ICT 4203

Computer Graphics and Animation

Lecture 08

Scan-line Polygon Filling
Aliasing Effects & Anti-Aliasing

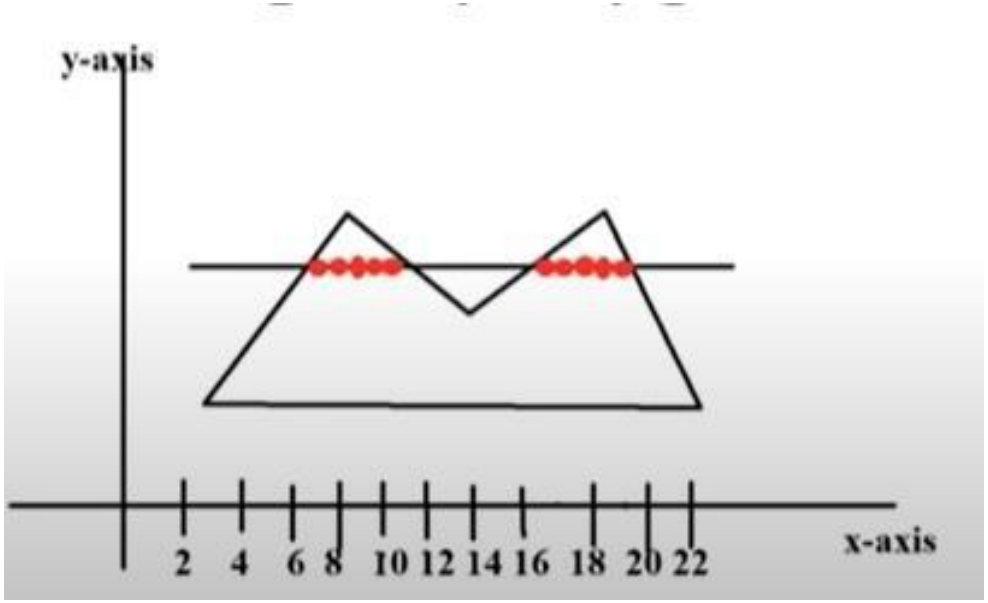
Md. Mahmudur Rahman
Lecturer
Institute of Information Technology

Contents

- Scan-line Algorithm
- Scan-Converting a Character
- Aliasing Effects
- Anti-Aliasing Techniques

Scan-Line Algorithm

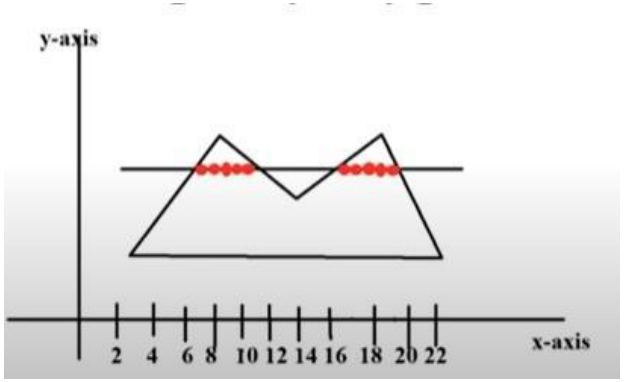
- Scan line polygon filling algorithm is used for solid color filling in polygons.



Steps to Perform:

For scan line polygon filling, there are three steps to perform in the following order:

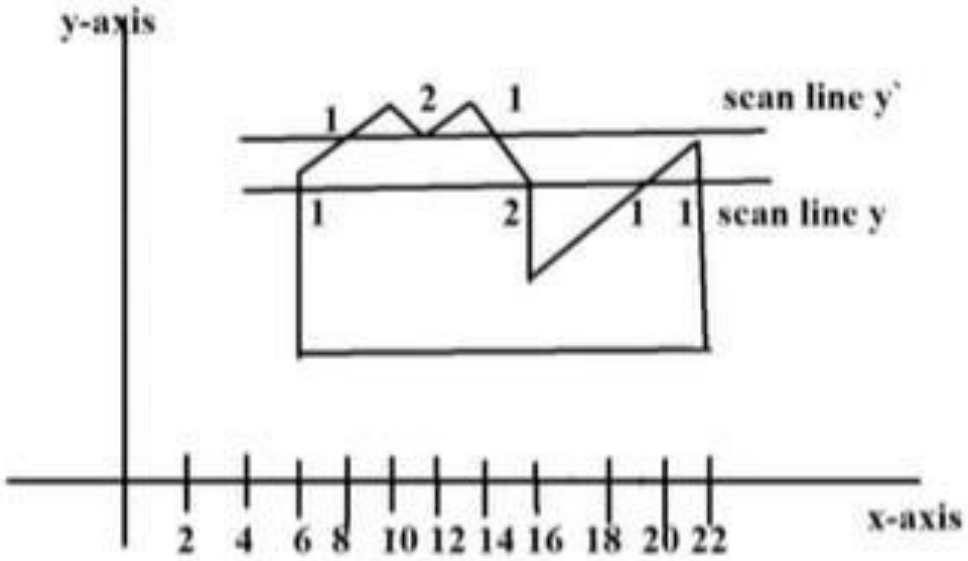
1. Find the intersections of scan line with all edges of the polygon.
2. Sort the intersections by increasing X co-ordinate i.e. from left to right. [6, 12, 16, 20]
3. Make pairs of the intersections and fill in color within all the pixels inside the pair. (6, 12) (16, 20)



Continue....

Special Cases (Intersection Point - 2)

- **Line y':** (8, 12) & (12, 14) (both edges – same side)
- **Line y:** (6, 16) & (19, 22) (both edges – different sides)

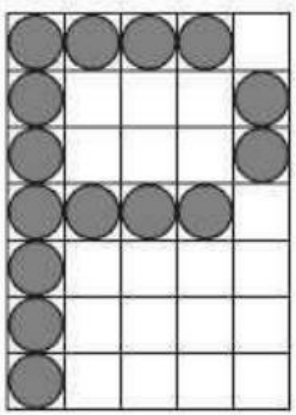


Scan-Converting a Character

- Character such as letters & digits are the building blocks of an image's textual contents.
- They can be presented in a variety of styles and sizes.
- The overall design style of a set of characters is referred to as its typeface or font.
- Font characteristics include: font-size, font-family, font-appearance.
- There are two basic approaches to character representation :
  - ✓ Bitmap Font
  - ✓ Outline Font

Bitmap Font

- Each character is represented by the on pixels in a bi- level pixel grid pattern called a bitmap.
- This approach is simple & effective since characters are defined in already-scan-converted form.
- Putting a character into an image basically entails a direct mapping or copying of its bitmap to a specific location in the image space.



Continue...

- Although one might generate variations in appearance and size from one font, the overall result tends to be less than satisfactory.
- The example in following figure shows that we may overlay bitmap in the previous figure onto itself with a horizontal offset of one pixel to produce **bold font**, and shift rows of pixels to produce **italic font**.

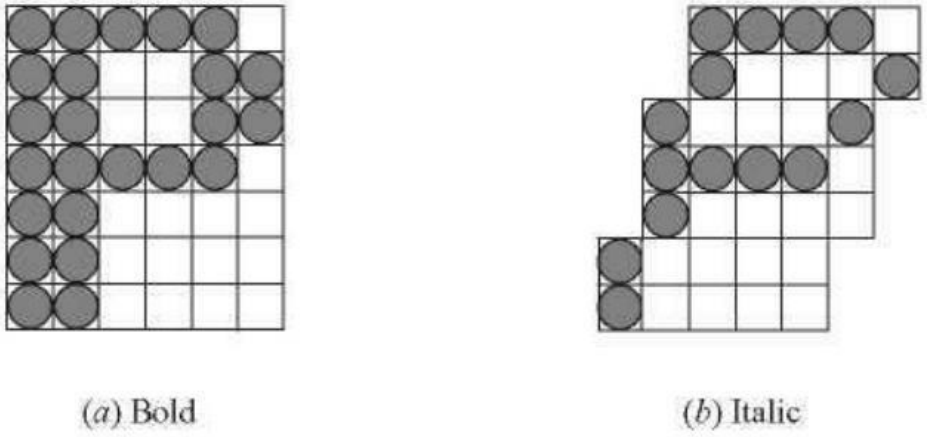


Fig. Generating variations in appearance

Outline Font

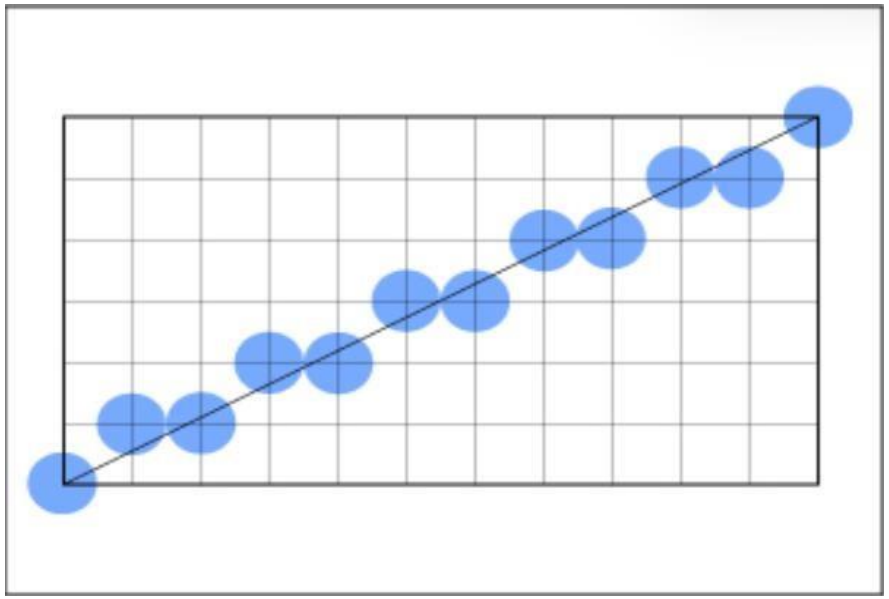
- Graphical primitives such as lines & arcs are used to define the outline of each character.
- Although an outline definition tends to be less compact than a bitmap definition, and requires relatively time- consuming operations, it can be used to produce characters of varying size, appearance and even orientation.
- The following outline definition can be resized through a scaling transformation, made into italic through a shearing transformation, and turned around with respect to a reference point through a rotation transformation.

Aliasing Effect

- Scan conversion is essentially a systematic approach to mapping objects that are defined in continuous space to their discrete approximation.
- The various forms of distortion that result from this operation are collectively referred as the aliasing effects of scan-conversion.
- Some effects are:
  - Staircase;
  - Unequal Brightness/Intensity;
  - The Picket Fence Problem;

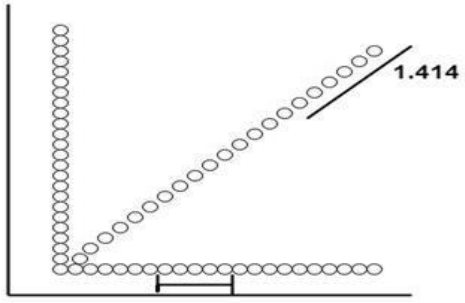
Staircase

- A common example of aliasing effects is the staircase or jagged appearance.
- It happens when primitives like line or circle are scan-converted.
- It also found sometimes along the border of a filled region.



Unequal Intensity

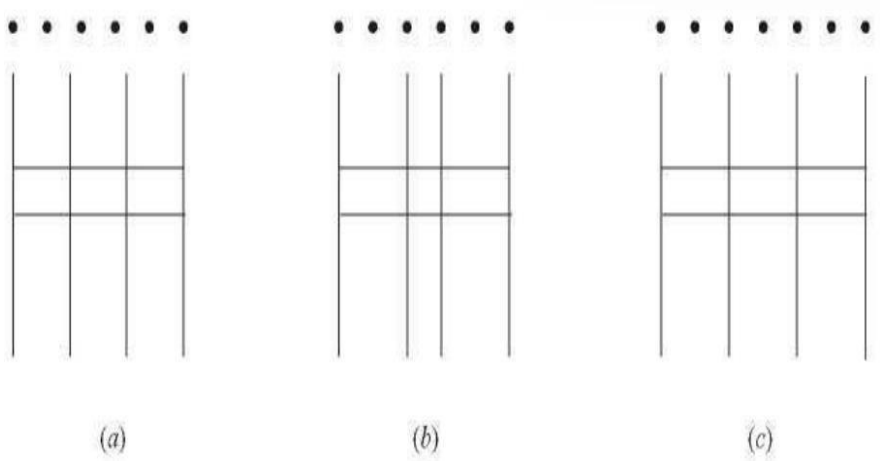
- It deals with unequal appearance of the brightness of different lines. A slanted line appears dimmer than a horizontal or vertical line, although all are presented at the same intensity level.
- The reason for this problem can be explained using Fig. where the pixels on the horizontal line are placed one unit apart, whereas those on the diagonal line are approximately 1.414 units apart. This difference in density produces the perceived difference in brightness.



Pixels along with horizontal line are 1 unit apart and vertical.
Pixels along diagonal line are 1.414 units.

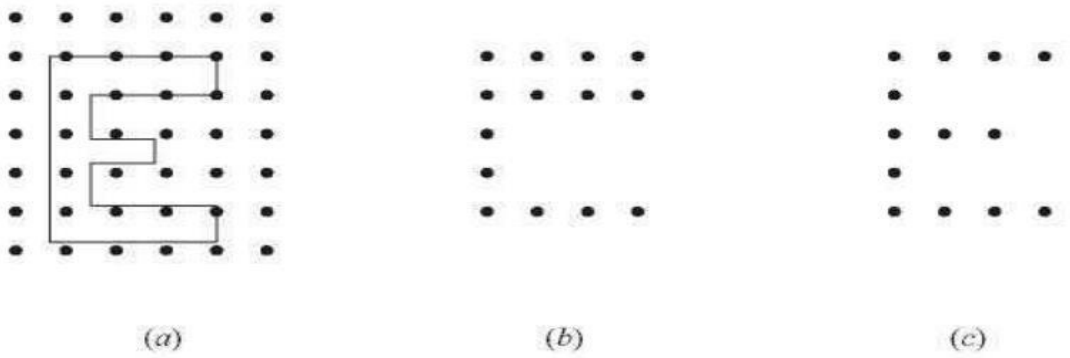
The Picket Fence Problem

- The picket fence problem occurs when an object is not aligned with, or does not fit into, the pixel grid properly.
- Figure (a) shows a picket fence where the distance between two adjacent pickets is not a multiple of the unit distance between pixels. Scan-converting it normally into the image space will result in uneven distances between pickets since the endpoints of the pickets will have to be snapped to pixel coordinates [Fig. (b)]. This is called **global aliasing**, as the overall length of the picket fence is approximately correct.
- On the other hand, an attempt to maintain equal spacing will greatly distort the overall length of the fence [Fig. (c)].
- This is sometimes called **local aliasing**, as the distances between pickets are kept close to their true distances.



Continue...

- Another example of such a problem arises with the outline font.
- Suppose we want to scan-convert the uppercase character "E" in Fig. (a) from its outline description to a bitmap consisting of pixels inside the region defined by the outline.
- The result in Fig. (b) exhibits both asymmetry (the upper arm of the character is twice as thick as the other parts) and dropout (the middle arm is absent).
- A slight adjustment and/or realignment of the outline can lead to a reasonable outcome [see Fig. (c)].

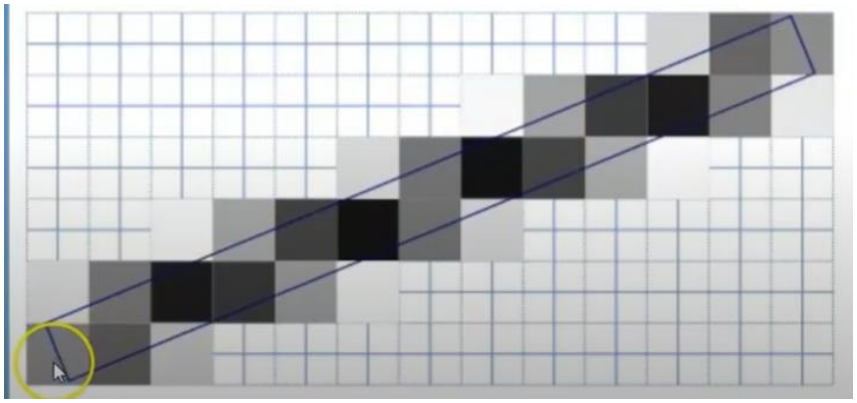


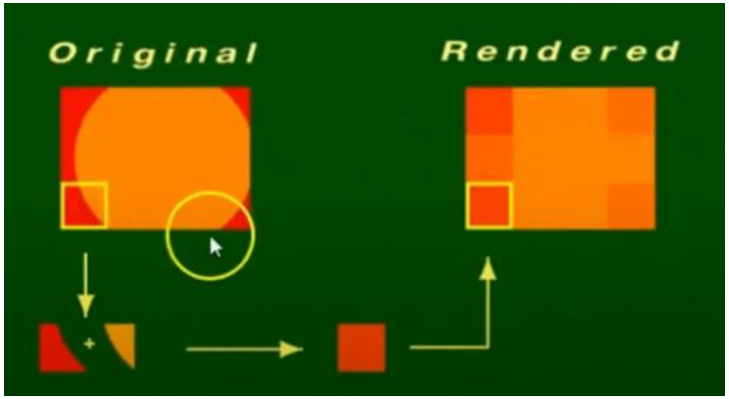
Anti-Aliasing

- Anti-aliasing is a technique used in computer graphics to reduce the visual artifacts known as aliasing, which appear as jagged edges on lines and curves. In some cases, aliasing are tolerable, in many cases, negligible.
- However, they can have significant impacts on our viewing experience when left untreated in a series of images that animate moving objects.
- Increasing image resolution is a straight-forward way of mitigating aliasing effects. But it is not always feasible.
- There are techniques, that can greatly reduce aliasing artifacts and improve the appearance of images without increasing resolution.
- These techniques are collectively called anti-aliasing technique.
    - ✓ Pre-filtering and post-filtering;
    - ✓ Area sampling;
    - ✓ Super-sampling;
    - ✓ Lowpass-filtering;
    - ✓ Pixel Phasing;

Pre-filtering and Post-filtering

- Pre-filtering: This method involves processing the image before it is sampled. It typically uses area sampling, where the color of a pixel is determined by averaging the colors of all samples within a certain area. This helps in capturing more information about the scene and results in smoother transitions between colors.





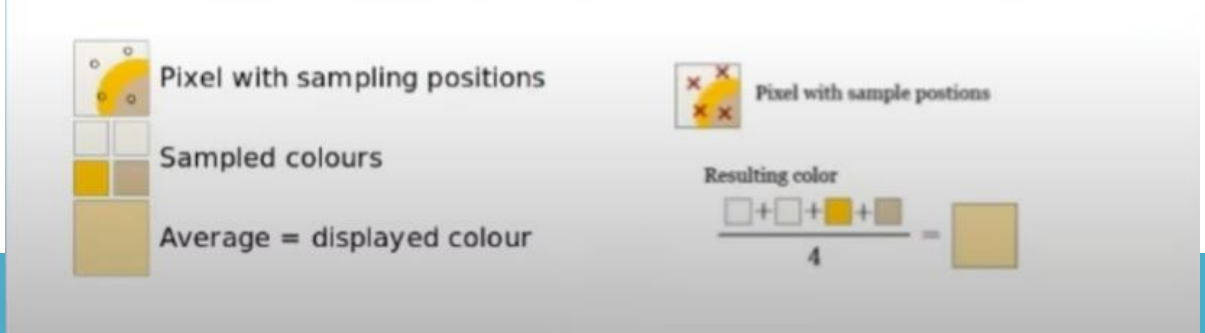
- Post-filtering: Also known as supersampling, this technique involves rendering an image at a higher resolution and then downsampling it to the desired resolution. The process averages multiple samples per pixel to reduce aliasing effects, resulting in smoother edges.

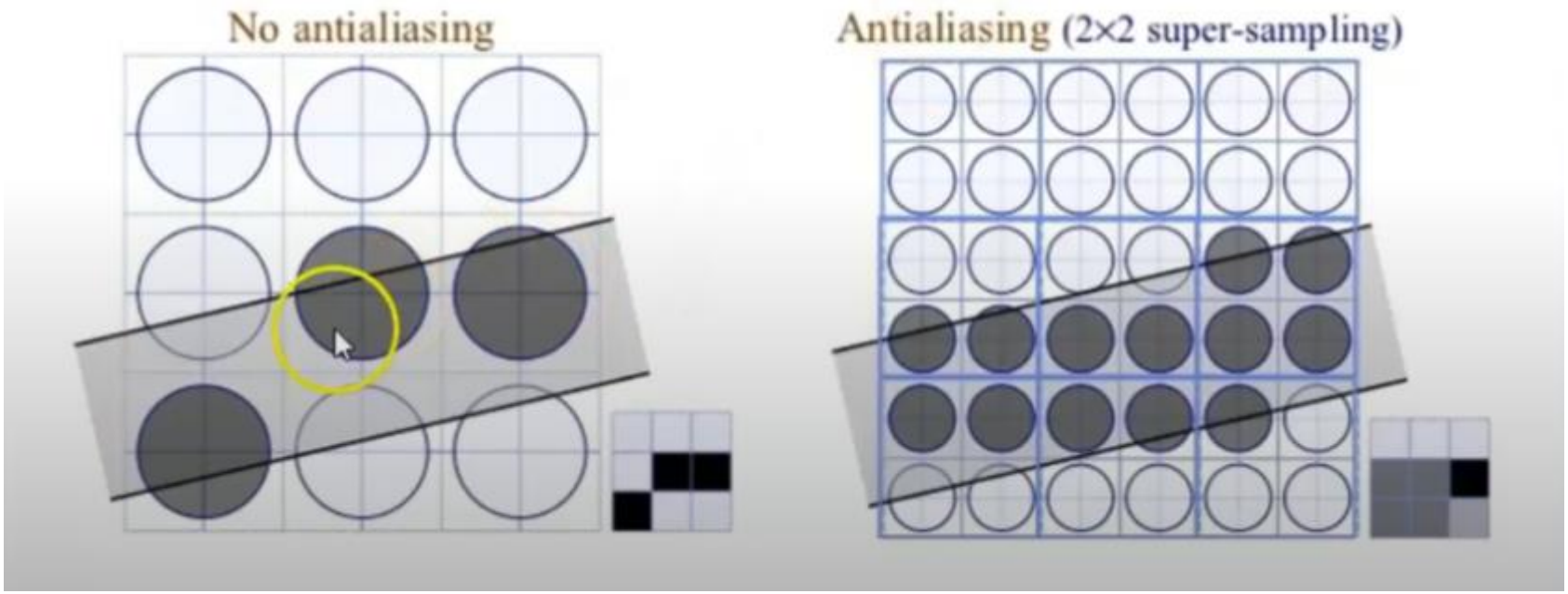
Area Sampling

Area sampling is a pre-filtering technique where each pixel's color is computed by averaging the colors of multiple samples taken from within the pixel's area. This method helps in reducing aliasing by considering more information about the scene before rendering

Super-sampling

- Supersampling, or Supersampling Anti-Aliasing (SSAA), is a high-quality anti-aliasing technique that involves rendering an image at a much higher resolution than required and then downsampling it to fit the display.
- This method effectively reduces aliasing by averaging multiple samples per pixel, resulting in smoother lines and improved visual fidelity.
- However, it is computationally intensive and may not be suitable for real-time applications due to its high resource demands.




No antialiasing
Antialiasing (2×2 super-sampling)

Lowpass filtering

- Lowpass filtering is a fundamental concept in signal processing. It involves removing high-frequency components from a signal while allowing low-frequency components to pass through.
- In the context of computer graphics, lowpass filtering can be applied to reduce high-frequency noise or aliasing artifacts.
- For example, when down sampling an image, you might apply a lowpass filter to avoid aliasing effects.

Pixel Phasing

Pixel phasing involves adjusting the position of pixels slightly to reduce aliasing effects. By shifting pixels to align more closely with object edges, this technique can help create smoother transitions between colors and reduce the appearance of jagged lines.

Thank You!