

Computación gráfica y ambientes virtuales (0494)

Guía de prácticos n° 1

Digitalización de líneas

Practico 1

Extienda el algoritmo de líneas de Bresenham para poder graficar líneas de cualquier pendiente (excepto horizontales y verticales) y con cualquier orden de los puntos inicial y final de la recta.

Sugerencia: genere un programa que tome como argumentos los puntos inicial y final de la línea y genere como salida un archivo de imagen con formato ppm.

Practico 2

Usando los algoritmos de digitalización para líneas se pide lo siguiente:

1. Compare los resultados que se obtiene para digitalizar una misma recta usando el algoritmo DDA (Digital Differential Analyzer) y el algoritmo de Bresenham
2. Genere un programa que grafique dos rectas con las misma pendiente pero diferentes punto inicial a un archivo de imagen con formato ppm usando los algoritmos DDA y de Bresenham

Digitalización de circunferencia y elipses

Practico 3

Extienda el algoritmo de punto medio para circunferencias en el segundo octante para que sea posible graficar una circunferencia completa (360°).

Sugerencia: genere un programa que tome como argumentos un punto (x,y) correspondiente al centro de la circunferencia y el radio de la misma y genere como salida un archivo de imagen con formato ppm.

Practico 4

Considerando el algoritmo de punto para graficar elipses en la “posición estándar”, demuestre que los incrementos de las variables de decisión para la región 1 y para la región 2 son:

Región 1:

$$\begin{aligned}((\Delta D_k)_{ady})_1 &= 4r_y^2(2x_k + 1) + 8r_y^2 \\ ((\Delta D_k)_{aInf})_1 &= 4r_y^2(2x_k + 1) + 8r_y^2 + 2r_x^2(1 - 4y_k) + 6r_x^2\end{aligned}$$

Región 2:

$$\begin{aligned}((\Delta D_k)_{ady})_2 &= 4r_x^2(1 - 2y_k) + 8r_x^2 \\ ((\Delta D_k)_{aInf})_2 &= 2r_y^2(4x_k + 1) + 6r_y^2 + 4r_x^2(1 - 2y_k) + 8r_x^2\end{aligned}$$

Practico 5

Extienda el algoritmo de punto medio para elipses en “posición estándar” para el primer cuadrante para que sea posible graficar una elipse completa (4 cuadrantes).

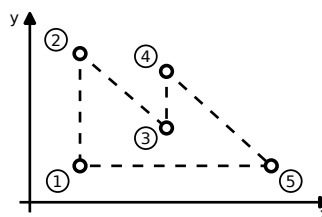
Sugerencia: genere un programa que tome como argumentos un punto (x,y) correspondiente al centro de la elipse y los radios del eje mayor y el eje menor de la misma y genere como salida un archivo de imagen con formato ppm.

Relleno de áreas

Practico 6

Genere un programa que implemente el algoritmo de relleno de polígonos por medio del algoritmo optimizado de “scanline”. Utilice los siguientes vértices para generar una imagen con un polígono relleno en un archivo de imagen ppm.

#	x	y
1	10	10
2	10	50
3	40	20
4	40	40
5	70	10



Practico 7

Grafique los siguientes conjuntos de vértices mediante el algoritmo optimizado de “scanline” para rellenos de polígonos. ¿A qué polígonos corresponde cada uno de los conjuntos de vértices dados?

#	x	y
1	80	50
2	130	140
3	30	40

#	x	y
1	200	100
2	250	150
3	230	200
4	170	200
5	150	150

#	x	y
1	210	70
2	260	120
3	310	95
4	300	70
5	310	45
5	260	20

Antialiasing

Practico 8

Aplicar la técnica de suavizado por supersampling con un nivel de resolución de 3x para dibujar líneas mediante el algoritmo de Bresenham.

Sugerencia: genere un programa que tome como argumentos los puntos inicial y final de la línea y genere como salida un archivo de imagen con formato ppm. Sería óptimo graficar una segunda línea sin suavizado con un cierto “offset” respecto a la línea original.

Infomación auxiliar

Formato de imagen PPM

El formato PPM es un formato de archivo de imagen de mapa de bits (rasterizado) para imágenes a color, que significa “Portable Pixmap”. Almacena la información de cada píxel en forma no comprimida, lo que resulta en archivos grandes. Consiste en un encabezado con información sobre la imagen (como tipo y dimensiones) y los datos de color de los píxeles. Es un formato relativamente raro en la actualidad, pero es extremadamente fácil de generar en un programa y además tiene la ventaja que el contenido es legible por lo que sirve para hacer debugging de algoritmos.

Especificación del formato PPM: [ver acá](#)