Scan-conversion pentru cercuri și elipse

Marian Ioan MUNTEANU

Al.I.Cuza University of Iasi, Romania
webpage: http://www.math.uaic.ro/∼munteanu

22 Octombrie 2012

Cuprins

Scan-conversion pentru cercuri

Ecuația unui cerc cu centrul în origine și de rază R este

$$x^2 + y^2 = R^2$$

Modalitatea cea mai simplă: $y = \pm\sqrt{R^2 - x^2}$

și apoi de a calcula y pentru fiecare creștere a lui x.

Pe un sfert de cerc: x crește, din unitate în unitate, de la 0 la R.

- √ această aproximare este foarte costisitoare: la fiecare pas avem de făcut două ridicări la pătrat și o extragere a rădăcinii pătrate
- √ nu dă nici măcar rezultate corecte, anume că apropierea de punctul de tangență cu dreapta $x = R$ duce la un efect de "rărire" (vezi figura).

Scan-conversion pentru cercuri

Ecuația unui cerc cu centrul în origine și de rază R este

$$x^2 + y^2 = R^2$$

Modalitatea cea mai simplă: $y = \pm\sqrt{R^2 - x^2}$

și apoi de a calcula y pentru fiecare creștere a lui x.

Pe un sfert de cerc: x crește, din unitate în unitate, de la 0 la R.

- √ această aproximare este foarte costisitoare: la fiecare pas avem de făcut două ridicări la pătrat și o extragere a rădăcinii pătrate
- √ nu dă nici măcar rezultate corecte, anume că apropierea de punctul de tangență cu dreapta $x = R$ duce la un efect de "rărire" (vezi figura).

Scan-conversion pentru cercuri

Ecuația unui cerc cu centrul în origine și de rază R este

$$x^2 + y^2 = R^2$$

Modalitatea cea mai simplă: $y = \pm\sqrt{R^2 - x^2}$

și apoi de a calcula y pentru fiecare creștere a lui x.

Pe un sfert de cerc: x crește, din unitate în unitate, de la 0 la R.

- √ această aproximare este foarte costisitoare: la fiecare pas avem de făcut două ridicări la pătrat și o extragere a rădăcinii pătrate
- √ nu dă nici măcar rezultate corecte, anume că apropierea de punctul de tangență cu dreapta $x = R$ duce la un efect de "rărire" (vezi figura).

Scan-conversion pentru cercuri

Ecuația unui cerc cu centrul în origine și de rază R este

$$x^2 + y^2 = R^2$$

Modalitatea cea mai simplă: $y = \pm\sqrt{R^2 - x^2}$

și apoi de a calcula y pentru fiecare creștere a lui x.

Pe un sfert de cerc: x crește, din unitate în unitate, de la 0 la R.

- √ această aproximare este foarte costisitoare: la fiecare pas avem de făcut două ridicări la pătrat și o extragere a rădăcinii pătrate
- √ nu dă nici măcar rezultate corecte, anume că apropierea de punctul de tangență cu dreapta $x = R$ duce la un efect de "rărire" (vezi figura).

Scan-conversion pentru cercuri

Ecuația unui cerc cu centrul în origine și de rază R este

$$x^2 + y^2 = R^2$$

Modalitatea cea mai simplă: $y = \pm\sqrt{R^2 - x^2}$

și apoi de a calcula y pentru fiecare creștere a lui x.

Pe un sfert de cerc: x crește, din unitate în unitate, de la 0 la R.

- √ această aproximare este foarte costisitoare: la fiecare pas avem de făcut două ridicări la pătrat și o extragere a rădăcinii pătrate
- √ nu dă nici măcar rezultate corecte, anume că apropierea de punctul de tangență cu dreapta $x = R$ duce la un efect de "rărire" (vezi figura).

Scan-conversion pentru cercuri

Ecuația unui cerc cu centrul în origine și de rază R este

$$x^2 + y^2 = R^2$$

Modalitatea cea mai simplă: $y = \pm\sqrt{R^2 - x^2}$

și apoi de a calcula y pentru fiecare creștere a lui x.

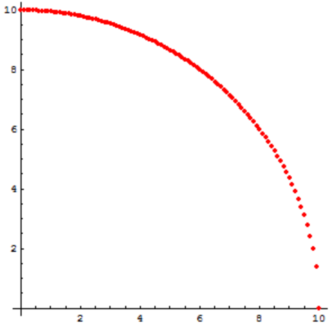
Pe un sfert de cerc: x crește, din unitate în unitate, de la 0 la R.

- √ această aproximare este foarte costisitoare: la fiecare pas avem de făcut două ridicări la pătrat și o extragere a rădăcinii pătrate
- √ nu dă nici măcar rezultate corecte, anume că apropierea de punctul de tangență cu dreapta $x = R$ duce la un efect de "rărire" (vezi figura).

Rărirea punctelor

```
In[1]:= R = 10;

        Show[Graphics[{RGBColor[1, 0, 0], PointSize[.015],
          Point /@ Table[{x, √(R^2 - x^2)}, {x, 0, R, .1}]}, AspectRatio → Automatic, Axes → True]];
```



Scan-conversion pentru cercuri

O observație banală, dar extrem de utilă: rasterizarea unui cerc poate fi făcută pe optimi de cerc

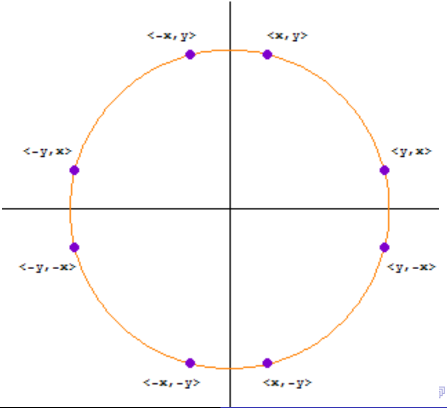
$(x, y) : (-x, y), (x, -y), (-x, -y), (y, x), (-y, x), (y, -x), (-y, -x).$

Scan-conversion pentru cercuri

O observație banală, dar extrem de utilă: rasterizarea unui cerc poate fi făcută pe optimi de cerc
$(x, y) : (-x, y), (x, -y), (-x, -y), (y, x), (-y, x), (y, -x), (-y, -x)$.

Scan-conversion pentru cercuri

O observație banală, dar extrem de utilă: rasterizarea unui cerc poate fi făcută pe optimi de cerc
$(x, y) : (-x, y), (x, -y), (-x, -y), (y, x), (-y, x), (y, -x), (-y, -x)$.



Algoritmul lui Bresenham pentru cercuri

La fel ca și pentru drepte, strategia este aceea de a alege dintre doi pixeli pe cel care reprezintă cel mai bine punctul de pe cerc cu ajutorul calculului unei funcții în mijlocul segmentului care unește cei doi pixeli.

Funcția care definește cercul este $F(x, y) = x^2 + y^2 - R^2$.

Aceasta ia valoarea 0 pe cerc, ia valori negative în interior și valori pozitive în exteriorul său.

Facem calculul pentru al doilea octant (x de la 0 la $\frac{R}{\sqrt{2}}$ și y>0).

Algoritmul lui Bresenham pentru cercuri

La fel ca și pentru drepte, strategia este aceea de a alege dintre doi pixeli pe cel care reprezintă cel mai bine punctul de pe cerc cu ajutorul calculului unei funcții în mijlocul segmentului care unește cei doi pixeli.

Funcția care definește cercul este $F(x, y) = x^2 + y^2 - R^2$.

Aceasta ia valoarea 0 pe cerc, ia valori negative în interior și valori pozitive în exteriorul său.

Facem calculul pentru al doilea octant (x de la 0 la $\frac{R}{\sqrt{2}}$ și y>0).

Algoritmul lui Bresenham pentru cercuri

La fel ca și pentru drepte, strategia este aceea de a alege dintre doi pixeli pe cel care reprezintă cel mai bine punctul de pe cerc cu ajutorul calculului unei funcții în mijlocul segmentului care unește cei doi pixeli.

Funcția care definește cercul este $F(x, y) = x^2 + y^2 - R^2$.

Aceasta ia valoarea 0 pe cerc, ia valori negative în interior și valori pozitive în exteriorul său.

Facem calculul pentru al doilea octant (x de la 0 la $\frac{R}{\sqrt{2}}$ și y>0).

Algoritmul lui Bresenham pentru cercuri

Presupunem că am ales pixelul P de coordonate (x_p, y_p).
Următorul pixel care va fi ales: E sau SE.

Astfel, vom alege SE dacă M este în afara cercului i.e. $F(M) > 0$
și E dacă M este în interiorul cercului i.e. $F(M) < 0$.

La fel ca și în cazul dreptei definim d, variabila de decizie, prin:

$$d = F(M) = F(x_p + 1, y_p - \tfrac{1}{2}) = (x_p + 1)^2 + (y_p - \tfrac{1}{2})^2 - R^2 =$$
$$= x_p^2 + y_p^2 + 2x_p - y_p - R^2 + \tfrac{5}{4}\ .$$

Algoritmul lui Bresenham pentru cercuri

Presupunem că am ales pixelul P de coordonate (x_p, y_p).
Următorul pixel care va fi ales: E sau SE.

Astfel, vom alege SE dacă M este în afara cercului i.e. $F(M) > 0$
și E dacă M este în interiorul cercului i.e. $F(M) < 0$.

La fel ca și în cazul dreptei definim d, variabila de decizie, prin:

$$d = F(M) = F(x_p + 1, y_p - \frac{1}{2}) = (x_p + 1)^2 + (y_p - \frac{1}{2})^2 - R^2 =$$
$$= x_p^2 + y_p^2 + 2x_p - y_p - R^2 + \frac{5}{4}\ .$$

Algoritmul lui Bresenham pentru cercuri

Presupunem că am ales pixelul P de coordonate (x_p, y_p).
Următorul pixel care va fi ales: E sau SE.

Astfel, vom alege SE dacă M este în afara cercului i.e. $F(M) > 0$
și E dacă M este în interiorul cercului i.e. $F(M) < 0$.

La fel ca și în cazul dreptei definim d, variabila de decizie, prin:

$$d = F(M) = F(x_p + 1, y_p - \tfrac{1}{2}) = (x_p + 1)^2 + (y_p - \tfrac{1}{2})^2 - R^2 =$$
$$= x_p^2 + y_p^2 + 2x_p - y_p - R^2 + \tfrac{5}{4}\ .$$

Algoritmul lui Bresenham pentru cercuri

- dacă $d > 0$ alegem SE, iar valoarea următoare a lui d va fi

$$d_{\text{new}} = F(x_p + 2, y_p - \tfrac{3}{2}) = (x_p + 2)^2 + (y_p - \tfrac{3}{2})^2 - R^2 = \\ = d + \underbrace{2x_p - 2y_p + 5}_{\text{delta}SE} \longrightarrow d_{new} = d + deltaSE \ \ (\mathbf{d{+}{=}deltaSE}).$$

- dacă $d \leq 0$ alegem E și astfel valoarea următoare a lui d va fi

$$d_{new} = F(x_p + 2, y_p - \tfrac{1}{2}) = (x_p + 2)^2 + (y_p - \tfrac{1}{2})^2 - R^2 = \\ = d + \underbrace{2x_p + 3}_{\text{delta}E} \longrightarrow d_{new} = d + deltaE \ \ (\mathbf{d{+}{=}deltaE})$$

Algoritmul lui Bresenham pentru cercuri

- dacă $d > 0$ alegem SE, iar valoarea următoare a lui d va fi

$$\begin{aligned} d_{\text{new}} &= F(x_p + 2, y_p - \tfrac{3}{2}) = (x_p + 2)^2 + (y_p - \tfrac{3}{2})^2 - R^2 = \\ &= d + \underbrace{2x_p - 2y_p + 5}_{\text{delta}SE} \quad \longrightarrow d_{new} = d + deltaSE \;\; (\mathbf{d+=deltaSE}). \end{aligned}$$

- dacă $d \leq 0$ alegem E și astfel valoarea următoare a lui d va fi

$$\begin{aligned} d_{new} &= F(x_p + 2, y_p - \tfrac{1}{2}) = (x_p + 2)^2 + (y_p - \tfrac{1}{2})^2 - R^2 = \\ &= d + \underbrace{2x_p + 3}_{\text{delta}E} \qquad \longrightarrow d_{new} = d + deltaE \;\; (\mathbf{d+=deltaE}) \end{aligned}$$

Algoritmul lui Bresenham pentru cercuri

valorile diferențelor *deltaE* și *deltaSE* care apar **nu mai sunt constante** ci funcții de pixelul precedent al rasterizării **!!!!**

Valoarea de start a lui d, dată de primul mijloc, este:

$$d_{start} = F\left(1, R - \frac{1}{2}\right) = 1^2 + \left(R - \frac{1}{2}\right)^2 - R^2 = \frac{5}{4} - R.$$

Înlocuind d cu $h + \frac{1}{4}$ se obține $h_{start} = 1 - R$,

$d \leq$ (sau > 0) devine $h \leq$ (sau $>$) $\frac{1}{4}$.

Algoritmul lui Bresenham pentru cercuri

valorile diferențelor *deltaE* și *deltaSE* care apar **nu mai sunt constante** ci funcții de pixelul precedent al rasterizării **!!!!**

Valoarea de start a lui d, dată de primul mijloc, este:

$$d_{start} = F\left(1, R - \frac{1}{2}\right) = 1^2 + \left(R - \frac{1}{2}\right)^2 - R^2 = \frac{5}{4} - R.$$

Înlocuind d cu $h + \frac{1}{4}$ se obține $h_{start} = 1 - R$,

$d \leq$ (sau $>$ 0) devine $h \leq$ (sau $>$) $\frac{1}{4}$.

Algoritmul lui Bresenham pentru cercuri

valorile diferențelor *deltaE* și *deltaSE* care apar **nu mai sunt constante** ci funcții de pixelul precedent al rasterizării **!!!!**

Valoarea de start a lui d, dată de primul mijloc, este:

$$d_{start} = F\left(1, R - \frac{1}{2}\right) = 1^2 + \left(R - \frac{1}{2}\right)^2 - R^2 = \frac{5}{4} - R.$$

Înlocuind d cu $h + \frac{1}{4}$ se obține $h_{start} = 1 - R$,

$d \leq (\text{sau} > 0)$ devine $h \leq (\text{sau} >)\ \frac{1}{4}$·.

Calculul diferențelor

deltaE și *deltaSE* sunt funcții polinomiale de gradul I,

pe care le vom calcula printr-o metodă incrementală.

Metoda se numește *metoda diferențelor parțiale de ordinul al II-lea.*

Presupunem că se alege pixelul E;

atunci punctul P se mută în $E(x_P + 1, y_P)$.

Funcția *deltaE* care avea valoarea $deltaE = 2x_p + 3 \quad (în\ P)$

are acum valoarea $deltaE_{new} = 2(x_p + 1) + 3 = 2x_p + 5 \qquad (în\ E)$

Astfel, $deltaE_{new} = deltaE + 2 \quad (\textbf{deltaE += 2})$.

În mod similar $deltaSE = 2x_p - 2y_p + 5 \quad (în\ P)$

iar $deltaSE_{new} = 2(x_p + 1 - 2y_p + 5) = 2x_p - 2y_p + 7 \quad (în\ E)$.

Astfel, $deltaSE_{new} = deltaSE + 2 \quad (\textbf{deltaSE += 2})$.

Calculul diferențelor

deltaE și *deltaSE* sunt funcții polinomiale de gradul I,
pe care le vom calcula printr-o metodă incrementală.

Metoda se numește *metoda diferențelor parțiale de ordinul al II-lea.*

Presupunem că se alege pixelul E;
atunci punctul P se mută în $E(x_P + 1, y_P)$.

Funcția *deltaE* care avea valoarea $deltaE = 2x_p + 3 \quad (\hat{\imath}n\ P)$
are acum valoarea $deltaE_{new} = 2(x_p + 1) + 3 = 2x_p + 5 \qquad (\hat{\imath}n\ E)$

Astfel, $deltaE_{new} = deltaE + 2$ (**deltaE += 2**).

În mod similar $deltaSE = 2x_p - 2y_p + 5 \quad (\hat{\imath}n\ P)$

iar $deltaSE_{new} = 2(x_p + 1 - 2y_p + 5) = 2x_p - 2y_p + 7 \quad (\hat{\imath}n\ E)$.

Astfel, $deltaSE_{new} = deltaSE + 2$ (**deltaSE += 2**).

Calculul diferențelor

deltaE și *deltaSE* sunt funcții polinomiale de gradul I,

pe care le vom calcula printr-o metodă incrementală.

Metoda se numește *metoda diferențelor parțiale de ordinul al II-lea.*

Presupunem că se alege pixelul E;

atunci punctul P se mută în $E(x_P + 1, y_P)$.

Funcția *deltaE* care avea valoarea $deltaE = 2x_p + 3 \quad (\hat{i}n\ P)$

are acum valoarea $deltaE_{new} = 2(x_p + 1) + 3 = 2x_p + 5 \qquad (\hat{i}n\ E)$

Astfel, $deltaE_{new} = deltaE + 2 \quad (\mathbf{deltaE\ += 2})$.

În mod similar $deltaSE = 2x_p - 2y_p + 5 \quad (\hat{i}n\ P)$

iar $deltaSE_{new} = 2(x_p + 1 - 2y_p + 5) = 2x_p - 2y_p + 7 \quad (\hat{i}n\ E)$.

Astfel, $deltaSE_{new} = deltaSE + 2 \quad (\mathbf{deltaSE\ += 2})$.

Calculul diferențelor

deltaE și *deltaSE* sunt funcții polinomiale de gradul I,
pe care le vom calcula printr-o metodă incrementală.
Metoda se numește *metoda diferențelor parțiale de ordinul al II-lea.*
Presupunem că se alege pixelul E;
atunci punctul P se mută în $E(x_P + 1, y_P)$.

Funcția *deltaE* care avea valoarea $deltaE = 2x_p + 3 \quad (\hat{\imath}n\ P)$
are acum valoarea $deltaE_{new} = 2(x_p + 1) + 3 = 2x_p + 5 \qquad (\hat{\imath}n\ E)$

Astfel, $deltaE_{new} = deltaE + 2$ $\quad$ (**deltaE += 2**).

În mod similar $deltaSE = 2x_p - 2y_p + 5 \quad (\hat{\imath}n\ P)$

iar $deltaSE_{new} = 2(x_p + 1 - 2y_p + 5) = 2x_p - 2y_p + 7 \quad (\hat{\imath}n\ E)$.

Astfel, $deltaSE_{new} = deltaSE + 2$ $\quad$ (**deltaSE += 2**).

Calculul diferențelor

deltaE și *deltaSE* sunt funcții polinomiale de gradul I,
pe care le vom calcula printr-o metodă incrementală.
Metoda se numește *metoda diferențelor parțiale de ordinul al II-lea.*
Presupunem că se alege pixelul E;
atunci punctul P se mută în $E(x_P + 1, y_P)$.

Funcția *deltaE* care avea valoarea $deltaE = 2x_p + 3 \quad (\hat{\imath}n\ P)$
are acum valoarea $deltaE_{new} = 2(x_p + 1) + 3 = 2x_p + 5 \qquad (\hat{\imath}n\ E)$

Astfel, $deltaE_{new} = deltaE + 2 \quad$ (**deltaE += 2**).

În mod similar $deltaSE = 2x_p - 2y_p + 5 \quad (\hat{\imath}n\ P)$

iar $deltaSE_{new} = 2(x_p + 1 - 2y_p + 5) = 2x_p - 2y_p + 7 \quad (\hat{\imath}n\ E)$.

Astfel, $deltaSE_{new} = deltaSE + 2 \quad$ (**deltaSE += 2**).

Calculul diferențelor

deltaE și *deltaSE* sunt funcții polinomiale de gradul I,
pe care le vom calcula printr-o metodă incrementală.
Metoda se numește *metoda diferențelor parțiale de ordinul al II-lea.*
Presupunem că se alege pixelul E;
atunci punctul P se mută în $E(x_P + 1, y_P)$.

Funcția *deltaE* care avea valoarea $deltaE = 2x_p + 3 \quad (\hat{\imath}n\ P)$
are acum valoarea $deltaE_{new} = 2(x_p + 1) + 3 = 2x_p + 5 \qquad (\hat{\imath}n\ E)$

Astfel, $deltaE_{new} = deltaE + 2$ (**deltaE += 2**).

În mod similar $deltaSE = 2x_p - 2y_p + 5 \quad (\hat{\imath}n\ P)$

iar $deltaSE_{new} = 2(x_p + 1 - 2y_p + 5) = 2x_p - 2y_p + 7 \quad (\hat{\imath}n\ E)$.

Astfel, $deltaSE_{new} = deltaSE + 2$ (**deltaSE += 2**).

Calculul diferențelor

deltaE și *deltaSE* sunt funcții polinomiale de gradul I,
pe care le vom calcula printr-o metodă incrementală.
Metoda se numește *metoda diferențelor parțiale de ordinul al II-lea.*
Presupunem că se alege pixelul E;
atunci punctul P se mută în $E(x_P + 1, y_P)$.

Funcția *deltaE* care avea valoarea $deltaE = 2x_p + 3 \quad (\hat{\imath}n\ P)$
are acum valoarea $deltaE_{new} = 2(x_p + 1) + 3 = 2x_p + 5 \qquad (\hat{\imath}n\ E)$

Astfel, $deltaE_{new} = deltaE + 2 \quad$ (**deltaE += 2**).

În mod similar $deltaSE = 2x_p - 2y_p + 5 \quad (\hat{\imath}n\ P)$

iar $deltaSE_{new} = 2(x_p + 1 - 2y_p + 5) = 2x_p - 2y_p + 7 \quad (\hat{\imath}n\ E)$.

Astfel, $deltaSE_{new} = deltaSE + 2 \quad$ (**deltaSE += 2**).

Calculul diferențelor

deltaE și *deltaSE* sunt funcții polinomiale de gradul I,
pe care le vom calcula printr-o metodă incrementală.
Metoda se numește *metoda diferențelor parțiale de ordinul al II-lea.*
Presupunem că se alege pixelul E;
atunci punctul P se mută în $E(x_P + 1, y_P)$.

Funcția *deltaE* care avea valoarea $deltaE = 2x_p + 3 \quad (\hat{\imath}n\ P)$
are acum valoarea $deltaE_{new} = 2(x_p + 1) + 3 = 2x_p + 5 \quad (\hat{\imath}n\ E)$

Astfel, $deltaE_{new} = deltaE + 2$ (**deltaE += 2**).

În mod similar $deltaSE = 2x_p - 2y_p + 5 \quad (\hat{\imath}n\ P)$

iar $deltaSE_{new} = 2(x_p + 1 - 2y_p + 5) = 2x_p - 2y_p + 7 \quad (\hat{\imath}n\ E)$.

Astfel, $deltaSE_{new} = deltaSE + 2$ (**deltaSE += 2**).

Calculul diferențelor

Dacă însă se alege pixelul SE

punctul P se mută în $SE(x_p + 1, y_p - 1)$

$deltaE_{new} = 2x_p + 5$ (în SE)

și deci $deltaE_{new} = deltaE + 2$ **(deltaE += 2)**

$deltaSE_{new} = 2x_p - 2y_p + 9$ (în SE)

și deci $deltaSE_{new} = deltaSE + 4$ **(deltaSE += 4)**

Calculul diferențelor

Dacă însă se alege pixelul SE

punctul P se mută în $SE(x_p + 1, y_p - 1)$

$deltaE_{new} = 2x_p + 5$ (în SE)

și deci $deltaE_{new} = deltaE + 2$ **(deltaE += 2)**

$deltaSE_{new} = 2x_p - 2y_p + 9$ (în SE)

și deci $deltaSE_{new} = deltaSE + 4$ **(deltaSE += 4)**

Calculul diferențelor

Dacă însă se alege pixelul SE

punctul P se mută în $SE(x_p + 1, y_p - 1)$

$deltaE_{new} = 2x_p + 5$ (în SE)

și deci $deltaE_{new} = deltaE + 2$ **(deltaE += 2)**

$deltaSE_{new} = 2x_p - 2y_p + 9$ (în SE)

și deci $deltaSE_{new} = deltaSE + 4$ **(deltaSE += 4)**

Calculul diferențelor

Dacă însă se alege pixelul SE

punctul P se mută în $SE(x_p + 1, y_p - 1)$

$deltaE_{new} = 2x_p + 5$ (în SE)

și deci $deltaE_{new} = deltaE + 2$ **(deltaE += 2)**

$deltaSE_{new} = 2x_p - 2y_p + 9$ (în SE)

și deci $deltaSE_{new} = deltaSE + 4$ **(deltaSE += 4)**

Calculul diferențelor

Dacă însă se alege pixelul SE

punctul P se mută în $SE(x_p + 1, y_p - 1)$

$deltaE_{new} = 2x_p + 5$ (în SE)

și deci $deltaE_{new} = deltaE + 2$ **(deltaE += 2)**

$deltaSE_{new} = 2x_p - 2y_p + 9$ (în SE)

și deci $deltaSE_{new} = deltaSE + 4$ **(deltaSE += 4)**

Calculul diferențelor

Dacă însă se alege pixelul SE

punctul P se mută în $SE(x_p + 1, y_p - 1)$

$deltaE_{new} = 2x_p + 5$ (în SE)

și deci $deltaE_{new} = deltaE + 2$ **(deltaE += 2)**

$deltaSE_{new} = 2x_p - 2y_p + 9$ (în SE)

și deci $deltaSE_{new} = deltaSE + 4$ **(deltaSE += 4)**

Algoritmul C

```
x=0;
y=raza;
d=1-raza;
deltaE=3;
deltaSE=5-2*raza;
putpixel(x,y,RED);
```

Algoritmul C

```
do {
    if (d < 0) {              /* alegem E */
        d+=deltaE;
        deltaE+=2;
        deltaSE+=2;
        x++;
    } else {                  /* alegem SE */
        d+=deltaSE;
        deltaE+=2;
        deltaSE+=4;
        x++;
        y--;
    }
```

Algoritmul C

/* se vor pune in evidenta cele 8 arce de cerc folosindu-se culori diferite */

```
        putpixel(x, y, GREEN);
        putpixel(-x, -y, RED);
        putpixel(x, -y, MAGENTA);
        putpixel(-x, y, BLUE);
        putpixel(y, x, LIGHTRED);
        putpixel(-y, -x, LIGHTGREEN);
        putpixel(y, -x, LIGHTBLUE);
        putpixel(-y, x, LIGHTMAGENTA);
    }
while (y > x);
```

Algoritmul da Silva

Considerăm elipsa $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$.

Se divide primul cadran în două regiuni în urma determinării punctului în care tangenta la elipsă are panta -1.

Considerăm $F(x, y) = b^2 \cdot x^2 + a^2 \cdot y^2 - a^2 \cdot b^2$.

În punctul $A(x_0, y_0)$ în care componentele normalei sunt egale avem $m = f'(x_0) = -1$, unde $f(x) = b \cdot \sqrt{1 - \frac{x^2}{a^2}}$ este ecuația explicită a elipsei (în primul cadran) iar x_0 și y_0 verifică $\frac{x_0^2}{a^2} + \frac{y_0^2}{b^2} = 1$.

Rezultă $a^2 \cdot y_0 = b^2 \cdot x_0$.

Algoritmul da Silva

Considerăm elipsa $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$.

Se divide primul cadran în două regiuni în urma determinării punctului în care tangenta la elipsă are panta -1.

Considerăm $F(x, y) = b^2 \cdot x^2 + a^2 \cdot y^2 - a^2 \cdot b^2$.

În punctul $A(x_0, y_0)$ în care componentele normalei sunt egale avem $m = f'(x_0) = -1$, unde $f(x) = b \cdot \sqrt{1 - \frac{x^2}{a^2}}$ este ecuația explicită a elipsei (în primul cadran) iar x_0 și y_0 verifică $\frac{x_0^2}{a^2} + \frac{y_0^2}{b^2} = 1$.

Rezultă $a^2 \cdot y_0 = b^2 \cdot x_0$.

Algoritmul da Silva

Considerăm elipsa $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$.

Se divide primul cadran în două regiuni în urma determinării punctului în care tangenta la elipsă are panta -1.

Considerăm $F(x, y) = b^2 \cdot x^2 + a^2 \cdot y^2 - a^2 \cdot b^2$.

În punctul $A(x_0, y_0)$ în care componentele normalei sunt egale avem $m = f'(x_0) = -1$, unde $f(x) = b \cdot \sqrt{1 - \frac{x^2}{a^2}}$ este ecuația explicită a elipsei (în primul cadran) iar x_0 și y_0 verifică $\frac{x_0^2}{a^2} + \frac{y_0^2}{b^2} = 1$.

Rezultă $a^2 \cdot y_0 = b^2 \cdot x_0$.

Algoritmul da Silva

Considerăm elipsa $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$.

Se divide primul cadran în două regiuni în urma determinării punctului în care tangenta la elipsă are panta -1.

Considerăm $F(x, y) = b^2 \cdot x^2 + a^2 \cdot y^2 - a^2 \cdot b^2$.

În punctul $A(x_0, y_0)$ în care componentele normalei sunt egale avem $m = f'(x_0) = -1$, unde $f(x) = b \cdot \sqrt{1 - \frac{x^2}{a^2}}$ este ecuația explicită a elipsei (în primul cadran) iar x_0 și y_0 verifică $\frac{x_0^2}{a^2} + \frac{y_0^2}{b^2} = 1$.

Rezultă $a^2 \cdot y_0 = b^2 \cdot x_0$.

Algoritmul da Silva

Considerăm elipsa $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$.

Se divide primul cadran în două regiuni în urma determinării punctului în care tangenta la elipsă are panta -1.

Considerăm $F(x, y) = b^2 \cdot x^2 + a^2 \cdot y^2 - a^2 \cdot b^2$.

În punctul $A(x_0, y_0)$ în care componentele normalei sunt egale avem $m = f'(x_0) = -1$, unde $f(x) = b \cdot \sqrt{1 - \frac{x^2}{a^2}}$ este ecuația explicită a elipsei (în primul cadran) iar x_0 și y_0 verifică $\frac{x_0^2}{a^2} + \frac{y_0^2}{b^2} = 1$.

Rezultă $a^2 \cdot y_0 = b^2 \cdot x_0$.

Algoritmul da Silva

Considerăm elipsa $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$.

Se divide primul cadran în două regiuni în urma determinării punctului în care tangenta la elipsă are panta -1.

Considerăm $F(x, y) = b^2 \cdot x^2 + a^2 \cdot y^2 - a^2 \cdot b^2$.

În punctul $A(x_0, y_0)$ în care componentele normalei sunt egale avem $m = f'(x_0) = -1$, unde $f(x) = b \cdot \sqrt{1 - \frac{x^2}{a^2}}$ este ecuația explicită a elipsei (în primul cadran) iar x_0 și y_0 verifică $\frac{x_0^2}{a^2} + \frac{y_0^2}{b^2} = 1$.

Rezultă $a^2 \cdot y_0 = b^2 \cdot x_0$.

Algoritmul da Silva

Să studiem regiunea I: $\boxed{b^2 \cdot x < a^2 \cdot y}$

Variabila de decizie va fi

$d = F(x_P+1, y_P-\frac{1}{2}) = b^2 \cdot x_p^2 + a^2 \cdot y_p^2 - a^2 \cdot b^2 + 2b^2 \cdot x_p - a^2 \cdot y_p + \frac{a^2}{4}$

- dacă $d = F(M) < 0$, M este în interior, atunci alegem E

$d_{new} = F(x_P+2, y_P-\frac{1}{2}) =$
$b^2 \cdot x_P^2 + a^2 \cdot y_P^2 - a^2 \cdot b^2 + 4b^2 \cdot x_P - a^2 \cdot y_P + 4b^2 + \frac{a^2}{4}$
$\rightsquigarrow d_{new} = d + \underbrace{2b^2 \cdot x_P + 3b^2}_{deltaE}$ **(d += deltaE)**

- dacă $d = F(M) > 0$, M este în exterior, alegem SE

$d_{new} = F(x_P+2, y_P-\frac{3}{2}) =$
$b^2 \cdot x_P^2 + a^2 \cdot y_P^2 - a^2 \cdot b^2 + 4b^2 \cdot x_P - 3a^2 \cdot y_P + 4b^2 + \frac{9a^2}{4}$.
$\rightsquigarrow d_{new} = d + \underbrace{2b^2 \cdot x_P - 2a^2 \cdot y_P + 3b^2 + 2a^2}_{deltaSE}$ **(d += deltaSE)**

Algoritmul da Silva

Să studiem regiunea I: $\boxed{b^2 \cdot x < a^2 \cdot y}$

Variabila de decizie va fi

$$d = F(x_P + 1, y_P - \tfrac{1}{2}) = b^2 \cdot x_p^2 + a^2 \cdot y_p^2 - a^2 \cdot b^2 + 2b^2 \cdot x_p - a^2 \cdot y_p + \frac{a^2}{4}$$

- dacă $d = F(M) < 0$, M este în interior, atunci alegem E

$d_{new} = F(x_P + 2, y_P - \frac{1}{2}) =$

$b^2 \cdot x_P^2 + a^2 \cdot y_P^2 - a^2 \cdot b^2 + 4b^2 \cdot x_P - a^2 \cdot y_P + 4b^2 + \frac{a^2}{4}$

$\rightsquigarrow d_{new} = d + \underbrace{2b^2 \cdot x_P + 3b^2}_{deltaE}$ **(d += deltaE)**

- dacă $d = F(M) > 0$, M este în exterior, alegem SE

$d_{new} = F(x_P + 2, y_P - \frac{3}{2}) =$

$b^2 \cdot x_P^2 + a^2 \cdot y_P^2 - a^2 \cdot b^2 + 4b^2 \cdot x_P - 3a^2 \cdot y_P + 4b^2 + \frac{9a^2}{4}.$

$\rightsquigarrow d_{new} = d + \underbrace{2b^2 \cdot x_P - 2a^2 \cdot y_P + 3b^2 + 2a^2}_{deltaSE}$ **(d += deltaSE)**

Algoritmul da Silva

Să studiem regiunea I: $\boxed{b^2 \cdot x < a^2 \cdot y}$

Variabila de decizie va fi

$d = F(x_P + 1, y_P - \frac{1}{2}) = b^2 \cdot x_p^2 + a^2 \cdot y_p^2 - a^2 \cdot b^2 + 2b^2 \cdot x_p - a^2 \cdot y_p + \frac{a^2}{4}$

• dacă $d = F(M) < 0$, M este în interior, atunci alegem E

$d_{new} = F(x_P + 2, y_P - \frac{1}{2}) =$

$b^2 \cdot x_P^2 + a^2 \cdot y_P^2 - a^2 \cdot b^2 + 4b^2 \cdot x_P - a^2 \cdot y_P + 4b^2 + \frac{a^2}{4}$

$\rightsquigarrow d_{new} = d + \underbrace{2b^2 \cdot x_P + 3b^2}_{deltaE}$ **(d += deltaE)**

• dacă $d = F(M) > 0$, M este în exterior, alegem SE

$d_{new} = F(x_P + 2, y_P - \frac{3}{2}) =$

$b^2 \cdot x_P^2 + a^2 \cdot y_P^2 - a^2 \cdot b^2 + 4b^2 \cdot x_P - 3a^2 \cdot y_P + 4b^2 + \frac{9a^2}{4}.$

$\rightsquigarrow d_{new} = d + \underbrace{2b^2 \cdot x_P - 2a^2 \cdot y_P + 3b^2 + 2a^2}_{deltaSE}$ **(d += deltaSE)**

Algoritmul da Silva

Să studiem regiunea I: $\boxed{b^2 \cdot x < a^2 \cdot y}$

Variabila de decizie va fi

$$d = F(x_P + 1, y_P - \tfrac{1}{2}) = b^2 \cdot x_p^2 + a^2 \cdot y_p^2 - a^2 \cdot b^2 + 2b^2 \cdot x_p - a^2 \cdot y_p + \tfrac{a^2}{4}$$

- dacă $d = F(M) < 0$, M este în interior, atunci alegem E

$$d_{new} = F(x_P + 2, y_P - \tfrac{1}{2}) =$$
$$b^2 \cdot x_P^2 + a^2 \cdot y_P^2 - a^2 \cdot b^2 + 4b^2 \cdot x_P - a^2 \cdot y_P + 4b^2 + \tfrac{a^2}{4}$$
$$\rightsquigarrow d_{new} = d + \underbrace{2b^2 \cdot x_P + 3b^2}_{deltaE} \qquad \textbf{(d += deltaE)}$$

- dacă $d = F(M) > 0$, M este în exterior, alegem SE

$$d_{new} = F(x_P + 2, y_P - \tfrac{3}{2}) =$$
$$b^2 \cdot x_P^2 + a^2 \cdot y_P^2 - a^2 \cdot b^2 + 4b^2 \cdot x_P - 3a^2 \cdot y_P + 4b^2 + \tfrac{9a^2}{4}.$$
$$\rightsquigarrow d_{new} = d + \underbrace{2b^2 \cdot x_P - 2a^2 \cdot y_P + 3b^2 + 2a^2}_{deltaSE} \qquad \textbf{(d += deltaSE)}$$

Algoritmul da Silva

Să studiem regiunea I: $\boxed{b^2 \cdot x < a^2 \cdot y}$

Variabila de decizie va fi

$$d = F(x_P + 1, y_P - \tfrac{1}{2}) = b^2 \cdot x_p^2 + a^2 \cdot y_p^2 - a^2 \cdot b^2 + 2b^2 \cdot x_p - a^2 \cdot y_p + \tfrac{a^2}{4}$$

• dacă $d = F(M) < 0$, M este în interior, atunci alegem E

$$d_{new} = F(x_P + 2, y_P - \tfrac{1}{2}) =$$
$$b^2 \cdot x_P^2 + a^2 \cdot y_P^2 - a^2 \cdot b^2 + 4b^2 \cdot x_P - a^2 \cdot y_P + 4b^2 + \tfrac{a^2}{4}$$
$$\rightsquigarrow d_{new} = d + \underbrace{2b^2 \cdot x_P + 3b^2}_{deltaE} \qquad \textbf{(d += deltaE)}$$

• dacă $d = F(M) > 0$, M este în exterior, alegem SE

$$d_{new} = F(x_P + 2, y_P - \tfrac{3}{2}) =$$
$$b^2 \cdot x_P^2 + a^2 \cdot y_P^2 - a^2 \cdot b^2 + 4b^2 \cdot x_P - 3a^2 \cdot y_P + 4b^2 + \tfrac{9a^2}{4}.$$
$$\rightsquigarrow d_{new} = d + \underbrace{2b^2 \cdot x_P - 2a^2 \cdot y_P + 3b^2 + 2a^2}_{deltaSE} \qquad \textbf{(d += deltaSE)}$$

Algoritmul da Silva

Să studiem regiunea I: $\boxed{b^2 \cdot x < a^2 \cdot y}$

Variabila de decizie va fi

$$d = F(x_P+1, y_P-\tfrac{1}{2}) = b^2 \cdot x_p^2 + a^2 \cdot y_p^2 - a^2 \cdot b^2 + 2b^2 \cdot x_p - a^2 \cdot y_p + \frac{a^2}{4}$$

• dacă $d = F(M) < 0$, M este în interior, atunci alegem E

$d_{new} = F(x_P + 2, y_P - \frac{1}{2}) =$

$b^2 \cdot x_P^2 + a^2 \cdot y_P^2 - a^2 \cdot b^2 + 4b^2 \cdot x_P - a^2 \cdot y_P + 4b^2 + \frac{a^2}{4}$

$\rightsquigarrow d_{new} = d + \underbrace{2b^2 \cdot x_P + 3b^2}_{deltaE}$ **(d += deltaE)**

• dacă $d = F(M) > 0$, M este în exterior, alegem SE

$d_{new} = F(x_P + 2, y_P - \frac{3}{2}) =$

$b^2 \cdot x_P^2 + a^2 \cdot y_P^2 - a^2 \cdot b^2 + 4b^2 \cdot x_P - 3a^2 \cdot y_P + 4b^2 + \frac{9a^2}{4}.$

$\rightsquigarrow d_{new} = d + \underbrace{2b^2 \cdot x_P - 2a^2 \cdot y_P + 3b^2 + 2a^2}_{deltaSE}$ **(d += deltaSE)**

Algoritmul da Silva

Să studiem regiunea I: $\boxed{b^2 \cdot x < a^2 \cdot y}$

Variabila de decizie va fi

$$d = F(x_P + 1, y_P - \tfrac{1}{2}) = b^2 \cdot x_p^2 + a^2 \cdot y_p^2 - a^2 \cdot b^2 + 2b^2 \cdot x_p - a^2 \cdot y_p + \frac{a^2}{4}$$

• dacă $d = F(M) < 0$, M este în interior, atunci alegem E

$d_{new} = F(x_P + 2, y_P - \frac{1}{2}) =$

$b^2 \cdot x_P^2 + a^2 \cdot y_P^2 - a^2 \cdot b^2 + 4b^2 \cdot x_P - a^2 \cdot y_P + 4b^2 + \frac{a^2}{4}$

$$\rightsquigarrow d_{new} = d + \underbrace{2b^2 \cdot x_P + 3b^2}_{deltaE} \qquad \textbf{(d += deltaE)}$$

• dacă $d = F(M) > 0$, M este în exterior, alegem SE

$d_{new} = F(x_P + 2, y_P - \frac{3}{2}) =$

$b^2 \cdot x_P^2 + a^2 \cdot y_P^2 - a^2 \cdot b^2 + 4b^2 \cdot x_P - 3a^2 \cdot y_P + 4b^2 + \frac{9a^2}{4}$.

$$\rightsquigarrow d_{new} = d + \underbrace{2b^2 \cdot x_P - 2a^2 \cdot y_P + 3b^2 + 2a^2}_{deltaSE} \qquad \textbf{(d += deltaSE)}$$

Algoritmul da Silva

Să calculăm acum prima valoare a lui d.

$$d_{initial} = F(1, b - \tfrac{1}{2}) = b^2 + a^2 \cdot (b - \tfrac{1}{2})^2 - a^2 b^2 =$$
$$= b^2 + a^2 b^2 - a^2 b + \tfrac{a^2}{4} - a^2 b^2 = b^2 - a^2 b + \tfrac{a^2}{4}.$$

Scriem acum creșterile pentru *deltaE* și *deltaSE*:

E: $(x_P + 1, y_P)$

$deltaE_{new} = deltaE + 2b^2$ (**deltaE += 2b²**)

$deltaSE_{new} = deltaSE + 2b^2$ (**deltaSE += 2b²**)

SE: $(x_P + 1, y_P - 1)$

$deltaE_{new} = deltaE + 2b^2$ (**deltaE += 2b²**)

$deltaSE_{new} = deltaSE + 2b^2 + 2a^2$ (**deltaSE += 2b² + 2a²**)

Valorile inițiale sunt: $deltaE_{initial} = 3b^2$

$deltaSE_{initial} = -2a^2 \cdot b + 3b^2 + 2a^2$

Algoritmul da Silva

Să calculăm acum prima valoare a lui d.

$$d_{initial} = F(1, b - \tfrac{1}{2}) = b^2 + a^2 \cdot (b - \tfrac{1}{2})^2 - a^2 b^2 =$$
$$= b^2 + a^2 b^2 - a^2 b + \tfrac{a^2}{4} - a^2 b^2 = b^2 - a^2 b + \tfrac{a^2}{4}.$$

Scriem acum creșterile pentru *deltaE* și *deltaSE*:

E: $(x_P + 1, y_P)$

$deltaE_{new} = deltaE + 2b^2$ **(deltaE += 2b^2)**

$deltaSE_{new} = deltaSE + 2b^2$ **(deltaSE += 2b^2)**

SE: $(x_P + 1, y_P - 1)$

$deltaE_{new} = deltaE + 2b^2$ **(deltaE += 2b^2)**

$deltaSE_{new} = deltaSE + 2b^2 + 2a^2$ **(deltaSE += 2b^2 + 2a^2)**

Valorile inițiale sunt: $deltaE_{initial} = 3b^2$

$deltaSE_{initial} = -2a^2 \cdot b + 3b^2 + 2a^2$

Algoritmul da Silva

Să calculăm acum prima valoare a lui d.

$$d_{initial} = F(1, b - \tfrac{1}{2}) = b^2 + a^2 \cdot (b - \tfrac{1}{2})^2 - a^2 b^2 =$$
$$= b^2 + a^2 b^2 - a^2 b + \tfrac{a^2}{4} - a^2 b^2 = b^2 - a^2 b + \tfrac{a^2}{4}.$$

Scriem acum creșterile pentru *deltaE* și *deltaSE*:

E: $(x_P + 1, y_P)$

$deltaE_{new} = deltaE + 2b^2$ **(deltaE += 2b²)**

$deltaSE_{new} = deltaSE + 2b^2$ **(deltaSE += 2b²)**

SE: $(x_P + 1, y_P - 1)$

$deltaE_{new} = deltaE + 2b^2$ **(deltaE += 2b²)**

$deltaSE_{new} = deltaSE + 2b^2 + 2a^2$ **(deltaSE += 2b² + 2a²)**

Valorile inițiale sunt: $deltaE_{initial} = 3b^2$

$deltaSE_{initial} = -2a^2 \cdot b + 3b^2 + 2a^2$

Algoritmul da Silva

Să calculăm acum prima valoare a lui d.

$$d_{initial} = F(1, b - \tfrac{1}{2}) = b^2 + a^2 \cdot (b - \tfrac{1}{2})^2 - a^2 b^2 =$$
$$= b^2 + a^2 b^2 - a^2 b + \tfrac{a^2}{4} - a^2 b^2 = b^2 - a^2 b + \tfrac{a^2}{4}.$$

Scriem acum creșterile pentru *deltaE* și *deltaSE*:

E: $(x_P + 1, y_P)$

$deltaE_{new} = deltaE + 2b^2$ **(deltaE += 2b²)**

$deltaSE_{new} = deltaSE + 2b^2$ **(deltaSE += 2b²)**

SE: $(x_P + 1, y_P - 1)$

$deltaE_{new} = deltaE + 2b^2$ **(deltaE += 2b²)**

$deltaSE_{new} = deltaSE + 2b^2 + 2a^2$ **(deltaSE += 2b² + 2a²)**

Valorile inițiale sunt: $deltaE_{initial} = 3b^2$

$deltaSE_{initial} = -2a^2 \cdot b + 3b^2 + 2a^2$

Algoritmul da Silva

Să calculăm acum prima valoare a lui d.

$$d_{initial} = F(1, b - \tfrac{1}{2}) = b^2 + a^2 \cdot (b - \tfrac{1}{2})^2 - a^2 b^2 =$$
$$= b^2 + a^2 b^2 - a^2 b + \tfrac{a^2}{4} - a^2 b^2 = b^2 - a^2 b + \tfrac{a^2}{4}.$$

Scriem acum creșterile pentru *deltaE* și *deltaSE*:

E: $(x_P + 1, y_P)$

$deltaE_{new} = deltaE + 2b^2$ **(deltaE += 2b^2)**

$deltaSE_{new} = deltaSE + 2b^2$ **(deltaSE += 2b^2)**

SE: $(x_P + 1, y_P - 1)$

$deltaE_{new} = deltaE + 2b^2$ **(deltaE += 2b^2)**

$deltaSE_{new} = deltaSE + 2b^2 + 2a^2$ **(deltaSE += 2b^2 + 2a^2)**

Valorile inițiale sunt: $deltaE_{initial} = 3b^2$

$deltaSE_{initial} = -2a^2 \cdot b + 3b^2 + 2a^2$

Algoritmul da Silva

Să se studieze regiunea a-II-a:

$$\boxed{b^2 \cdot x \geq a^2 \cdot y}$$