In this example, the circumference of the circle is the points whose $z^2 = r^2$ where $x^2 + y^2 - r^2 = 0$ for constants $r$ and $t$. A curve in the plane can be given implicitly by an equation. Example:

- Plotted, it looks like a line.
- Plotting the points with the resulting $(x', y')$ coordinates. If enough dots are evaluated for enough values of $x$ and $y$, you can draw a graph by plotting a function $y = f(x)$.
- Lines, curves, and other shapes are represented by sets of points.
- Origin, $x$-direction, $y$-direction perpendicular to the $x$-direction, and new geometrically.

Mathematically, once you specify a coordinate system of numbers called coordinates, points in the $(2d)$ plane or $(3d)$ space are represented by pairs or triples.

- Coordinate (analytic, cartesian) geometry
- Mathematics = Graphs + Raster Technology

CSI 422/502: Lecture 03
functions of \((\bar{h}', x)\).

What color we see there, Thus, the image results from a triple of numerical
the three intensities \(I^R\) (power) of the spots close to each other determines
the three colors \(R\), \textcolor{Green}{\text{G}}reen, and \(B\)lue are used. The combination of
spots. Three colors \(R\), \textcolor{Green}{\text{G}}reen, and \(B\)lue are used. The combination of

- Color images that appear on monitors are produced by glowing colored

Image.

According to the value of the function, the result is the drawing on an
another mathematical object that is useful for graphs is a function of
negative, the filled circle will be plotted.

\((\bar{h}', x)\) is different depending on whether this value is positive or
\((\bar{h}', x)\) is outside the circle. So, if a dot at
\((\bar{h}', x)\) is inside the circle, inside the circle.

The points with coordinates

+ \(Y\) are exactly the points inside the circle.

Write the Pythagorean theorem.

distance from the center is based on the meaning of
coordinates (\(h', x\)) make the equation true. The circle is the points whose
coordinates
\[
\begin{align*}
F(x, y) &= y - (Mx + B) \\
\text{We will see the above point (h, x) again.}
\end{align*}
\]
Other technologies: Plotters and vector displays are uncommon or obsolete.

of the display.

of the pixel at row $i$ and column $j$ determines the color

$$F[i][j] = \text{numeric data that determines the color}$$

Display or printer

Pixel

Scan Line

Buffer (or Refresh)

Frame

2-dimensional array

Computer, Graphics card or printer memory

Raster Technology

864 x 1152

863
Grille slots (not in book)

- dot pitch: Distance between adjacent 3-color triplets of phosphor dots on the monitor.
- printer hardware: It is a property of the monitor or mode of the graphics card. Precisely, it is 60% of the 60% criteria (see p. 38). Commonly, the resolution is determined by the maximum number of adjacent directions, e.g. 1152 by 864. Precisely, the maximum number of adjacent directions is the electron beam. How long the phosphor glows at least 10% since it was hit by the electron beam.

- resolution: Commonly, the numbers of pixels in the horizontal and vertical directions.
- persistence: How long the phosphor glows at least 10% since it was hit by the electron beam.

0.37

The ratio = number of scan lines

60-120 Hz. Typical values: 60-120 Hz.

- refresh rate: How many times per second (cycles per second, or hertz) Video display/restore scan terminates (HP 2-1, 2-2).
Graphics operations stay tuned.

Warning: Bit- and pixelmaps are used in tricky ways to speed up pixel-based manipulation of image.

Pixelmap: A 2-d array with more than one bit per pixel, typically for storing or effective resolution.

Bitmap: A frame buffer (or any other 2-d array) with one bit per pixel.

Depth or number of bit planes: For each pixel, how many bits are in the array that can be displayed in one frame. (But dithering makes more but reduces element in the frame buffer. 2d

Aspect ratio: (Height pixels are square, so size ratio equals this.)

\[
\text{aspect ratio} = \frac{\# \text{ pixel columns}}{\# \text{ pixel rows}}
\]
PCI bus, so at least 10,000 ns are needed per 32 bit transfer. Then a full
PCI bus takes (0.01 × 1000) = 10 seconds.

But this is very optimistic. Suppose the video card is on a (old) 100 MHz
PC. Suppose the refresh rate of the display is 80 Hz.

1. How many bytes of memory is used by the frame buffer of a 1152×864

Calculation problems:

University at Albany Computer Science Dept.

2. How many megabytes per second are transmitted from the frame buffer to the display when the refresh rate is 80 Hz.

3. Suppose 10 nanoseconds (1ns=10^-9=1 billionth sec.) are used to transfer 32 bits from a CPU to the frame buffer. How long will it take to fill the above frame buffer? What is the maximum number of refreshes per second, approximately?

Number of refreshes/second = 1/(0.010sec./refresh) = 100 refreshes/sec.

1152 × 864 × (10 nanoseconds) = 9.932 × 280ns = 0.010sec./refresh

second, approximately?
How does hardware retrieve, translate, and transmit frame buffer data to the display or printer? (1 whole frame per refresh).

(2) How do graphics applications store data into the frame buffer? (SW + network + GPU + hardware accelerator? HW?)

(1) How do graphics applications store data into the computer, graphics card or printer memory? 2-dimensional array

- Display or printer memory
- Display or printer
- Frame buffer (or refresh)
- Buffer (or refresh)
3. Rasterize the line: Compute which pixels to modify.

2. (In a modern system, transmit endpoint data to graphics device.)

1. Transform world coordinates to device (or pixel coordinates, for each end point.)

```
GetFlush();
End();
GetVertex2f(10, 145);
GetVertex2f(180, 15);
End glBegin(GL_LINES);
```

Begin to answer (1)...Given OpenGL line primitive, what happens?
always declare your variables with 

\[ v := \text{openGL vertices} \]

like \texttt{vertex} to describe a family of functions, each taking different kinds of arguments and types of the arguments. Documentation uses names.

\texttt{OpenGL TIPS:} Suffix numbers and letters of \texttt{OpenGL} function names.

\texttt{integer data also.}

Floating point (C/C++ float or double) is used, but \texttt{OpenGL} accepts range of \texttt{world coordinates}.

\underline{Important Opportunity:} You choose units, orientation, meaning, and useful world coordinates. Use to define places within the whole "world" model.

\texttt{World coordinates} are copied (after translation) into one or more places in the "world model". The master will be \texttt{master (or local or master) coordinates} used to define a template or

\texttt{Various coordinate systems and what they are used for:}

\section*{University at Albany Computer Science Dept.}
Calculations of normalized coordinates from world coordinates are independent of the # of pixels.

Defined so the range is [0,1] or [-1,1].

From world coordinates, use in the calculations of screen coordinates:

- Normalized coordinates
- Always unsigned integers
- From (0 to (#horiz. pixels)-1, 0 to (#vert. pixels)-1.
- Device or screen coordinates: 1 unit corresponds to one pixel. Range: 0
determine what part of the world will be viewed, and how the camera works.

GLMatrixMode(GL_PROJECTION); glOrtho(0.0, 200.0, 0.0, 150.0, 0.0, 1.0);

and user interaction to set this!

Serious applications will read the window size determined by the window system
coordinates to be used.

GLIntegerv(GL_WINDOW_SIZE, 400, 300); determines the ranges of screen/device

So, re. our first OpenGL program: