angles are preserved similar to any other UV unwrapping mode. The select faces in the UV face select mode are wrapped.

To calculate the panel of LCSM:

1. Press the U key and then choose the LSCM, or the LSCM Unwrap from the UV calculation panel.
2. Cut the surfaces to unwrap.
3. Name the cut edges as seams.
4. The edges are marked or cleared as seams in the edit mode by the following options (Mesh -> Edges -> Mark Seam (Ctrl E).

The following figure in 13.8 gives a better understanding about the view of the cube after applying LSCM.

Figure 13.8: View of Cube-Least Squares Conformal Map (LCSM)



Source: http://www.blender.org/development/release-logs/blender-234/uv-unwrapping/

It is difficult to unwrap a mesh as one group of faces. Therefore, the mesh is cut into multiple groups of faces. The LCSM unwrapping unwraps the faces separately when the seams divide the selective faces into multiple face groups and then position these faces in the UV editor to avoid overlapping of face groups.

Select -> Select Linked UVs (L Key) (All the linked faces are selected when the seams do not divide them). Thus face groups are selected by selecting one face of the group and by using the Select Linked.

13.2 Ray Tracing

Ray tracing is a technique for the three-dimensional graphics which use complex light interactions. This technique will enable you to create pictures with mirrors, transparent surfaces, and shadows. The concept of ray tracing is based on reflection and refraction of rays that follow an iterative process.

Ray tracing is a simple yet powerful 3-D graphics method. It is easy to understand and implement without getting into the intricacies of mathematics. A very few basics of mathematics like vectors, dot products, and cross products are generally considered for ray tracing.

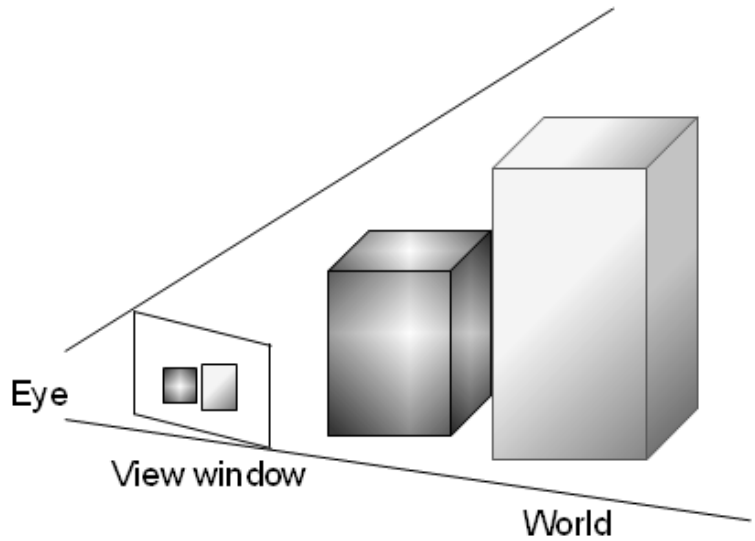
Let us now analyze ray tracing with the help of figure 10.9 which has the eye, view window, and world.

Caution

Before creating any kind of computer graphics ensure that you have a list of objects which you want your software to render. These objects are part of the scene or world.

The following figure 13.9 depicts the setup for ray tracing and is explained by considering the analogy of a camera, an eye and the view window.

Figure 13.9: Ray Tracing



In graphics, the viewpoint is called the **eye** or the camera. A camera requires a film on which the scene will be projected. In graphics, the scene is drawn on the view window. In a camera the film is placed behind the aperture or focal point while in graphics the view window lies in front of the focal point.

The light ray causes a color of each point on real film when passing through the aperture and hits the aperture. In computer graphics, a simulated light ray causes each pixel of the final image to hit the view window on its path towards the eye. The main intention is to find the color of each point which lies on the view window. The view window is divided into many small squares. Each of these squares corresponds to one pixel in the final image.

Notes

When you intend to create an image with resolution of 640x400 then the view window would be divided into a grid of 640 squares across and 400 square down. The challenging task lies in assigning a color to each of these squares and this is done by ray tracing

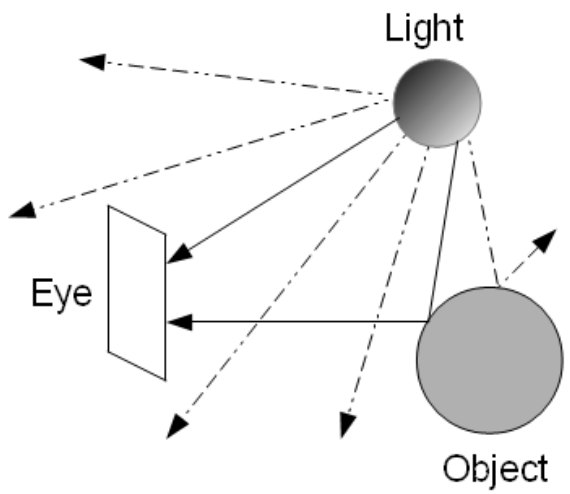
Let us now discuss about the working principle of ray tracing.

Ray tracing adapts the simulated path taken by the light rays that travel both forwards and backwards. The main goal is to determine the color of each light ray which hits the view window before reaching the eye.

Ray tracing starts from the origin of light source and travels to its destination .i.e. to the eye as shown in figure 13.10. When you consider tracing one ray with one light and one object such as a ball then you start from the light bulb and shoot the number of rays required from the bulb. The direction of each ray is decided. It was found that rays traveled in different directions some of them hit directly and others bounced back to reach the eye while the rest never reached the eye. A considerable amount of time was wasted during this course.

Let us now analyze ray tracing with the help of following figure 13.10.

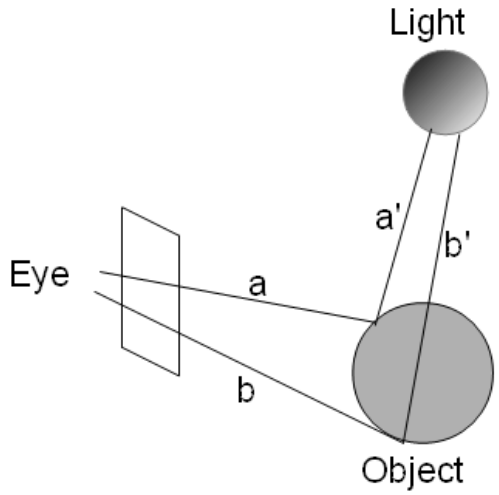
Figure 13.10: Ray Tracing



To avoid wastage of time, the rays which were guaranteed to hit the view window and reach the eye were traced. The point on the view window whose color is to be determined is selected. The color which passes through the selected point of the view window and reaches the eye is the color assigned to it.

The ray can also be traced backwards by starting at the eye and then passing through the point on the scene. These two rays would be similar except for their direction. Since, the original ray comes directly from the light source, the backward ray travels to the light source. When the original ray bounces back then the backward ray also bounces back. The following figure 13.11 depicts how rays travel backwards in ray tracing.

Figure 13.11: Ray Tracing



Ray tracing is a method which involves creation of a geometric scene through its lighting effects .i.e. arriving from a surface. This method was developed for modeling global illumination properties. Following are some of the examples of ray tracing.

Example: Secondary illumination (color bleeding) and reflection of objects from mirror.

Did you know? Ray tracing is named after its inventor by name Turner Whitted and hence known as Whitted ray tracing.

Ray tracing is used to synthesize realistic images. It can easily produce shadows and thus can model reflective and transparent objects. Due to its slow nature, ray tracing is used to generate realistic images, thereby enabling it to generate realistic texture maps and environment maps which are used for interactive rendering. This method is also used for producing very realistic types of complex shading and lighting in spite of its conceptual simplicity.

If the movement of light is modeled in a three-dimensional scene then it is possible to produce accurate renderings. However, this is not possible due to the computational complexity involved. As most of the rays, which originate from the incident light do not strike the viewer's eye and therefore, backward tracing is done to trace the light rays which hit the eye and this is the basic idea of ray tracing.

In ray tracing, only the perfect mirror reflections are considered by a recursive ray tracer.

Some of the advantages of ray tracing are that:

1. It is simple to implement and can generate impressive visual effects.
2. It is simple to analyze the effects like reflection and shadows using ray tracing (difficulty in simulating the algorithms is solved by using ray tracing).

13.3 Morphing

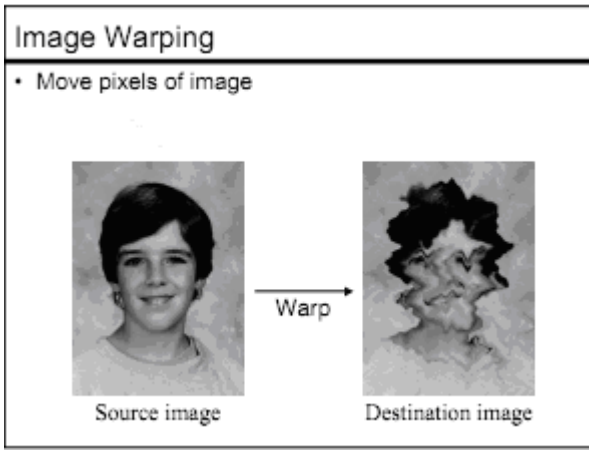
Morphing is the process used to transform (morph) one image to another image. To use morphing techniques, you do not require the knowledge of 3-D technology. This technique finds its application in photography, drawings, and image rendering scenes.

Morphing is widely used in motion pictures and animations where one image is changed to another image through continuous transitions. It is generally used to depict the transformation of one person to another person through technological means which appear bizarre and not real. This kind of depiction is obtained by using the cross fading technique. However, today the cross fading technique has been replaced by many computer software to develop more realistic transitions.

Notes Cross fading morphing is a type in which you come across events where, if one face fades, the other appears, then there is a blank screen and the other face appears simultaneously.

The following figure 13.12 gives a clear picture about how pixels from the source image are warped onto the destination image.

Figure 13.12: Image Warping



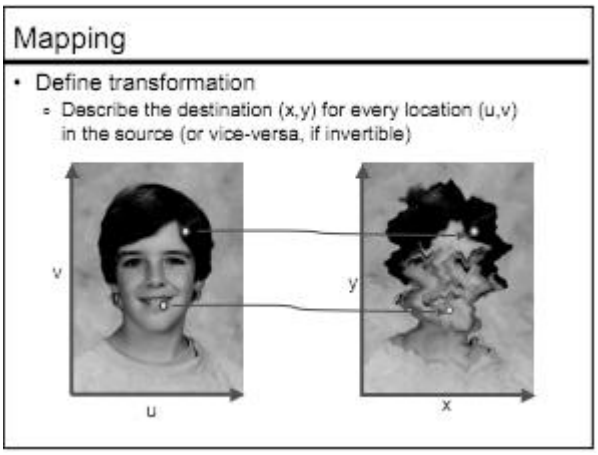
Source: http://www.cs.virginia.edu/~gfx/courses/2004/Intro.Fall.04/handouts/03-morph.pdf

View morphing is a technique carried out by pre-warping two images before the computation of morphing and then the images are post-warped after interpolating the images.

Did you know? Idea about the effects of morphing can be obtained from movies like Terminator 2 and Star Trek. In these movies, simple morphs were made through Reshape Filters in Adobe After Effects 5.5.

The following figure 13.13 depicts how an image is mapped from the source image to destination image, thereby facilitating in the transformation of the image.

Figure 13.13: Image Mapping



Source: http://www.cs.virginia.edu/~gfx/courses/2004/Intro.Fall.04/handouts/03-morph.pdf

Image morphing is a technique used to bring about 2-D transitions between images. It is difficult to correct image morphing manually due to its unnatural distortions.

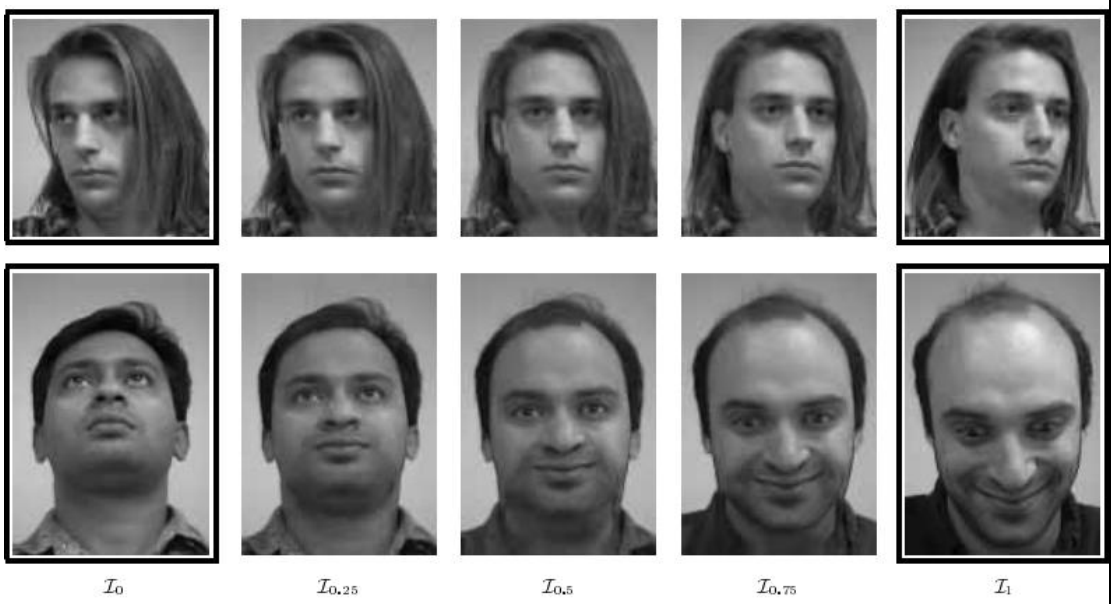
The technique of morphing involves changing the shape of image over and over again into another image. Morphing is generally used for visual effect purposes where a normal looking person can be morphed into a cartoon character. Using simple morphing with 2-D images and with a basic knowledge of after effects, it is possible to complete the task of morphing at a very fast pace.

Simple morphing happens due to the concurrent happening of three events such as:

1. Morphing the shape of the source image to the same shape to that of the final or destination image.
2. Warping the final or destination image to the source image without distortion to retain the normal shape.
3. Changing due to distortion causes the destination image to become visible over the source images.

The following figure 13.14 depicts the various transformations that can be made on an image. The first row of images show the morphs made between two views of the same person. The second rows of images show the morphs between views of two different people. In both the rows, morphing captures the change in facial pose between images I_0 and I_1. This image conveys a natural 3-D rotation.

Figure 13.14: Image Morphing



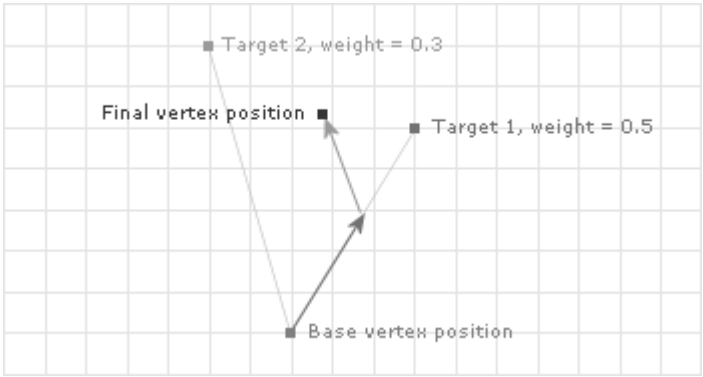
Vertex Morphing

Vertex morphing is a process which allows deformation of a base mesh by using morph targets and meshes, with identical vertices but having altered vertex positions to the base mesh. The vertex morphing technique is used for bringing out facial expressions on 3-D characters. As, the transformation of the image cannot be described due to the presence of bones on the face, vertex morphing method is used.

Let us now analyze the working of vertex morphing as shown in figure 13.15. Vertex morphing works on the principle of interpolation where the final vertex position between a base position and a target position is defined in separate meshes known as **morph targets.**

The figure 13.15 depicts the process of morphing a single vertex using two morph targets. This method uses linear interpolation to calculate the position of the final vertex. The following figure 13.15 depicts the method involved in vertex morphing.

Figure 13.15 Vertex Morphing



Source: http://www.mdxinfo.com/tutorials/hardwarevertexmorphing.php

Morphing techniques are used in 3-D modeling packages. However, in real time graphics it is used to interpolate to a specific target and a subset of the vertices is morphed. Although, this technique is not very popular, it is considered to be an efficient one. This technique is used for animating models with complex and active morph targets. The following is the example of the Johnny fish character, a 3-D animation created many years ago. The creation of the character is explained using multiple vertex streams.

Example: Rendering the Johnny fish character (animation character) with three morph targets.

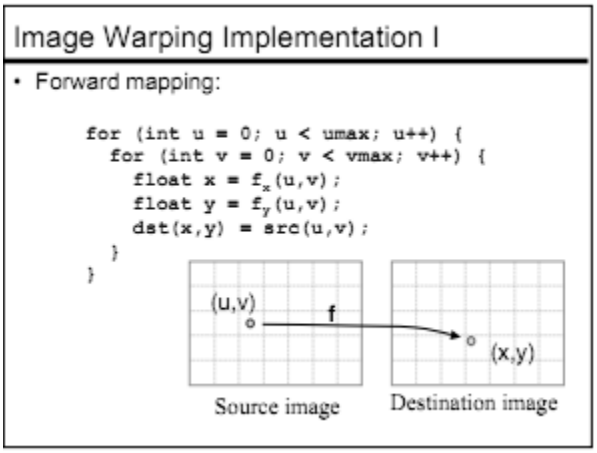
The example on multiple vertex streams can be analyzed by the following example. The Johnny base mesh was loaded to set 4 separate streams where one was a base mesh while the other 3 were targets. This was done by using the appropriate vertex declaration to interleave the data on the GPU. Thus, it was possible to access the morph target vertices in the vertex shader and the final vertex positions were interpolated based on the weights set for each of the targets.

The two methods by which images can be morphed are:

1. ***Forward Mapping:*** In the forward mapping method, the pixels from the source images are mapped onto the destination image at appropriate places. There are possibilities of some pixels not getting mapped on to the destination image. In this situation, interpolation is used to determine these pixels. The point morphing algorithm uses this method of morphing.

The following figure 13.16 gives a better idea about forward mapping, where the pixels from the source image are mapped onto the pixels in the destination image.

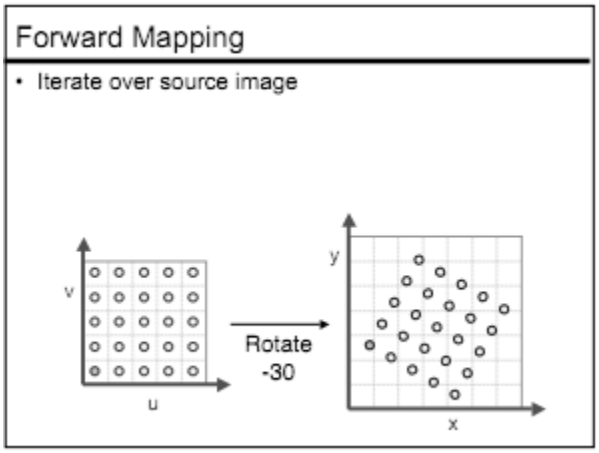
Figure 13.16: Forward Mapping



Source: http://www.cs.virginia.edu/~gfx/courses/2004/Intro.Fall.04/handouts/03-morph.pdf

The following figure 13.17 depicts how the pixels in the source image appear when rotated at an angle of -30 degree in forward mapping.

Figure 13.17: Forward Mapping

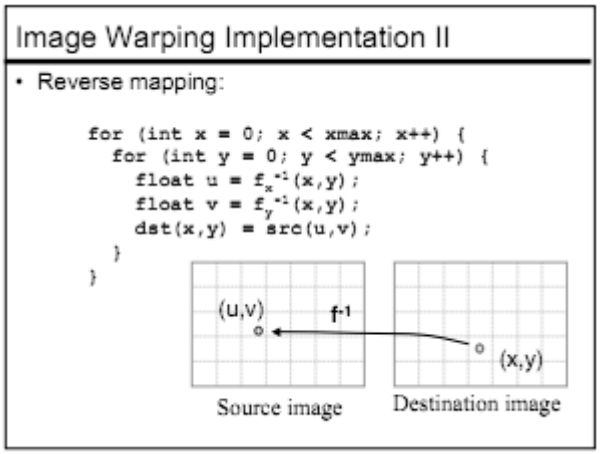


Source: http://www.cs.virginia.edu/~gfx/courses/2004/Intro.Fall.04/handouts/03-morph.pdf

2. ***Reverse Mapping*:** The reverse mapping method involves mapping each pixel in the destination image and sampling an appropriate source image. This assures mapping of all the destination image pixels to the source image pixel. Beier/Neely line morphing method uses this method of morphing.

The following figure 13.18 depicts reverse mapping where each pixel in the destination image is mapped by sampling appropriately in the source image.

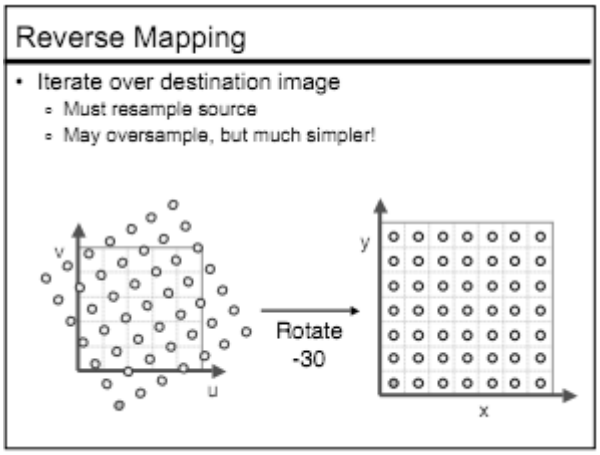
Figure 13.18: Reverse Mapping



Source: http://www.cs.virginia.edu/~gfx/courses/2004/Intro.Fall.04/handouts/03-morph.pdf

The following figure 13.19 depicts how the pixels in the source image appear when re-sampled using reverse mapping.

Figure 13.19: Reverse Mapping

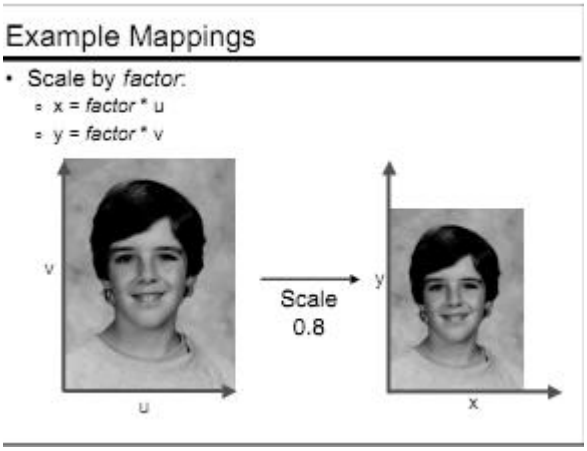


Source: http://www.cs.virginia.edu/~gfx/courses/2004/Intro.Fall.04/handouts/03-morph.pdf

The reverse mapping method is a tedious process to identify the way in which pixels of the destination image are mapped on to the pixels in the source image. In order to know how each pixel moves between the destination and source image, only few of the important pixels are mapped. The motion of the other pixels is obtained by extrapolating the information of the control pixels. These control pixels are specified as lines in one image and are mapped onto the lines in the other image, or the points from one image are mapped onto the points in the other image.

The following figure 13.20 enables you to have a better understanding about mapping images using a scaling factor.

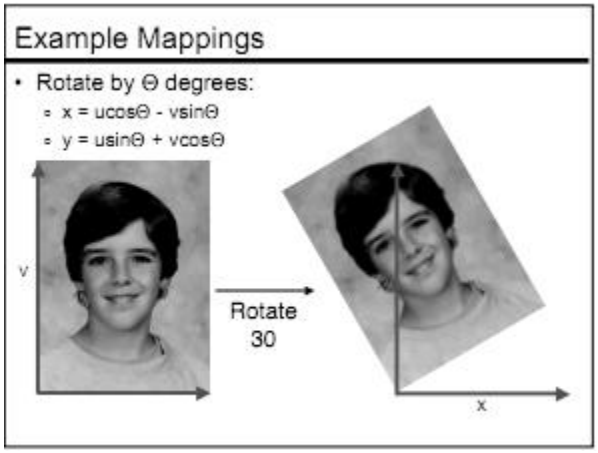
Figure 13.20: Image Mapping



Source: http://www.cs.virginia.edu/~gfx/courses/2004/Intro.Fall.04/handouts/03-morph.pdf

The following figure 13.21 depicts the view of an image when rotated by angle theta (θ).

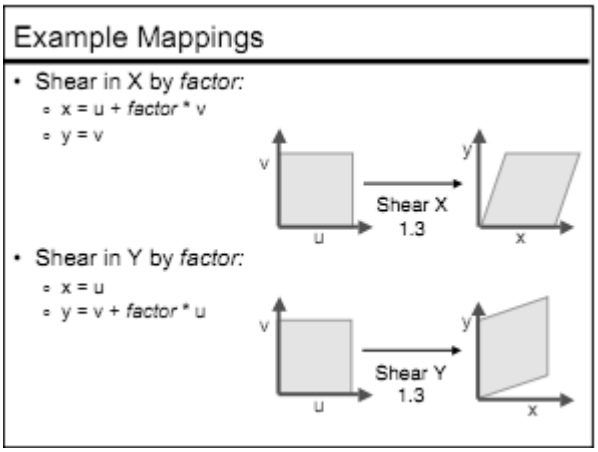
Figure 13.21: Image Mapping



Source: http://www.cs.virginia.edu/~gfx/courses/2004/Intro.Fall.04/handouts/03-morph.pdf

The following figure 13.22 shows technique of image mapping done using a shear in factor X and Y.

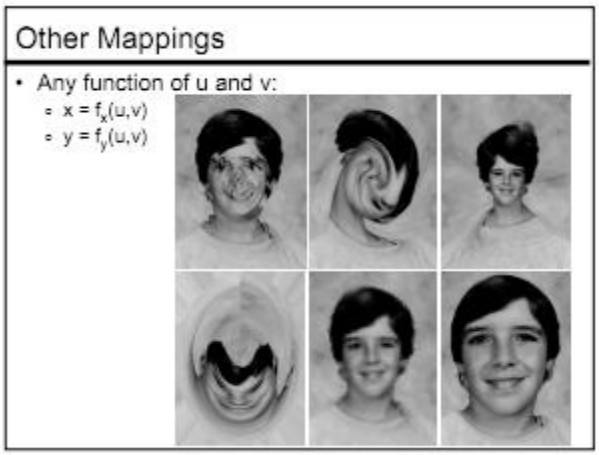
Figure 13.22: Image Mapping



Source: http://www.cs.virginia.edu/~gfx/courses/2004/Intro.Fall.04/handouts/03-morph.pdf

The following figure 13.23 shows the various transformations that an image undergoes when subjected to morphing technique.

Figure 13.23: Image Mapping



Source: http://www.cs.virginia.edu/~gfx/courses/2004/Intro.Fall.04/handouts/03-morph.pdf

Task

Research to study about the various image transformations, pertaining to morphing.

13.4 Summary

- Texture mapping is considered to be the most appropriate type of texturing.
- Texture mapping involves pasting the images on the surface of objects.
- The main goal of ray tracing is to determine the color of each light ray which hits the view window before reaching the eye.
- Morphing is generally used to generate visual effects; thereby a normal looking person can be morphed into a cartoon character by employing 2-D and After Effects 5.5.
- Forward mapping, maps the pixels from the source image onto the destination image at appropriate places.
- Reverse mapping, maps each pixel in the destination image and samples an appropriate source image.
- Vertex morphing works on the principle of interpolation, where the final vertex position between the base position and the target position are defined in separate meshes.

13.5 Keywords

API: It is a set of rules and specifications that a software program must follow to be able to access the services and resources of the other software program.

Beier/Neely line morphing: This method is used to warp the shape & cross-dissolve the colors independently.

Bas-reliefs: It is a type of art, where the shapes are cut from the surrounding stone to make them stand, slightly against a flat background.

Bizarre: Unusual situation.

Cuneiform: Writings related to ancient times (Babylonians and Persians). These characters are formed by the arrangement of wedge-shaped elements.

***Curvature Coloring*:** The regions with similar curvatures are colored with similar colors.

***Deciphering*:** To discover the meaning of something written in a hidden manner.

GPU: Graphical or visual processing unit.

***Pictograms*:** Pictograms are also known as pictographs. The meaning of objects is conveyed pictorially based on their resemblance to the object.

***Plotters*:** Plotters are printing devices used for printing vector graphics.

Pre-warping: It is the method used for solving problems related to warping.

***Raking Lights*:** Moving light.

***Reflectance*:** It is the ratio of the total amount of radiation of light that is reflected by a surface to the total amount of radiation that is incident on a surface.

Rendering Time: Rendering is the process of generating an image from a model by using computer graphics. The time used for the purpose of generation of the image is known as rendering time. The Graphical Process Unit (GPU) is the built-in device that helps the Central Processing Unit (CPU) to perform complex rendering.

Tactile Texture: Pleasant texture.

Warping: Warping is a process of digitally manipulating an image with the aim of distorting the image. It can also be used to rectify image distortions for creative purposes known as morphing.

13.6 Self Assessment

1. State whether the following statements are true or false:
   (a) A bitmap is represented by 'zeros' and 'ones' of a rectangular array of points.
   (b) Texturing can be either two-dimensional or three-dimensional.
   (c) Physical textures depict variations on surfaces.
   (d) Visual textures are created by repetitive use of shape and line.
   (e) Forward mapping and reverse mapping are two methods of texturing.
   (f) Unwrapping is similar to texture mapping.
   (g) Cross-fading is the latest technique used in morphing.
2. Fill in the blanks:
   (a) To unwrap the cuneiform tablets ……. shading is employed.
   (b) Pixels are made brighter and lighter by employing ……. shading.
   (c) Environmental mapping is also known as ……. mapping.
   (d) In ray tracing, rays are also traced …….
   (e) 2-D and After Effects help to perform ……. in a better manner.
3. Select a suitable choice for every question:
   (a) Beier /Neely line-morphing method is used in which of the following mapping.
      (i) Forward mapping
      (ii) Reverse mapping
      (iii) Vertex mapping
      (iv) Texture mapping
   (b) In vertex morphing, the target position is defined in separate meshes known as.
      (i) Morph target
      (ii) Texture
      (iii) Physical texture
      (iv) Warping
   (c) Transformation of an image to a cartoon, relates to which of the following.
      (i) Morphing
      (ii) Texturing
      (iii) Shading
      (iv) Reflectance

13.7 Review Questions

1. Define texturing and discuss the physical and the visual textures.
2. Distinguish between texturing and unwrapping.
3. Photography was not adequate to visualize the cuneiform tablets. Justify
4. Ray tracing uses backward tracing of rays. Elaborate.
5. Cross-fading is taken over by computer graphics. Discuss.

6. Discuss how reverse mapping and forward mapping are used in morphing?
7. Is morphing of images required for animation?
8. The process of vertex morphing. Elaborate.

Answers: Self Assessment

1. (a) True (b) True (c) True (d) True
   (e) False (f) False (g) False
2. (a) Phong (b) Cosine (c) Reflective
   (d) Backwards (e) Morphing
3. (a) Reverse mapping (b) Morph targets (c) Morphing

13.8 Further Readings

Books

Shirley, P., & Ashikhmin, M., & Willemsen, P.(2005). Fundamentals of Computer Graphics, 2nd ed. USA: SAMS Publishing.

Foley, J.D., & Van, D.,& Feiner., & Hughes.(1996). Computer Graphics-Principles and Practice, 2nd ed in C. USA: Addison-Wesley Publishing.

Jones, H.(2001). Computer Graphics Through Key Mathematics. UK: Springer-Verlag.

Online link

http://www.cs.kent.edu/~farrell/cg05/lectures/cg22.pdf

http://books.google.co.in/books?id=A4k29b0BdVMC&dq=computer+graphics+best+books&printsec=frontcover&source=in&hl=en&ei=93KDTb3WFYe0rAf4p-zNCA&sa=X&oi=book_result&ct=result&resnum=11&ved=0CGUQ6AEwCg#v=onepage&q=computer%20graphics%20best%20books&f=false

http://books.google.co.in/books?id=0VOEjFmPB-0C&dq=computer+graphics+best+books&printsec=frontcover&source=in&hl=en&ei=93KDTb3WFYe0rAf4p-zNCA&sa=X&oi=book_result&ct=result&resnum=13&ved=0CGkQ6AEwDA#v=onepage&q=computer%20graphics%20best%20books&f=false

http://books.google.co.in/books?id=x6yYmUODjhQC&printsec=frontcover&dq=tEXTURE&hl=en&ei=xeGBTaCXN43PrQeh8t27CA&sa=X&oi=book_result&ct=result&resnum=1&ved=0CDAQ6AEwAA#v=onepage&q&f=false

http://books.google.co.in/books?id=z14FpzYluWQC&pg=PA319&dq=Computer+graphics-Texture&hl=en&ei=DeOBTZSsDonIrQehzvTNCA&sa=X&oi=book_result&ct=result&resnum=3&ved=0CD8Q6AEwAg#v=onepage&q&f=false

Unit 14: Animation in Computer Graphics

CONTENTS

Objectives

Introduction

14.1 Animation

14.2 Additional Visual Effects

14.3 Summary

14.4 Keywords

14.5 Self Assessment

14.6 Review Questions

14.7 Further Readings

Objectives

After studying this unit, you will be able to:

- Discuss animation
- Explain additional visual effects

Introduction

Visual effect is part of advanced computer graphics that pertains to the various processes through which images are created and manipulated in the absence of a live environment. When the sets are costly to setup and when it becomes impossible to capture a film or object, visual effects come into picture. They provide realistic image of the live environment. The images in visual effects are computer generated and can be seen in most of the movies.

Visual effects provide more appeal to the story line of a movie and are generally completed after the completion of movie but are pre-planned before the commencement of a movie shoot.

Visual effects employ graphic designing, modeling, animation and other relevant software but special effects are made on the sets.

Today, one can find the application of computer graphics and visual effects and animation in diverse fields such as science, technology, military, advertising, entertainment, and so on.

14.1 Animation

The word animation is derived from the Latin word 'anima' which is the animating principle. It is used as a translation for the Greek word psyche and relates to the Christian concept of soul. Hence, animation can be defined as a technique of giving soul to drawings and art work which are lifeless. Earliest examples of animation can be traced back to the Paleolithic cave paintings where attempts were made to capture the phenomenon of motion drawing.

The term animation is used to display a sequence of images rapidly employing either 2-D or 3-D art works or positions of various models to bring out an illusionary movement to the models and art work. This is done through optical illusion of motion by employing the phenomenon of **persistence of vision** in which images can be created and depicted in several ways. Motion picture or video programs are the commonly used methods of animation.

Persistence of Vision

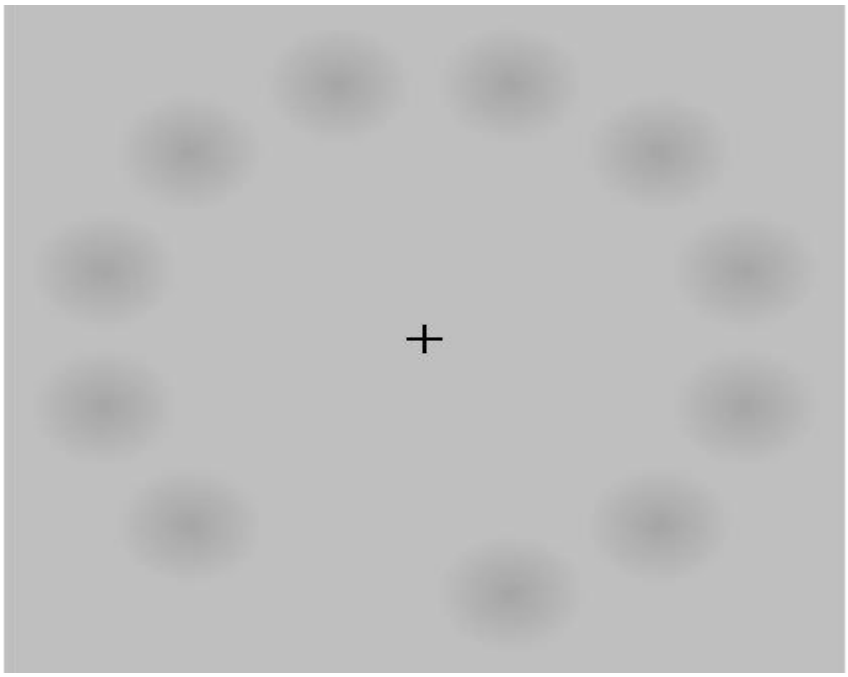
Persistence of vision is a phenomenon of the eye where an afterimage is thought to persist for one twenty-fifth of a second on the retina of the eye. In the visual perception phenomenon, the eye is not considered as a camera. The phi phenomenon has a more constructionist approach towards cinema while persistence of vision is a realistic approach.

Today, persistence of vision is an accepted phenomenon in the history of cinema. In the early days, it was determined that a frame rate of less than 16 frames per second had enabled the mind to see flashing images. Audiences, sometimes interpret motion at ten frames per second or even slower. The modern theatrical films run at 24 frames per second.

The two perceptual illusions are:

1. ***Phi Phenomenon:*** This method is an optical illusion defined by Max Wertheimer in the Gestalt psychology in 1912. It is based on the principle that the human eye is able to perceive movements based on pieces of information. This is similar to the succession of images. The speed of the images per second can be observed for the phi phenomenon. The figure 14.1 shows the phi phenomenon and depicts how an object moves in a circular motion with constant speed.

Figure 14.1: Phi Phenomenon



Source: http://en.wikipedia.org/wiki/Phi_phenomenon

2. ***Beta Movement:*** The beta movement is considered to be a perceptual illusion where the brain merges two more still images, imaginatively to be in motion. This phenomenon creates an illusion of motion both towards and away from the viewer. Consider phenomenon of a viewer watching a screen. This screen is projected with two images in succession. The first image shows a ball on the left side of the frame, while the second image shows a ball on the right side of the frame. These images are shown in rapid succession. Thus, as a viewer you can see one ball moving from the left towards the right but not two balls flashing in succession. When the two objects are of different size that is when the first image is large and the second image is small. As a viewer, you can only view that the object is moved away from us.

The following figure 14.2 depicts the beta phenomenon and provides a clear understanding about how the image keeps changing its position.

Figure 14.2: Beta Phenomenon

Source: http://upload.wikimedia.org/wikipedia/commons/0/09/Phi_phenomenom_no_watermark.gif

Did you know? The earliest example of animation is the 5,000 year old earthen bowl that was found in Iran in Shahr-i-Sokhta.

Animation came into existence with the advent of cinematography. Georges Melies is considered as the creator of special effect films and is also the first person to have used animation with special effects. George Melies came up with the stop motion animation. In this method, the rolling camera was stopped to make changes in the scene and then continued to roll the film.

This technique was discovered accidently when George Melies' camera broke while shooting a moving bus. Immediately after this incident, George restarted shooting; this time he happened to shoot a hearse pass by. Finally, George observed that he had transformed the moving bus into a moving hearse. This made George analyze about what he had created. This concept was later used in the field of animation and George Melies is considered to be a great contributor to the field of animation.

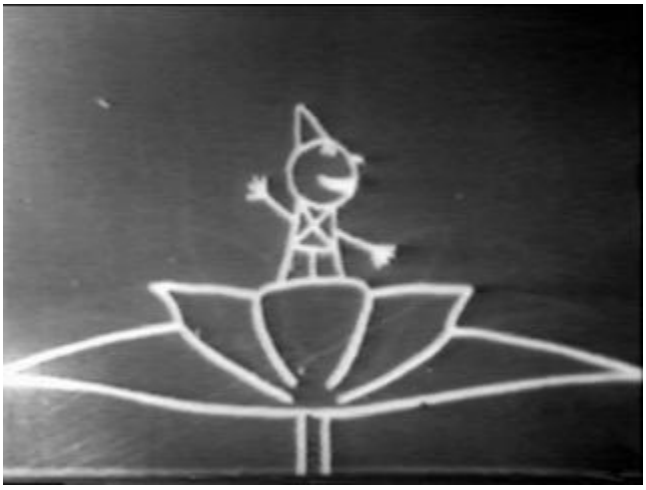
Notes

Matches was the earliest stop motion advertising film and was short by Arthur Melbourne Cooper while J. Stuart Blackton was the first American film maker who used both stop motion technique and hand drawn animation.

Emile Cohl, a French artist created a film called Fantasmagorie by drawing cartoon strips. This film was made by using a lot of a stick figures moving about and morphing them. Morphing a wine bottle into a flower is one such example. Each of the frames was drawn on paper and then they were shot onto negative films. This gave the picture a blackboard look.

The following figure 14.3 depicts Fantasmagorie drawings and is considered to be the first traditional hand-written animation.

Figure 14.3: Animation in Fantasmagorie



Source: http://en.wikipedia.org/wiki/Animation

Winsor McCay, a newspaper cartoonist created detailed animations that required attention for detail. In this method each frame was drawn on paper. This required backgrounds and characters to be redrawn and animated.

Did you know? Little Nero, Gertie the Dinosaur and The Sinking of Lusitania are some of the noted films made by McCay.

Animated films are also referred to as **cartoons,** which is an industry of its own today. John Randolph Bray is a successful producer of animations along with Earl Hurd who came up with the idea of cel animation process.

The photographs of the drawings are drawn on paper. To create an illusion of movement the drawings are done in such a way that they slightly differ from the previous one. Then these drawings are photocopied onto transparent acetate sheets called cels. These cels are filled with paints and are assigned colors on the opposite side of the line drawings. The cels are then photographed one by one onto motion picture film against a painted background by a rostrum camera. Cel animation is obsolete now.

Today, drawings with their backgrounds are scanned using a computer system. Software programs are used to color the drawings and their effects are simulated.

Did you know? Animation which uses extensive computer technology is known as Tradigital.

Example: Examples of traditional animated feature films are Pinocchio, Animal Farm and Akira.

Examples of traditional animated films made using computer technology are Lion King, Sen to Chihiro no Kamilakushi (Spirited Away), Les Triplettes de Belleville.

Nowadays, a complete animated film is done in a variety of styles such as the films produced by Walt Disney to the Cartoon Styles produced by the Warner Brothers.

Computer Animation

Computer animation uses a variety of techniques to be created digitally on a computer.

The following are some of the techniques:

1. ***2-D Animation:*** In 2-D animation, figures are created and edited on the computer by either using 2-D bitmap graphics or by using 2-D vector graphics. This method uses the traditional animation techniques like tweening, morphing, onion skinning, and interpolated rotoscoping.

Example: Foster's Home for Imaginary Friends, Danny Phantom, Waltz with Bashir, The Grim Adventures of Billy and Mandy are some of the examples for 2-D films.

2. ***3-D Animation:*** In 3-D animation the figures are digitally modeled and manipulated using an animator. A process known as rigging is carried out. In this process to manipulate a mesh, a digital skeletal structure is used to control the mesh. Techniques like mathematical functions, simulated fur or hair, effects such as fire and water and the use of motion capture, fall under the 3-D dynamics. Sometimes it is difficult to distinguish between 3-D animations and live actions. In such cases visual effects are used.

Notes

2-D animation techniques focus on image manipulation while 3-D techniques build virtual worlds in which characters and objects move and interact. 3-D animation can create images that seem real to the viewer.

14.2 Additional Visual Effects

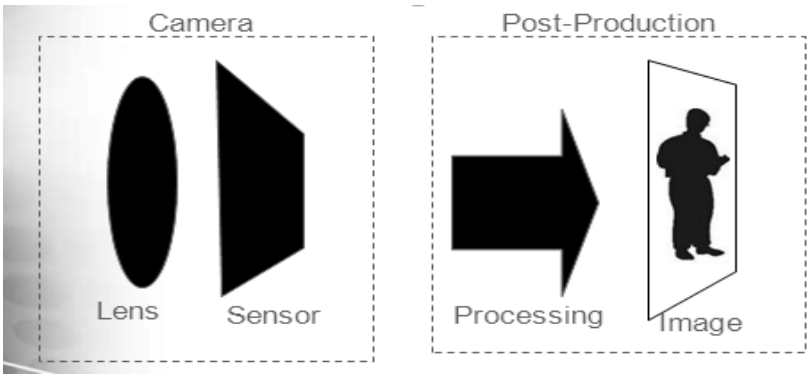
Visual effects, also known as special effects, or VFX or FX are applied to computer generated effects called Computer generated imagery (CGI). CGI deals with the field of special effects and is used to create effects in films and television commercials. Video games use real time computer graphics.

Visual effect is a process by which images are created and manipulated in the absence of a live action shoot. The live action footage and the generated imagery are integrated to create an environment which looks realistic.

Today, visual effects using CGI have become very common in big budget films. Through affordable animation and the right assembled software, a novice film maker can make brilliant visual effect films.

Visual effects modify the virtual world as shown in figure 14.4.

Figure 14.4: Visual Effects



The word virtual implies creating something that does not exist and making people believe that the creation is indeed real.

The three scenarios required for VFX are:

1. To check what we want to see does not exist.
2. To check what we want to see is too difficult or dangerous to photograph live.
3. To check what we need to fix something we have photographed.

The output got from visual effects is considered to be of high quality and the effects are more controllable when compared to the physical processes like constructing miniatures for introducing special effect shots in places which require crowd scenes to be added. The CGI enables creation of images which cannot be created using other technologies.

Notes

Visual Effects

1. Pyrotechnics such as mortars for earthy explosions and gasoline balls for flaming are created by visual effects.
2. Stunt related effects like falls or impacts can be created using special effects.
3. A single artist can produce content without using actors, expensive sets or properties.
4. Wired and unwired techniques can be created using visual effects.

The figure 14.5 illustrates a dynamite explosion that killed Dr. Leslie Arzt while handling the dynamite. This explosion was almost composited onto the screen using CGI.

Figure 14.5: Pyrotechnics Visual Effects

Source: http://lostpedia.wikia.com/wiki/Visual_effects#CGI

The figure 14.6 shows how Frank Torres performed the Turbine Man using a pneumatic ratchet rig stunt in the series Pilot, Part-1. The same footage was used again by compositing the Expose series employing CGI.

Figure 14.6: Stunt Related Visual Effects

Source: http://lostpedia.wikia.com/wiki/Visual_effects#CGI

The figure 14.7 shows the long shots of the Barracks that was created using composite images. The houses along with the other buildings were also created.

Figure 14.7: Creation of a Set (Barracks) Using Visual Effects

Source: http://lostpedia.wikia.com/wiki/Visual_effects#CGI

The figure 14.8(a) and figure 14.8 (b) provide a clear idea about how the computer generated imagery appears when it is wired and unwired. Wired and unwired techniques of visual effects method involve the process of digitally painting. The wires and the other accessories used in the film are removed.

Figure 14.8(a): Wired and Unwired Techniques of Visual Effects

The figure 14.8 (b) provides a clear idea about wired and unwired technique used in visual effects.

Figure 14.8(b): Wired and Unwired Techniques of Visual Effects

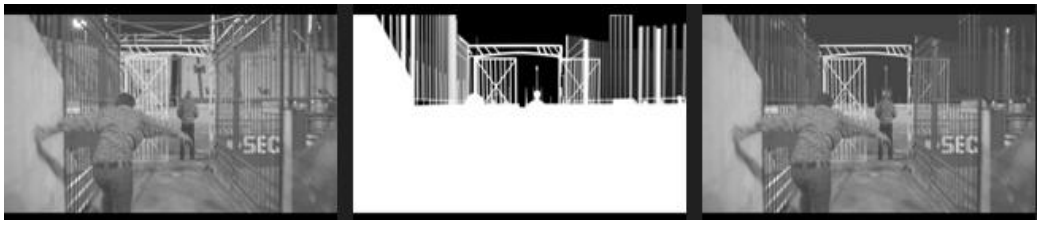
The following are the types of visual effects:

1. ***Rotoscoping:*** Rotoscoping is a method patented by Max Fleischer in 1917. In this method the animators trace live action movements frame by frame. Then the source film is directly copied by using the actor's outlines and this is changed to animated drawings of a live environment. It is a systematic method of isolating specific objects from the photographic plates. When a VFX shot is not filmed on an appropriate background then the background is created with the help of rotoscoping technique by digitally cutting the foreground of the actors frame by frame.

Example: The Lord of the Rings is an example for Rotoscoping method.

The figure 14.9 depicts how the unwanted objects in a shot are made invisible by employing rotoscoping, painting, and cloning methods.

Figure 14.9: Rotoscoping Technique



Source: http://www.malditochroma.com/services.html

2. ***Motion Tracking:*** In the motion tracking technique, the movements are made on the camera. While filming the 2-D/3-D, software is integrated with the CG elements and merged with the original shot to generate a fine motion tracking process and get realistic and integrated visual effects.

The figure 14.10 depicts motion tracking, where the CG elements are integrated with the original shot.

Figure 14.10: Motion Tracking

Source: http://www.malditochroma.com/services.html

3. ***Stereoscopy:*** Stereoscopy is also known as stereoscopic or 3-D imaging. This technique is used to create an illusion of depth in an image by presenting two separate offset images, one towards the right and the other towards the left of the viewer's eye. The two, 2-D off-set images are merged in the brain to provide the depth of perception of 3-D image. A high level of precision involved in the creation of the image provides images with the best visual impact. The Figure 14.11 depicts the stereoscopic conversion from 2-D to 3-D.

Figure 14.11: 2-D to 3-D Stereoscopic Conversion

Source: http://www.malditochroma.com/services.html

The listed points are required to follow the strategies used in additional visual effects:

1. To have the viewer wear eyeglasses to combine separate images from two offset sources.
2. To have the viewer wear eyeglasses to filter offset images from a single source separated to each eye.
3. To have the light source split the images directionally into the viewer's eyes (no glasses required).

14.3 Summary

- Computer graphics deals with high dimensional visual objects with special effects and virtual effects along with animation which is the recent development.
- The increase in the sophisticated animation systems has led to the generation of high quality computer animation making it simpler.
- Simulation softwares are used for graphical simulations.
- This is done through optical illusion of motion by employing the phenomenon of persistence of vision in which images can be created and depicted in several ways.
- Animation software is made up of the tricks and best practices of conventional animation.

14.4 Keywords

***Cloning*:** Make exact copy of a file.

Composited: Assembled, combined.

Cuneiform: Writings related to ancient times (Babylonians and Persians). These characters are formed by the arrangement of wedge-shaped elements.

***Curvature Coloring*:** The regions with similar curvatures are colored with similar colors.

***Deciphering*:** To discover the meaning of something written in a hidden manner.

***Digital Painting*:** It is a painting method that involves using digital tools and software to paint the surface required.

GPU: Graphical or visual processing unit.

Novice: A person who is not experienced in a job or a fresher in any field.

Paleolithic: Relating to Old Stone Age.

Pictograms: Pictograms are also known as pictographs. The meaning of objects is conveyed pictorially based on their resemblance to the object.

***Plotter*s**: Plotters are printing devices used for printing vector graphics.

Pre-warping: It is the method used for solving problems related to warping.

Raking Lights: Moving light.

Reflectance: It is the ratio of the total amount of radiation of light that is reflected by a surface to the total amount of radiation that is incident on a surface.

Rendering Time: Rendering is the process of generating an image from a model by using computer graphics. The time used for the purpose of generation of the image is known as rendering time. The Graphical Process Unit (GPU) is the built-in device that helps the Central Processing Unit (CPU) to perform complex rendering.

Tactile Texture: Pleasant texture.

Tweening: It is an interpolation technique used in an animation program to generate extra frames between the key frames that a user creates. It is considered as an advantageous method, as the user does not draw every frame.

Warping: Warping is a process of digitally manipulating an image with the aim of distorting the image. It can also be used to rectify image distortions for creative purposes known as morphing.

14.5 Self Assessment

1. State whether the following statements are true or false:
   (a) Phi Phenomenon is based on the principle that the human eye is able to perceive movements based on pieces of information.
   (b) Visual effect is a process by which images are created and manipulated in the absence of a live action shoot.
   (c) Rotoscoping is a method patented by Max Fleischer in 1937.
2. Fill in the blanks:
   (a) The …… phenomenon creates an illusion of motion both towards and away from the viewer.
   (b) In the ……..…, the movements are made on the camera.
   (c) The …… is a process by which images are created and manipulated in the absence of a live action shoot.
   (d) Animation employs the phenomenon of …….
3. Select a suitable choice for every question:
   (a) Today, animations are also referred as …….
      (i) Cartoons
      (ii) Drawings
      (iii) Pixels
      (iv) Morphs
   (b) 2-D and 3-D are the techniques of which of the following.
      (i) Animation
      (ii) Warping
      (iii) Texturing
      (iv) Morphing

14.6 Review Questions

1. Discuss the phenomenon of 'persistence of vision' with respect to animation.
2. Discuss briefly about the techniques of animation.
3. Elaborate on the various animation techniques.
4. Discuss the three scenarios required for additional visual effects.
5. Distinguish and discuss rotoscoping and stereoscopy.

Answers: Self Assessment

1. (a) True (b) True (c) False
2. (a) Beta movement (b) Motion tracking technique (c) Visual effect
   (d) Persistence of Vision
3. (a) Cartoons (b) Animation

14.7 Further Readings

Books

Shirley, P., & Ashikhmin, M., & Willemsen, P.(2005). Fundamentals of Computer Graphics, 2nd ed. USA: SAMS Publishing.

Foley, J.D., & Van, D., & Feiner., & Hughes.(1996). Computer Graphics-Principles and Practice, 2nd ed in C. USA: Addison-Wesley Publishing.

Jones, H.(2001). Computer Graphics Through Key Mathematics. UK: Springer-Verlag.

Online link

http://www.cs.kent.edu/~farrell/cg05/lectures/cg22.pdf

http://books.google.co.in/books?id=A4k29b0BdVMC&dq=computer+graphics+best+books&printsec=frontcover&source=in&hl=en&ei=93KDTb3WFYe0rAf4p-zNCA&sa=X&oi=book_result&ct=result&resnum=11&ved=0CGUQ6AEwCg#v=onepage&q=computer%20graphics%20best%20books&f=false

http://books.google.co.in/books?id=0VOEjFmPB-0C&dq=computer+graphics+best+books&printsec=frontcover&source=in&hl=en&ei=93KDTb3WFYe0rAf4p-zNCA&sa=X&oi=book_result&ct=result&resnum=13&ved=0CGkQ6AEwDA#v=onepage&q=computer%20graphics%20best%20books&f=false

http://books.google.co.in/books?id=x6yYmUODjhQC&printsec=frontcover&dq=tEXTURE&hl=en&ei=xeGBTaCXN43PrQeh8t27CA&sa=X&oi=book_result&ct=result&resnum=1&ved=0CDAQ6AEwAA#v=onepage&q&f=false