Dibujando Líneas, Círculos y Elipses en una Cuadrícula

N. Wirth

28 de mayo de 2001

Traducción: N. Wolovick
Título original: "Drawing Lines, Circles and Ellipses in a Raster" en "Beauty is our Business, a birthday salute to Edsger W. Dijkstra", Springer Verlag, 1990.

1 Resumen

En un estilo de tutorial, utilizamos la notación y metodología de Dijkstra a fin de desarrollar el algoritmo de Bresenham para dibujar líneas rectas y círculos. Luego generalizamos el algoritmo del círculo para dibujar elipses.

2 Introducción

Recientemente necesitaba incorporar un algoritmo para dibujar mapas de bits en uno de mis programas. El algoritmo de Bresenham es conocido por su eficiencia, por lo que se convertía en el objetivo de mi búsqueda. La literatura rápidamente reveló descripciones en varias fuentes [1, 3]; todo lo que necesitaba era traducirlos a mi notación favorita. Sin embargo, yo quería (en contraste con los deseos de la computadora) no sólo interpretar los algoritmos, sino *entenderlos*. Descubrí que las fuentes elegidas fueron, aunque típicas, un poco inadecuadas para este propósito. Ellas reflejaban la amplia visión que tienen los cursos de programación para enseñar el uso de un lenguaje (específico), mientras que los algoritmos simplemente se dan.

Dijkstra fue un temprano y abierto crítico de esta visión, y correctamente señaló que las dificultades de la programación son inherentes en este tema, es decir en el razonamiento constructivo. Para enfatizar el tema central, Dijkstra comprimió el problema notacional a un mínimo, postulando su propia notación que se define de manera concisa con unas pocas fórmulas [2].

En el siguiente ejemplo adoptamos su notación, pero nos desviamos de su metodología especificando la tarea de dibujar de manera algorítmica en vez de utilizar un predicado de resultado sobre el plano de dibujo. En cada una de las tres curvas, la principal estructura del algoritmo es la repetición. Por cada iteración, se marca el siguiente elemento de la cuadrícula (pixel). En una de las dimensiones del plano de dibujo, el siguiente valor de la coordenada se obtiene sumando 1. Luego, las consideraciones respecto a terminación son triviales: la repetición termina cuando un valor límite ha sido alcanzado. El problema se reduce a computar la otra coordenada del siguiente punto de la cuadrícula a ser marcado.

La idea central de Bresenham es, en vez de evaluar la función que define la curva, computar la coordenada *incrementalmente* a partir de la coordenada del último pixel, utilizando solamente aritmética entera. El problema ahora se reduce a encontrar una función auxiliar h que determina si la coordenada debe ser incrementada o no. Si la pendiente de la curva está garantizada a ser como máximo de 45^o, el incremento es 0 o 1. Las sentencias para computar la función auxiliar en cada paso se derivan a partir de la simple aplicación del axioma de la asignación. La definición de la función aparece en el invariante del lazo.

3 Líneas

Sea la línea recta L definida por la siguiente ecuación

$$L : bx - ay = 0$$

Ahora queremos dibujar el segmento desde el origen al punto $P(a, b)$ y aceptar, sin pérdida de generalidad, la condición $0 \leq b \leq a$. Evidentemente esto puede ser realizado incrementando x paso a paso, computando y redondeando y cada vez y marcando el cuadrado con la coordenada resultante.

$x := 0;$
do $x < a \rightarrow$
  $y := bx\ DIV\ a;\ Dibuja(x, y);\ x := x + 1$
od

La ocurrencia de una multiplicación y una división en cada paso, indican la necesidad de una solución más eficiente, particularmente una que evite el uso de las fracciones (números de punto flotante). Esta se basa en la idea de determinar, a través de la evaluación de una simple función entera, si la ordenada y debe ser incrementada o no. Dado un punto x, y, el valor exacto Y de la ordenada en el siguente punto se obtiene de la ecuación para L:

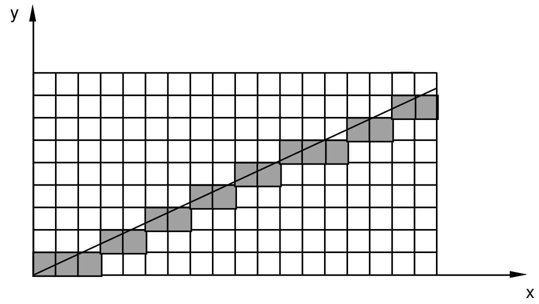
$$Y = b(x + 1)/a$$

El valor entero y más cercano a Y debe satisfacer la condición

$$Y - 1/2 \leq y \leq Y + 1/2$$

Si la primera parte de la desigualdad no se satisface, es necesario un incremento de y. Entonces un incremento en 1 implica que ambas partes se satisfacen:

$$\begin{aligned} b(x+1)/a - 1/2 &\leq y \\ bx + b - a/2 &\leq ay \\ bx - ay + b - a/2 &\leq 0 \end{aligned}$$



Introducimos una variable auxiliar $h = bx - ay - a/2$; esta igualdad es un invariante del lazo. Cada vez que x o y se incrementa, h debe ser ajustada de acuedo con su definición. Por aplicación directa del axioma de la asignación, determinamos la precondición para $x := x+1$:

$$\begin{array}{l} \{h = b(x+1) - ay + b - a/2\} \\ x := x + 1 \\ \{h = bx - ay + b - a/2\} \end{array}$$

y ahora queremos encontrar una expresión u tal que

$$\begin{array}{l} \{h = bx - ay + b - a/2\} \\ h := u \\ \{h = b(x+1) - ay + b - a/2\} \end{array}$$

Aplicando el axioma de asignación una vez más, encontramos $u = h + b$. Una aplicación similar brinda el ajuste necesario para h en caso de incrementar x e y, con el término k constante. Esto es, cuando x es incrementado solo, tenemos

$$\begin{array}{l} \{h = bx + k\} \\ h := h + b; \\ \{h = b(x+1) + k\} \\ x := x + 1 \\ \{h = bx + k\} \end{array}$$

y cuando los dos son incrementados, tenemos

$$\begin{array}{l} \{h = bx - ay + k\} \\ h := h + b - a; \\ \{h = b(x+1) - a(y+1) + k\} \\ y := y + 1; x := x + 1 \\ \{h = bx - ay + k\} \end{array}$$

El algoritmo resultante es conocido como el algoritmo de Bresenham. En la expresión que define el valor inicial de h, $a/2$ ha sido reemplazado por un DIV 2. Por lo tanto h puede ser (a lo más) $1/2$ más chico. Esto es aceptable pues para todos los enteros h, y para todo c tal que $0 \leq c < 1$, $h > 0 \equiv h > c$.

$$
\begin{array}{l}
x := 0;\ y := 0;\ h := b - a\ DIV\ 2; \\
\mathbf{do}\ \{\mathrm{P}\} \\
\quad x < a \rightarrow Dibuja(x, y); \\
\qquad\qquad \mathbf{if}\ h \leq 0 \rightarrow h := h + b \\
\qquad\qquad \Box\ h > 0 \rightarrow h := h + (b - a);\ y := y + 1 \\
\qquad\qquad \mathbf{fi}; \\
\qquad\qquad x := x + 1 \\
\mathbf{od}
\end{array}
$$

4 Círculos

Ahora nuestra tarea es dibujar un círculo C de radio r con su centro en el origen. Él está definido por la ecuación

$$C : x^2 + y^2 = r^2$$

El método obvio de computar los valores de x e y para diferentes ángulos usando funciones trigonométricas debe ser rechazado por ineficiente. Como en el caso de la línea recta, queremos operar sólo con enteros y encontrar una función simple que determine si la ordenada debe ser incrementada o no. Para esto solamente nos concentramos en dibujar el círculo en el primer octante del plano (0 a 45^o). Los 7 octantes restantes se cubren fácilmente con argumentos de simetía y no necesitan de más computación.

Comenzando desde el punto $P(r, 0)$, incrementamos y en pasos de una unidad. En cada paso la abcisa exacta X para el siguiente punto se deriva de la ecuación del círculo como

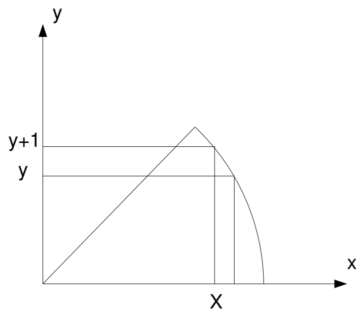
$$X = \sqrt{r^2 - (y+1)^2}$$

La abcisa del siguiente punto de la cuadrícula a ser marcado debe satisfacer las desigualdades

$$X - 1/2 \leq x \leq X + 1/2$$

Reemplazando X en la segunda parte se obtiene

$$
\begin{array}{rcl}
x - 1/2 & < & \sqrt{r^2 - (y+1)^2} \\
x^2 - x + 1/4 & < & r^2 - y^2 - 2y - 1 \\
x^2 + y^2 + 2y - x - r^2 + 5/4 & < & 0
\end{array}
$$



En cada paso y es incrementado y en caso de ser necesario x se decrementa de forma tal que la condición $h < 0$ sea reestablecida, donde

$$h = x^2 + y^2 + 2y - x - r^2 + 1$$

El ajuste necesario para h ante cambios de x o y se sigue de manera directa de la aplicación del axioma de la asignación:
cuando se incrementa y,

$$\begin{array}{l} \{h = y^2 + 2y + k\} \\ h := h + 2y + 3 \\ \{h = y^2 + 4y + 3 + k\} \\ y := y + 1 \\ \{h = y^2 + 2y + k\} \end{array}$$

cuando se decrementa x,

$$\begin{array}{l} \{h = x^2 - x + k\} \\ h := h - 2x + 2 \\ \{h = x^2 - 3x + 2 + k\} \\ x := x - 1 \\ \{h = x^2 - x + k\} \end{array}$$

De esto se sigue el muy eficiente algoritmo atribuído a Bresenham.

$x := r;\ y := 0;\ h := 1 - r;$
do $y < x \rightarrow Dibuja(x, y);$
    if $h < 0 \rightarrow h := h + 2y + 3$
    $\Box\ h \geq 0 \rightarrow h := h + 2(y - x) + 5;\ x := x - 1$
    fi;
    $y := y + 1$
od

5 Elipses

De manera similar al algoritmo del círculo, queremos diseñar un algoritmo para dibujar elipses para encontrar, paso a paso, los puntos de la cuadrícula a ser marcados. Nos concentramos en el primer cuadrante, los otros tres se cubren con argumentos de simetría y no requieren computaciones adicionales.

Sea la elipse definida por la siguiente ecuación. De nuevo, sin pérdida de generalidad, asumimos $0 < a \leq b$.

$$E : (x/a)^2 + (y/b)^2 = 1$$

Empezamos con el punto $P(0, b)$ y procedemos incrementando x en cada paso y decrementando y si es necesario. La ordenada exacta del siguiente punto se sigue de la ecuación que define:

$$Y = b\sqrt{1 - ((x + 1)/a)^2}$$

El punto de la cuadrícula debe satisfacer

$$\begin{aligned}
y - 1/2 &< b\sqrt{1 - ((x+1)/a)^2} \\
y^2 - y + 1/4 &< b^2 - b^2(x+1)^2/a^2 \\
a^2y^2 - a^2y + a^2/4 &< a^2b^2 - b^2x^2 - 2b^2x - b^2 \\
b^2x^2 + 2b^2x + a^2y^2 - a^2y + a^2/4 - a^2b^2 + b^2 &< 0
\end{aligned}$$

La condición necesaria y suficiente para decrementar y es entonces $h \geq 0$ con la variable auxiliar h definida por

$$h = b^2x^2 + 2b^2x + a^2y^2 - a^2y + a^2/4 - a^2b^2 + b^2$$

Como en el caso del círculo, la condición de terminación se da tan pronto como y deba ser decrementada por más que 1 después de incrementar x en 1, es decir cuando la tangente de la curva sea mayor a 45^o. Difiriendo del caso del círculo, obviamente la condición no está dada por $x = y$. Rechazamos la solución directa de computar la ordenada para la cual la derivada de la curva es -1, puesto que sólo esta computación incluiría la función de raíz cuadrada. En vez de esto computamos una función g similar a h de manera incremental. Se origina en la desigualdad

$$y - 3/2 < b\sqrt{1 - ((x+1)/a)^2}$$

implicando que la ordenada del siguiente punto debería ser al menos $3/2$ unidades por debajo del punto actual de la cuadrícula. Luego, un decremento de y en 2 sería necesario para un incremento de x en sólo 1. Un desarrollo similar al realizado con h genera una función g como

$$g = b^2x^2 + 2b^2x + a^2y^2 - 3a^2y + 9a^2/4 - a^2b^2 + b^2$$

y x puede ser incrementado siempre que $g < 0$. Luego el primer cuadrante de la elipse se completa en el mismo proceso, comenzando en el punto $P(a, 0)$, incrementando y y condicionalmente decrementando x. La función auxiliar h se obtiene del caso anterior de h sustituyendo sistemáticamente x, y, a, b por y, x, b, a. La derivación de los valores de incremento para h y g se siguen de la aplicación del axioma de la asignación: incrementando x, el incremento de h se obtiene de

$$\begin{aligned}
&\{h = b^2x^2 + 2b^2x + k\} \\
&h := h + b^2(2x + 3) \\
&\{h = b^2x^2 + 2b^2x + b^2 + 2b^2x + 2b^2 + k\} \\
&x := x + 1 \\
&\{h = b^2x^2 + 2b^2x + k\}
\end{aligned}$$

incrementando y, el incremento de h se obtiene de:

$$\begin{aligned}
&\{h = a^2y^2 - a^2y + k\} \\
&h := h - 2a^2(y - 1) \\
&\{h = a^2y^2 - 2a^2y + a^2 - (a^2y - a^2) + k\} \\
&y := y - 1 \\
&\{h = a^2y^2 - a^2y + k\}
\end{aligned}$$

mientras que el incremento de g se obtiene de

$$\begin{aligned}
&\{g = a^2y^2 - 3a^2y + k\}\\
&g := g - 2a^2(y-2)\\
&\{g = a^2y^2 - 2a^2y + a^2 - 3(a^2y - a^2) + k\}\\
&y := y - 1\\
&\{g = a^2y^2 - 3a^2y + k\}
\end{aligned}$$

Todo esto completa las consideraciones de diseño para el siguiente algoritmo.

$$\begin{array}{l}
x := 0;\ y := b;\\
h := (a^2\ DIV\ 4) - ba^2 + b^2;\ g := (9/4)a^2 - 3ba^2 + b^2;\\
\mathbf{do}\ g < 0 \rightarrow Dibuja(x,y);\\
\qquad\qquad \mathbf{if}\ h < 0 \rightarrow d := (2x+3)b^2;\ g := g + d\\
\qquad\qquad \Box\ h \geq 0 \rightarrow d := (2x+3)b^2 - 2(y-1)a^2;\\
\qquad\qquad\qquad\qquad g := g + d + 2a^2;\\
\qquad\qquad\qquad\qquad y := y - 1\\
\qquad\qquad \mathbf{fi};\\
\qquad\qquad h := h + d;\ x := x + 1\\
\mathbf{od};\ x := a;\ y1 := y;\ y := 0;\\
h := (b^2\ DIV\ 4) - ab^2 + 2a^2;\\
\mathbf{do}\ y < y1 \rightarrow Dibuja(x,y);\\
\qquad\qquad \mathbf{if}\ h < 0 \rightarrow h := h + (2y+3)a^2;\\
\qquad\qquad \Box\ h \geq 0 \rightarrow h := h + (2y+3)a^2 - 2(x-1)b^2;\ x := x - 1\\
\qquad\qquad \mathbf{fi};\\
\mathbf{od};
\end{array}$$

Cerramos este artículo remarcando que los valores de h pueden tornarse altos con lo cual puede ocurrir desbordamiento cuando el algoritmo es interpretado en computadoras con un tamaño de palabra insuficiente. Desafortunadamente la mayoría de los sistemas de computadora no indican el desbordamiento entero. Usando aritmética de 32 bits, se pueden dibujar elipses con valores de a y b hasta 1000 sin falla alguna.

Referencias

[1] N. Cossitt. *Line Drawing with NS32CG16 and Drawing Circles with the NS32CG16.* Technical Report AN-522 and AN-523, National Semiconductor Corp., 1988.

[2] E. W. Dijkstra. *Guarded commands, non-determinacy, and the formal derivation of programs.* Comm. ACM, 18(8):453-457, August 1975.

[3] J. D. Foley and A. Van Dam. *Fundamentals of Interactive Computer Graphics.* Addison-Wesley, 1982.