

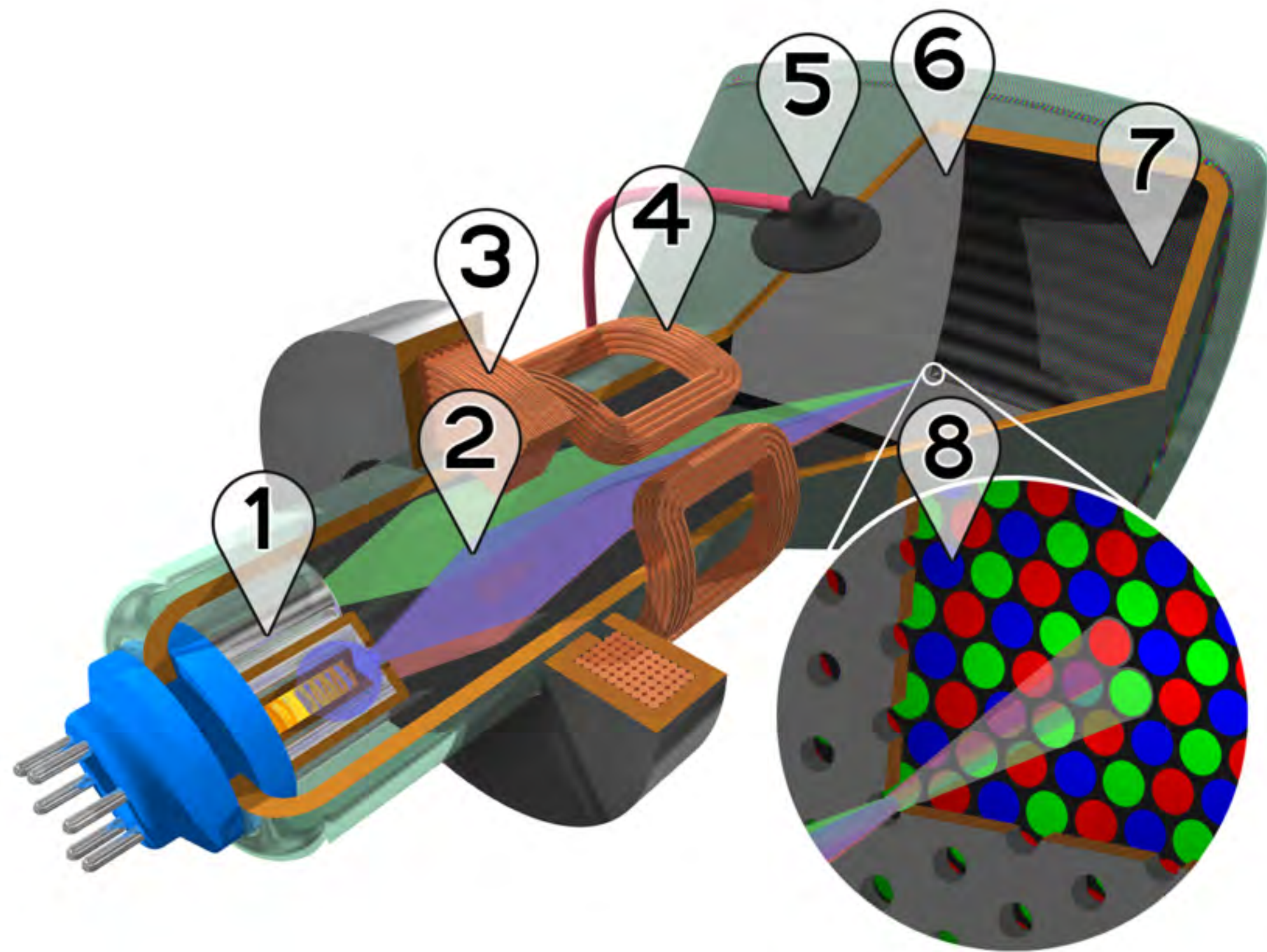
HARDWARE DE GRÁFICOS

Computación Gráfica

Dispositivos de despliegue de video

- Display de rayos catódicos (CRT)
- Display de cristal líquido (LCD)
- Display de plasma

CRT



1. Three electron guns (for red, green and blue phosphor dots)
2. Electron beams.
3. Focusing coils.
4. Deflection coils.
5. Anode connection.
6. Mask for separating beams for red, green, and blue part of displayed image.
7. Phosphor layer with red, green and blue zones.
8. Close-up of the phosphor-coated inner side of the screen.

Wikimedia commons

Tubo de rayos catódicos

- Los fotones se caracterizan por su color (r,g,b) y por su persistencia.
- **Persistencia:** tiempo en que la luz emitida decae al 10% de su intensidad inicial.
- Alta persistencia: **tasa de refrescado** (refresh rate) más baja para evitar **parpadeo** (flicker).
- Baja persistencia: buenos para animación pero requieren una tasa de refrescado mucho más alta para evitar parpadeo.
- Se necesita una tasa de refrescado de 50-60 Hz generalmente para evitar parpadeo, algunos sistemas refrescan a tasas de hasta 72-76 Hz.

<http://www.siggraph.org/education/materials/HyperGraph/hardware/hardware.htm>

Características de un monitor

- El usuario puede variar el voltaje en la rejilla de control para atenuar el flujo de electrones (botón de intensidad).
- **Resolución:**
 - máximo número de puntos que se pueden desplegar sin superponerse.
 - dado por el número de puntos (pixels o picture elements) horizontales vs. el número de puntos verticales. ej. uno monitor de 1024x768 pixels.
- **Relación de aspecto** (aspect ratio): tasa de los pixels verticales entre los pixels horizontales para una línea de misma longitud.
- Tamaño físico de la pantalla se mide en la diagonal.

<http://www.siggraph.org/education/materials/HyperGraph/hardware/hardware.htm>

Características de un monitor

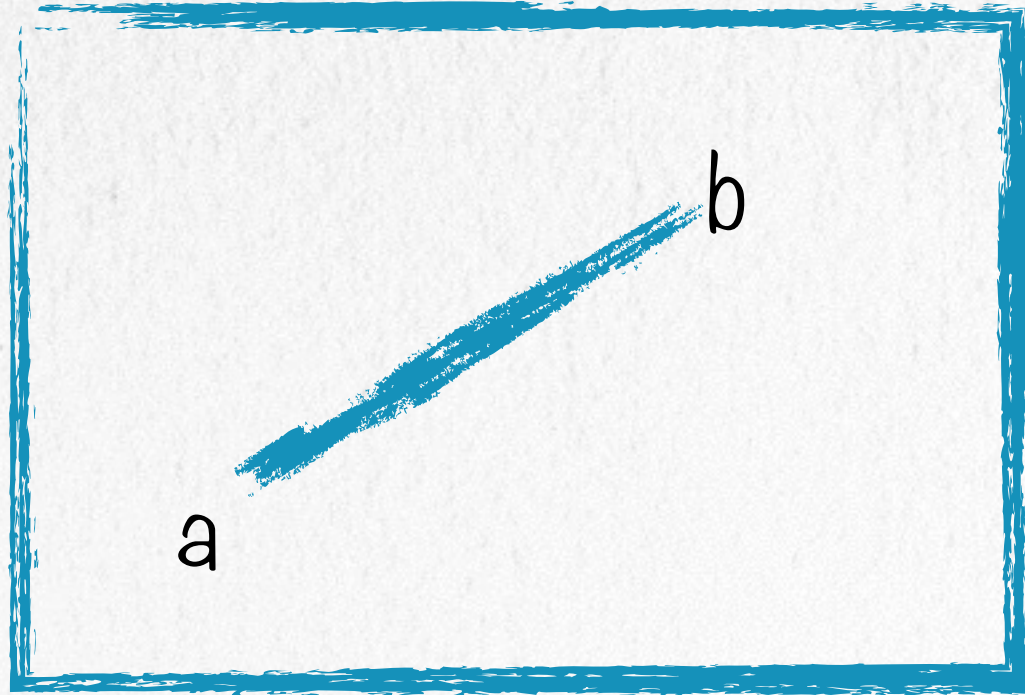


$$\text{aspect ratio} = 8/6 = 1.33$$

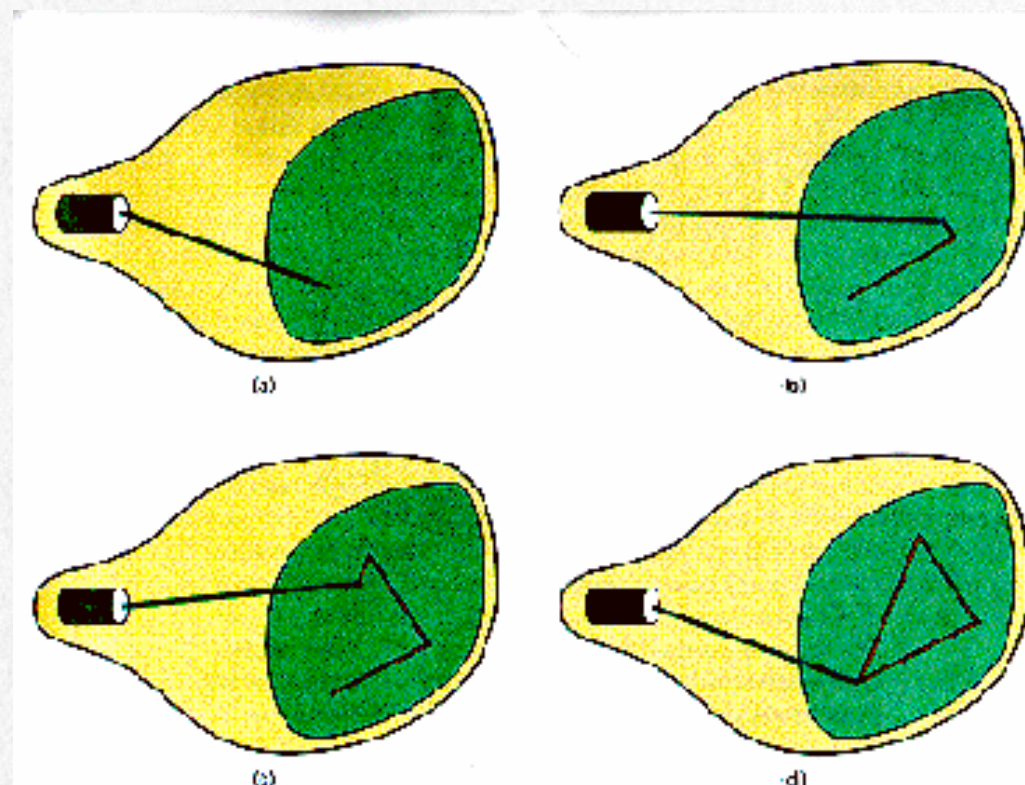
- Para este monitor y una resolución de 640x480 pixels:
- horizontal: $640/8 = 80$ pixels por pulgada.
- vertical: $480/6 = 80$ pixels por pulgada.
- En caso que los pixels horizontales no sean del mismo tamaño que los verticales hay que corregir para evitar que las imágenes se distorsionen.

<http://www.siggraph.org/education/materials/HyperGraph/hardware/hardware.htm>

Scan aleatorio (random scan) o vectorial.



1. Apagar el haz de electrones, moverse a "a".
2. Encender el haz, ir a "b" dibujando la recta.
3. Repetir.



- El flujo de electrones dibuja directamente la imagen.

- Ventajas:

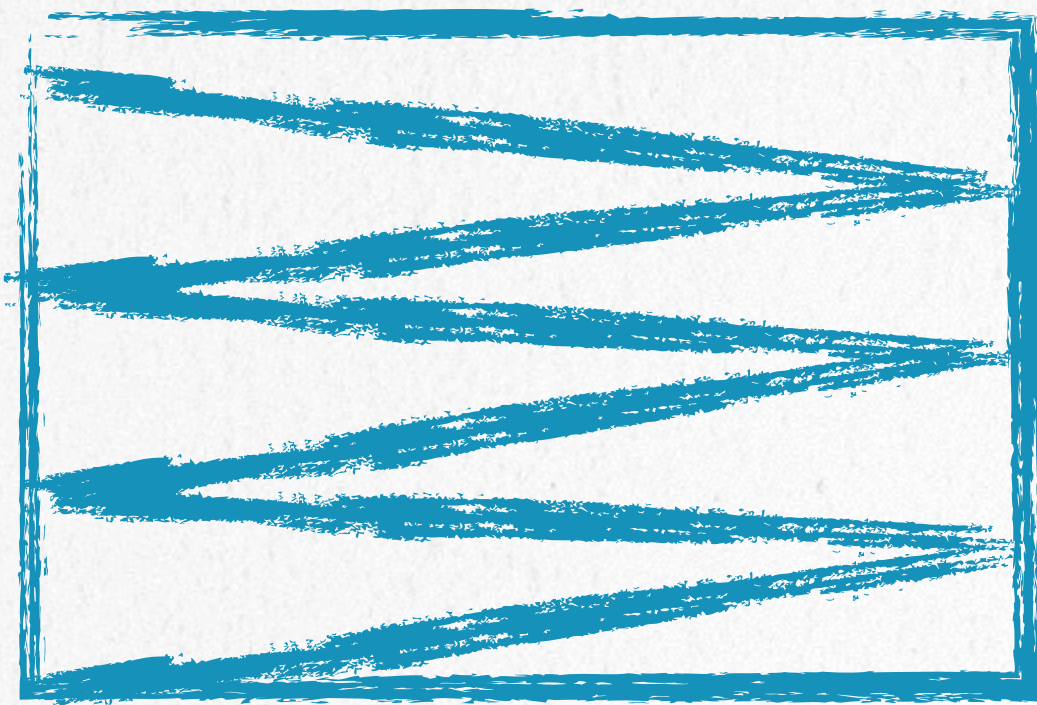
- > muy alta resolución, limitada solamente por el monitor.
- > fácil de animar, sólo dibujar en diferentes posiciones.
- > requiere poca memoria (lo suficiente para guardar el programa de despliegue).

- Desventajas:

- > requieren un rayo de electrones "inteligente", controlado por un procesador.
- > densidad limitada en la pantalla, no se pueden dibujar imágenes complicadas.
- > capacidad limitada de color (muy caro).

<http://www.siggraph.org/education/materials/HyperGraph/hardware/hardware.htm>

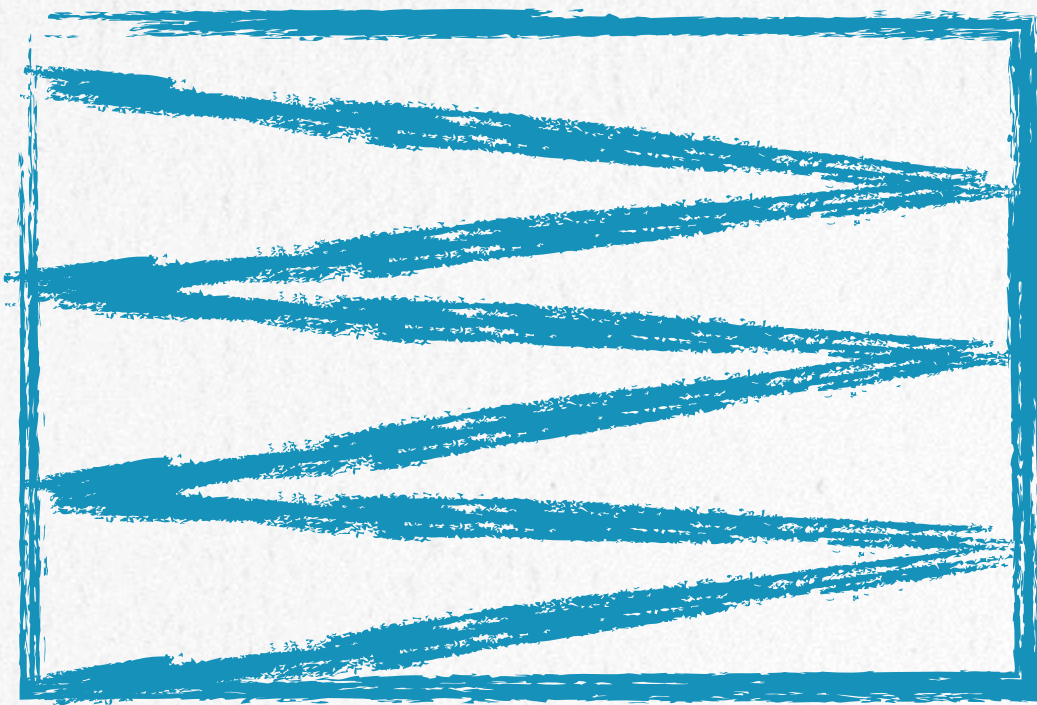
Raster Scan



- Hace un recorrido de la pantalla, de arriba hacia abajo con un patrón regular.
- Muy común en las televisiones.
- El haz de electrones se apaga y enciende para dibujar los puntos requeridos en la pantalla.
- Una trama o **raster** es una matriz de pixels cubriendo la pantalla.
- ¿Cómo decir cuando se enciende o apaga el haz?
- Almacenar el patrón de intensidades en una región de memoria gráfica llamado **frame buffer** (también **bit map**) donde cada localidad de memoria corresponde a un pixel.

<http://www.siggraph.org/education/materials/HyperGraph/hardware/hardware.htm>

Raster Scan



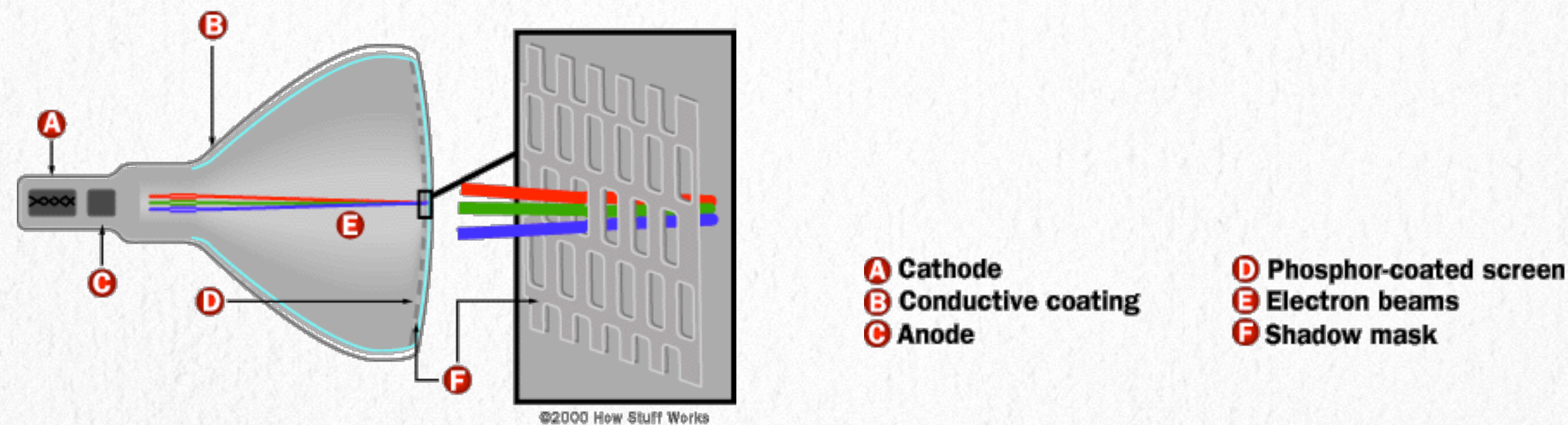
- El rango de colores que puede desplegar un sistema de raster depende de:
 - tipo de fósforo en el CRT.
 - número de bits disponible en el frame buffer.
 - e.j. un sistema blanco y negro puede requerir un solo bit para controlar su intensidad.
 - 24 bits por pixel en sistemas de alta calidad.
 - Un sistema con 24 bits por pixel y resolución de pantalla de 1024×1024 requiere 3 Mb de almacenamiento para el frame buffer.

Raster scan

- El número de bits por pixel en el frame buffer se llama **profundidad** (depth) del área del buffer o número de bit planes.
- Un frame buffer con 1 bit por pixel se llama **bitmap**.
- Un frame buffer con múltiples bits por pixel se llama **pixmap**.
- Los términos bitmap y pixmap también se usan para otros arreglos rectangulares que almacenan valores binarios o patrones multicolor.

CRT a color

- Despliegan imágenes a color usando una combinación de fósforos que emiten luces de diferentes colores.
- La luz emitida por los fósforos se combina para formar un solo color percibido.
- Monitores RGB de 24 bits por pixel en el frame buffer, permitiendo 256 ajustes de voltaje para cada rayo de electrones, almacena cerca de 17 millones de colores para cada pixel.
- Full-color system o True-color system.

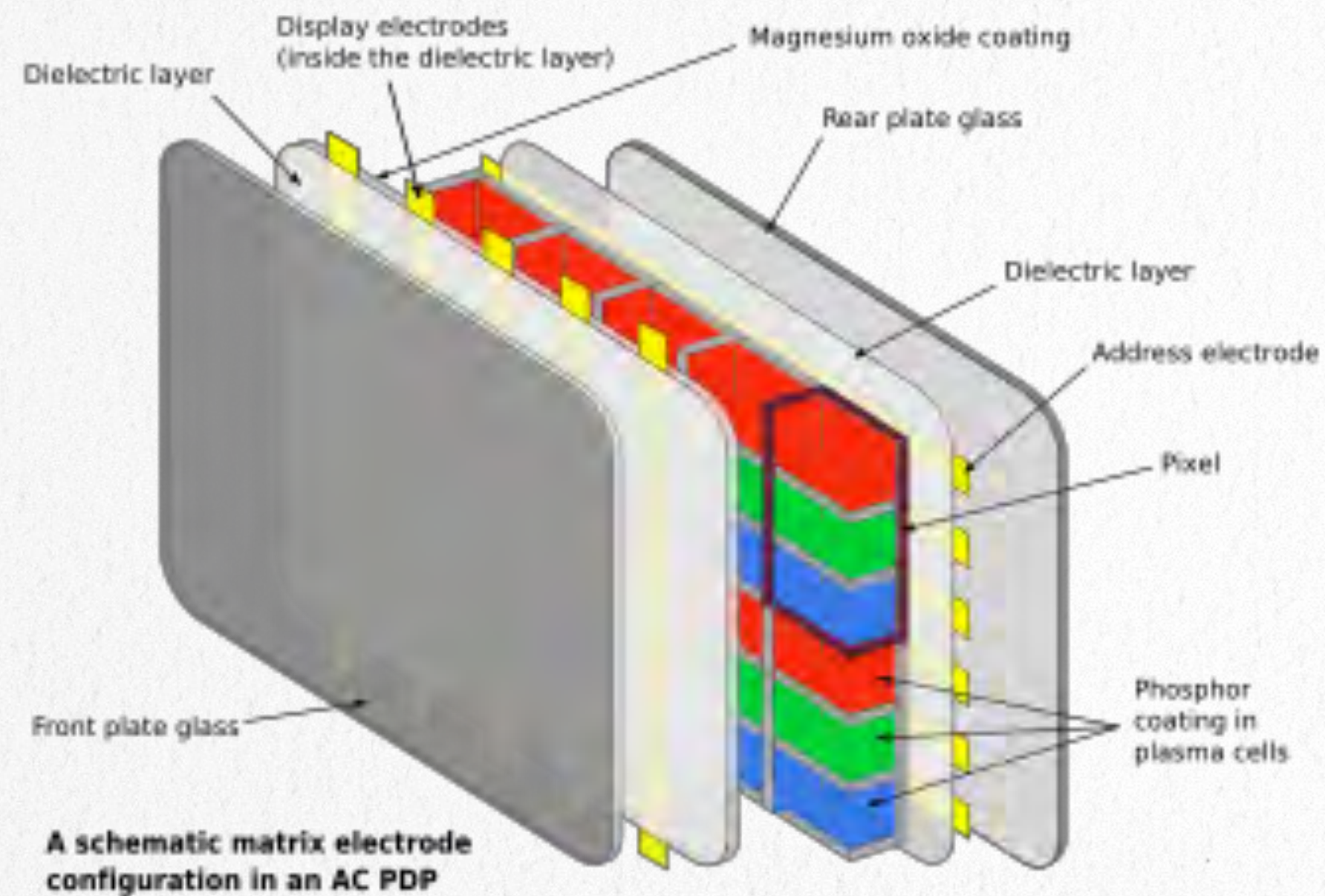


Flat panel displays



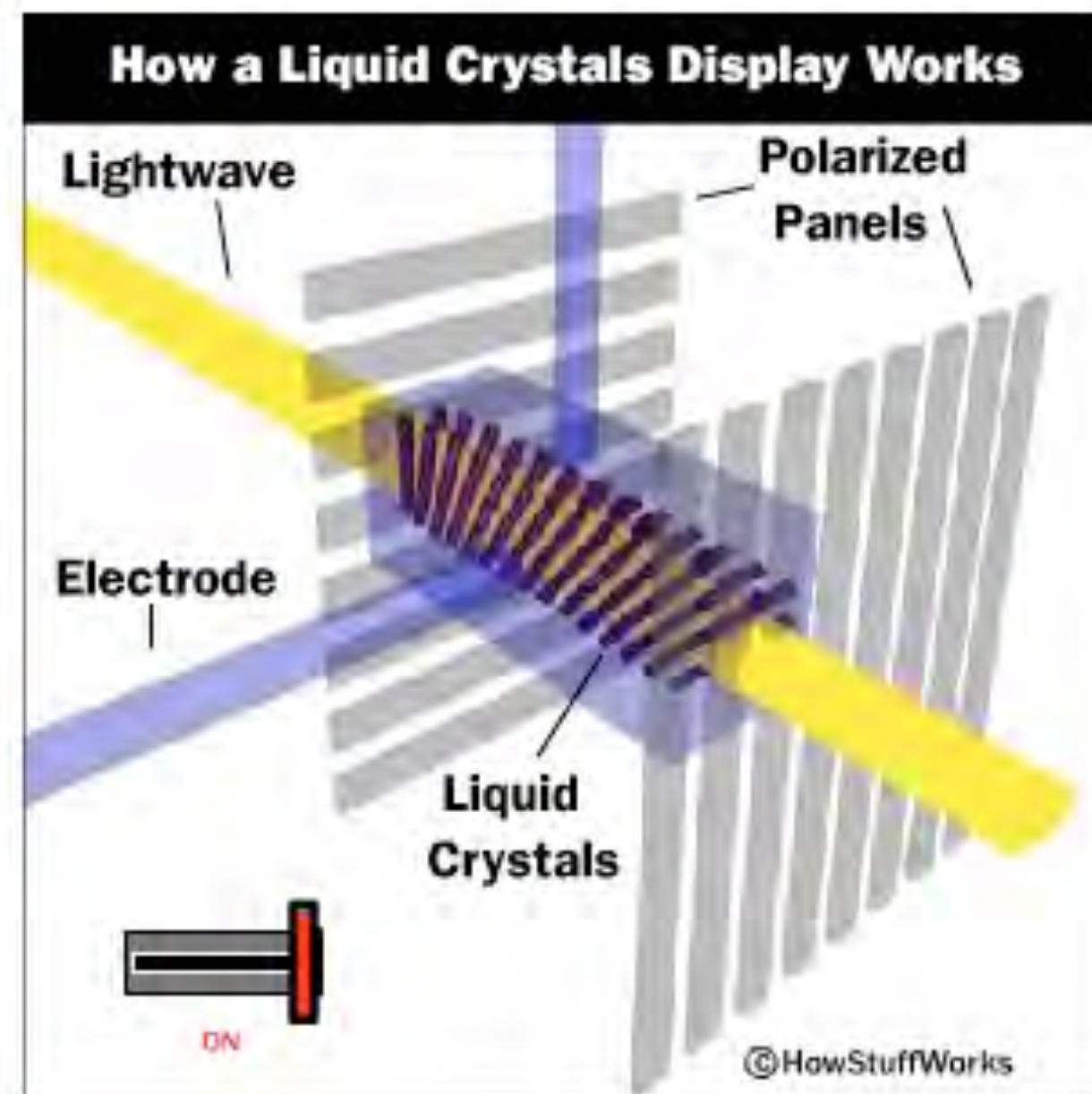
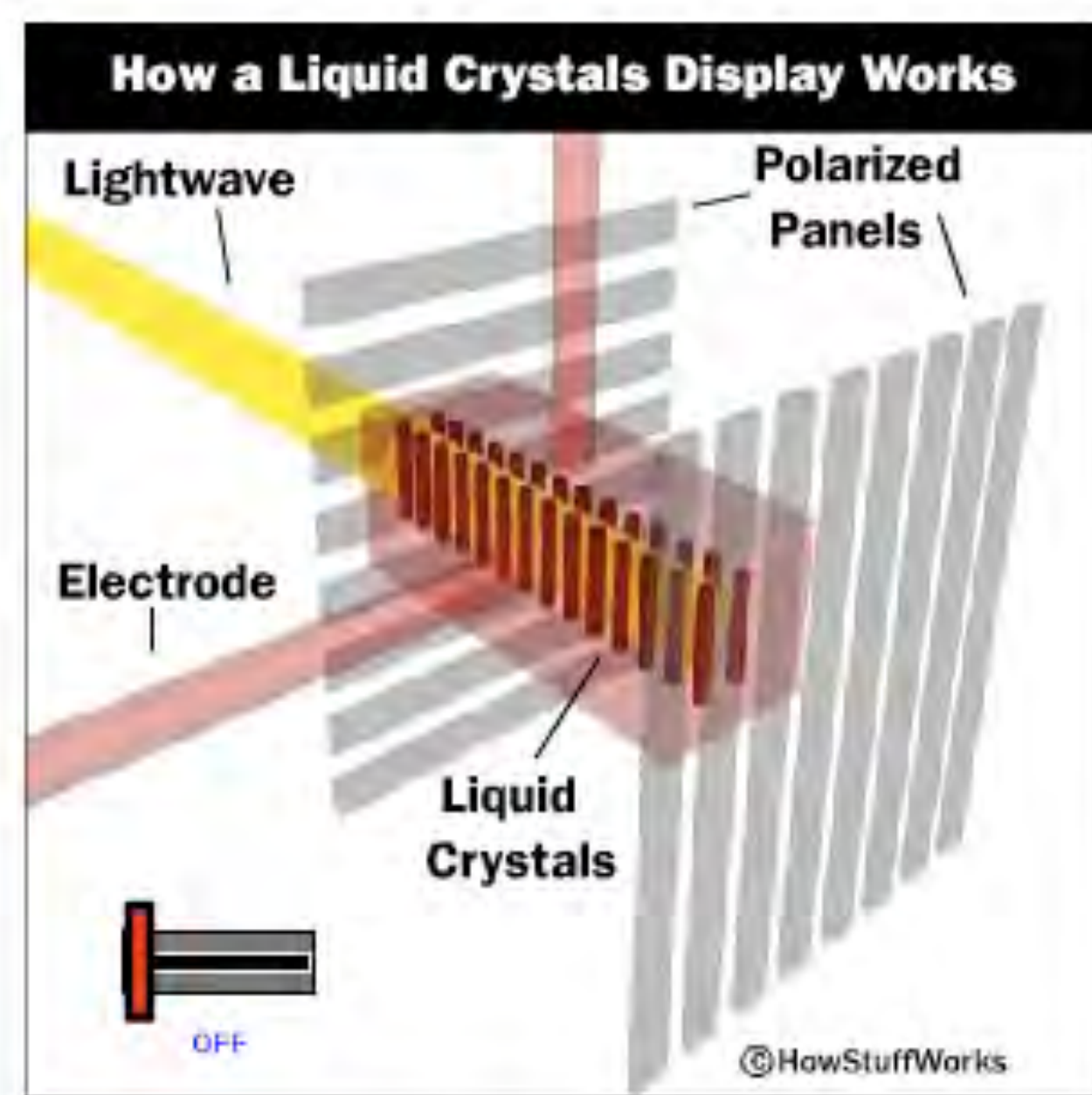
- Displays emisores: convierte la energía eléctrica en luz (monitores de plasma, monitores LEDs ...).
- Displays no emisores: utilizan efectos ópticos para convertir la luz solar o de alguna otra fuente en patrones gráficos (LCD ...)

Plasma panels



- Se construyen llenando la región entre dos placas de vidrio con una mezcla de gases que incluyen normalmente el neón.
- Se coloca una serie de conductores verticales en una placa y horizontales en la otra placa de vidrio.
- Se aplica voltaje a la intersección de dos conductores convirtiendo el gas en plasma.

LCD panels



- Dejan pasar luz polarizada de una fuente de luz a través de un material de cristal líquido que se puede alinear para bloquear o dejar pasar la luz.
- Las moléculas que no están alineadas tuercen la luz.



ALGORITMOS DE GRÁFICAS EN RASTER

Computación Gráfica

Imágenes discretas

- Un monitor de computadora se compone de pixels.
- Una imagen consiste generalmente de vistas proyectadas de modelos 3D que consisten de primitivas como puntos, líneas, círculos y polígonos rellenos.
- Las primitivas se definen en el espacio continuo con un nivel de abstracción más alto.
- El sistema gráfico tiene que convertir estas primitivas de su definición geométrica al conjunto de pixels en el espacio de imagen.
- “Scan conversion” o “Rasterization”

Convirtiendo un punto

- Una imagen (bitmap) en niveles de gris puede almacenarse en memoria en forma de matriz:

```
unsigned char image[xmax][ymax];
```

- Cada coeficiente `image[x][y]` representa el color del pixel (x,y) en nivel de gris entre 0 y 255, almacenado en 1 octeto.
- Un punto definido con coordenadas reales dentro del área de la imagen con coordenadas (x,y) se convierte a la posición de un pixel (x',y') .
- $x' = \text{Floor}(x)$, $y' = \text{Floor}(y)$
- Esta operación esencialmente pone el origen de un sistema coordenado continuo para (x,y) en la esquina izquierda inferior de la rejilla de pixels del espacio de imagen.

Convirtiendo un punto

- Otra opción es alinear los valores enteros del sistema coordenado de (x,y) con las coordenadas del pixel.
- (x,y) se mapea a $x'=\text{Floor}(x+0.5)$ y $y'=\text{Floor}(y+0.5)$
- Coloca el origen del sistema coordenado para (x,y) al centro del pixel $(0,0)$.
- En lo sucesivo utilizaremos esta segunda opción: pixels centrados en los valores enteros del sistema coordenado continuo.

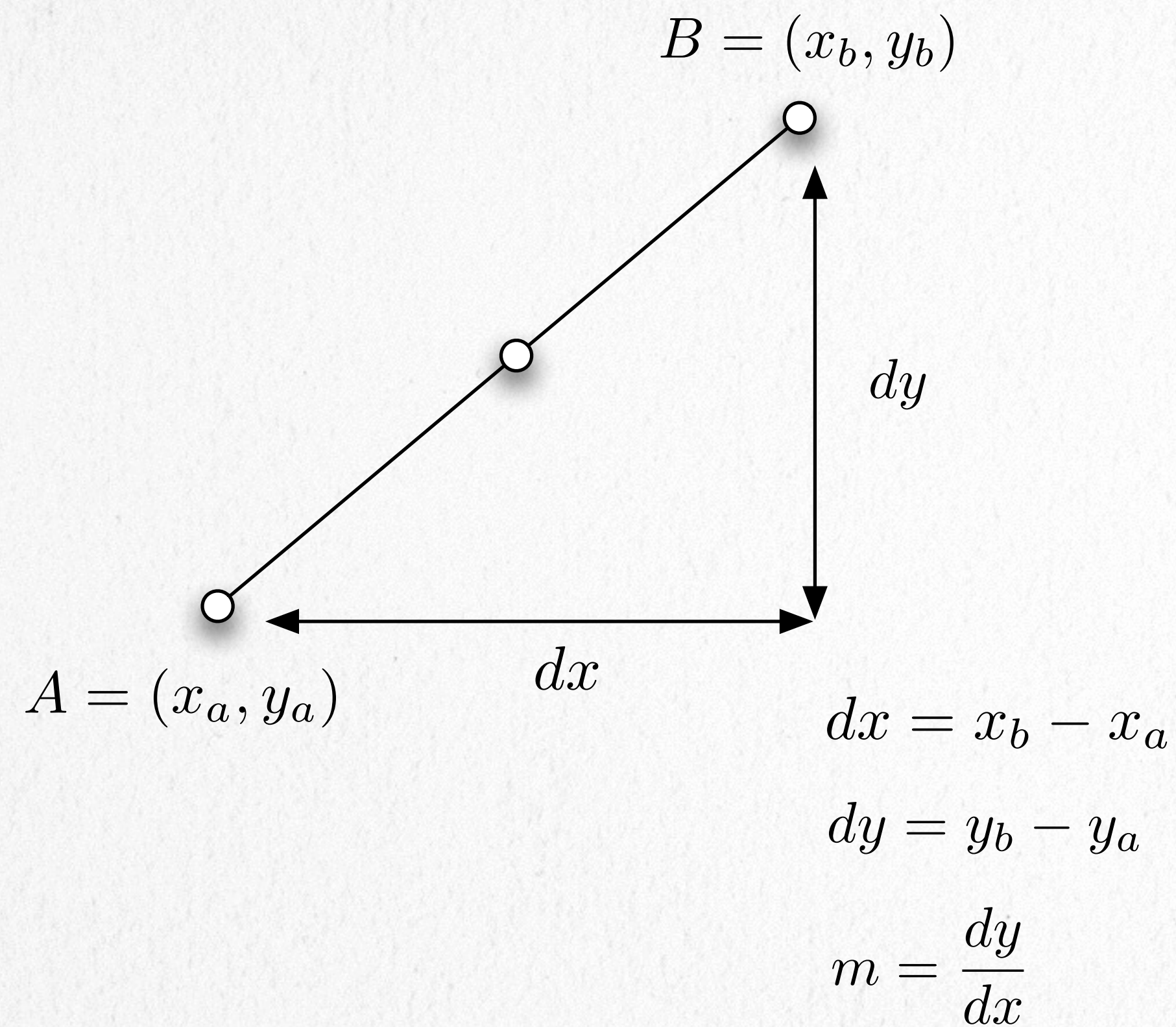
Segmentos de recta

- Los objetos geométricos en una imagen son aproximaciones discretas de ideales continuos.
- Se necesitan métodos eficientes para pasar de los objetos ideales a la discretización con la que será representada la imagen.
- Suponemos tener una primitiva $\text{PutPixel}(x,y, \text{color})$ que permite dibujar el pixel (x,y) del color en RGB de nuestra elección.
- Queremos dibujar segmentos de recta.
- Nos limitaremos a dibujar segmentos de grosor 1.
- Tomaremos solo pendientes entre -1 y 1 , las demás las calcularemos por simetría.

Convirtiendo una recta

- Conversión directa utilizando la ecuación de la recta.
- P_1 y P_2 con coordenadas (x_1', y_1') y (x_2', y_2') .
- $m = (y_2' - y_1') / (x_2' - x_1')$ y $b = y_1' - mx_1'$.
- Para cada valor entero de x entre (y excluyendo) x_1' y x_2' calcular el valor correspondiente de y usando la ecuación y discretizando y_1' .
- Si $|m| > 1$ para cada valor entero de y entre (y excluyendo) y_1' y y_2' calcular el valor correspondiente de x usando la ecuación y convertir (x, y) .
- Involucra operaciones de punto flotante (multiplicaciones y sumas) en cada paso que utilice a la ecuación ya que m y b son generalmente números reales.

Segmentos de recta



- Consideramos dos puntos distintos $A=(x_a, y_a)$ y $B=(x_b, y_b)$ en el plano.
- Por estos puntos pasa una sola recta que llamaremos D.
- Notamos $d_x = x_b - x_a$ y $d_y = y_b - y_a$ las diferencias entre las coordenadas de A y B.
- Suponemos que d_x es diferente a cero (la recta no es vertical).
- La pendiente de la recta D es $m = d_y / d_x$.

Segmentos de recta

- Punto en una recta:

$$F(x, y) = 0, \text{ con } F(x, y) = dy(x - x_a) - dx(y - y_a)$$

- Sea $M=(x_m, y_m)$ un punto en el plano

- Encontrar un punto a la izquierda o a la derecha de la recta:

$F(x_m, y_m) < 0$, izquierda del segmento $[A, B]$ dirigido de A a B

$F(x_m, y_m) > 0$, derecha del segmento $[A, B]$ dirigido de A a B

Segmentos de recta

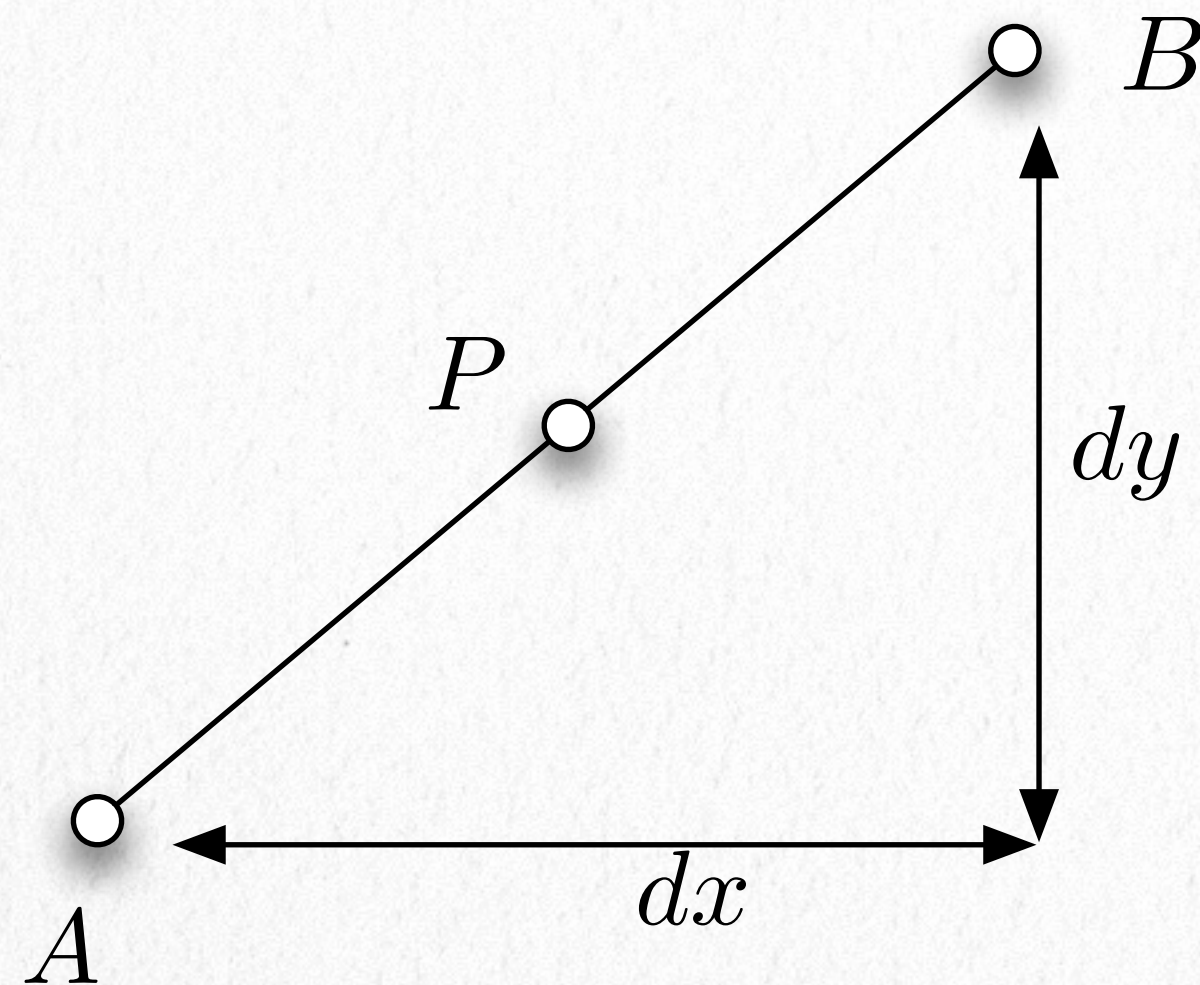
- Sea un punto $P=(x_p, y_p)$ de la recta R y que tenga por abscisa:

$$x_p = x_a + \Delta x$$

$$dx = x_b - x_a$$

$$dy = y_b - y_a$$

$$m = \frac{dy}{dx}$$



Segmentos de recta

- Sustituyendo en la ecuación

$$F(x, y) = 0, \text{ con } F(x, y) = dy(x - x_a) - dx(y - y_a)$$

$$0 = F(x_p, y_p)$$

$$= d_y(x_p - x_a) - d_x(y_p - y_a)$$

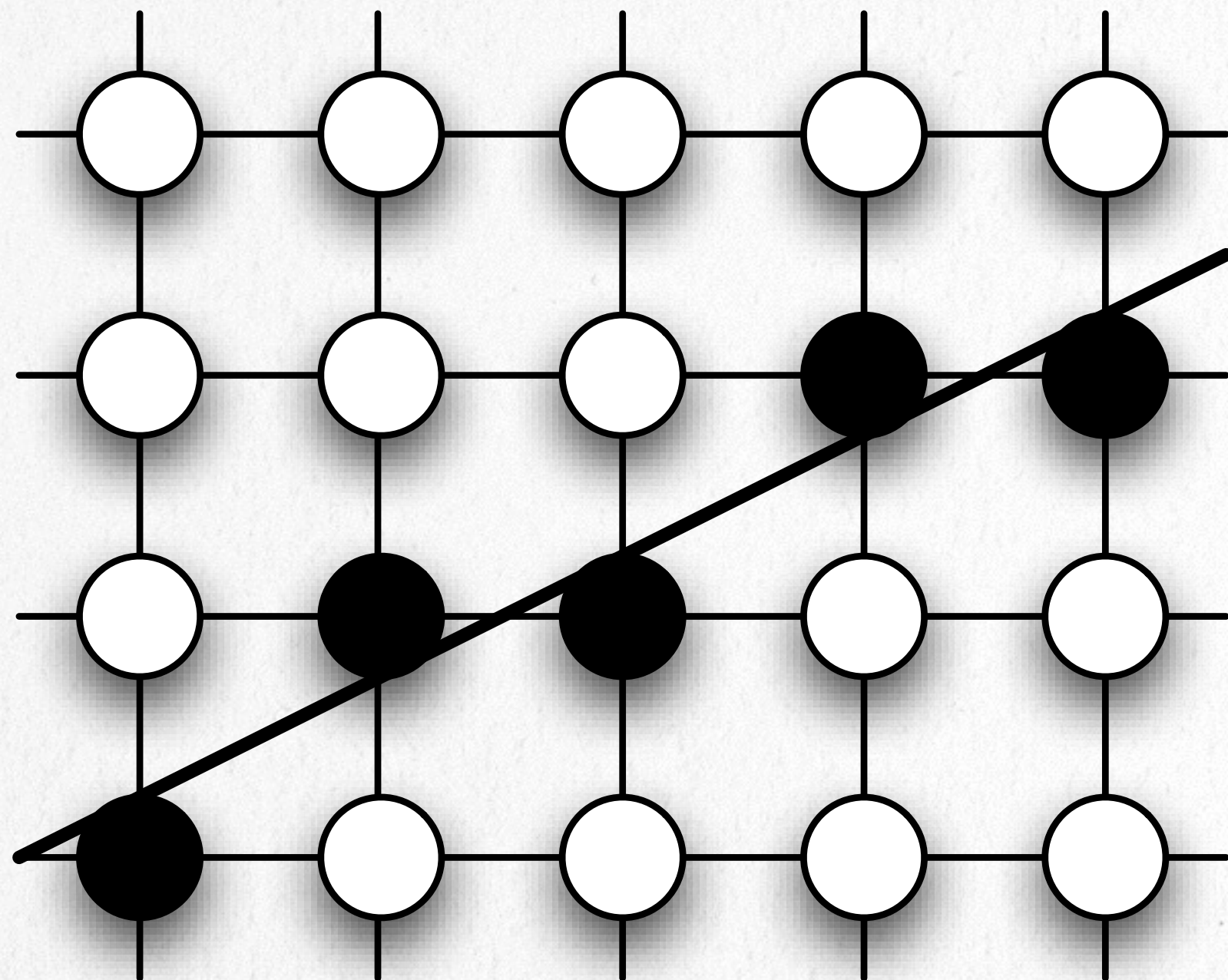
$$= d_y(x_a + \Delta x - x_a) - d_x(y_p - y_a)$$

- De donde obtenemos: $dy \cdot \Delta x = dx(y_p - y_a)$

$$y_p = y_a + \frac{dy}{dx} \cdot \Delta x$$

$$y_p = y_a + m \Delta x$$

Segmentos de recta



- La secuencia de pixels debe estar lo más cerca posible de la línea “ideal”.
- Consideramos una línea de 1 pixel de ancho.
 - si la pendiente está entre -1 y 1 , se debe iluminar un pixel por cada columna.
 - si la pendiente está fuera de ese rango, un pixel debe estar iluminado en cada renglón.
- todas las líneas deben dibujarse con brillo constante, independientemente de su longitud y orientación y lo más rápido posible.
- preveer otros anchos, tipos de pluma, tipo de puntos extremos, etc.

Convirtiendo una recta: Algoritmo DDA

- Digital Differential Analyzer (DDA).
- Método incremental de conversión de scan.
- Supongamos que en el paso i calculamos (x_i, y_i) como un punto sobre la recta.
- El siguiente punto (x_{i+1}, y_{i+1}) debe satisfacer $D_y/D_x = m$ donde $D_y = y_{i+1} - y_i$ y $D_x = x_{i+1} - x_i$.
- **Ventaja:** no hay multiplicaciones con punto flotante (si hay sumas)
- **Desventaja:** el error acumulado lleva al segmento a alejarse de su posición real cuando éste es largo.

Convirtiendo una recta: Algoritmo DDA

● Tenemos:

$$y_{i+1} = y_i + mD_x \qquad x_{i+1} = x_i + \frac{D_y}{m}$$

● Cuando $|m| \leq 1$ inicializamos con $x=x_1'$ (suponiendo que $x_1' < x_2'$) y $y=y_1'$.

● $D_x = 1$ (incremento unitario en dirección x).

● Calculamos y de la siguiente forma: $y_{i+1} = y_i + m$.

● Cuando $|m| > 1$ inicializamos con $x=x_1'$ y $y=y_1'$ (suponiendo que $y_1' < y_2'$).

● $D_y = 1$ (incremento unitario en dirección y).

● Calculamos x : $x_{i+1} = x_i + 1/m$

● Continuar hasta llegar a x_2' para $|m| \leq 1$ o a llegar a y_2' para $|m| > 1$.