Bevezetés a számítógépi grafikába

Raszteres algoritmusok (szakasz, kör)

Troll Ede Mátyás

Matematikai és Informatikai Intézet
Eszterházy Károly Katolikus Egyetem

Eger, 2022

EGER 1774
ESZTERHÁZY KÁROLY EGYETEM

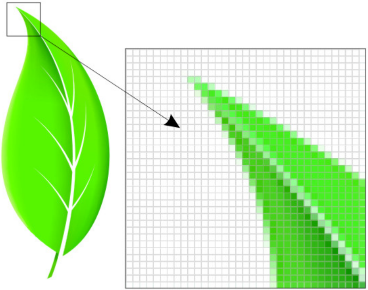
Áttekintés

Áttekintés

Raszteres grafika

A kép sorokból (scanline) és azon belül oszlopokba rendezett képpontokból (pixel) áll. Egy képpont leírásához az alábbi információkat tároljuk:

- Pozíció (x, y koordináták)
- Színinformáció (általában *ARGB*



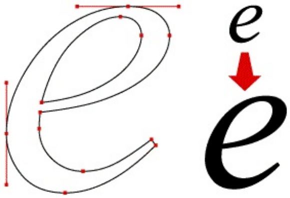
Problémák:

- Képméret
- Rugalmatlanság
- Felbontási probléma (nagyítás $\rightarrow$ lépcsőhatás)

Vektorgrafika

A képet különböző alakzatok leírásával definiáljuk

- Szakasz
- Kör (körív)
- Különböző görbék
- Egyéb alakzatok. . .
- Rasztergrafikus és digitális képek
- Szövegek



Szabvány: SVG (Scalable Vector Graphics)

- XML alapú leírónyelv
- W3C által definiált nyílt szabvány
- Képes animációk, eseménykezelés kezelésére

Vektorgrafika

A képet különböző alakzatok leírásával definiáljuk

- Szakasz `<line x1="0" y1="0" x2="0" y2="0" />`
- Kör (körív) `<circle cx="100" cy="100" r="20" />`
- Különböző görbék `<path d="M 10 180 Q 70 110 180 260 500 110 320 230" />`
- Egyéb alakzatok. . .
- Rasztergrafikus és digitális képek
- Szövegek

Szabvány: SVG (Scalable Vector Graphics) http://svg.elte.hu/

- XML alapú leírónyelv
- W3C által definiált nyílt szabvány
- Képes animációk, eseménykezelés kezelésére

Áttekintés

Kívánalmak

- Látszódjon egyenes vonalnak
  - Alapvető kívánalomnak tűnik, de ez csak pontos függőleges, vízszintes és 45°-os meredekség esetén egyértelmű
- Pontos (a kezdőpontból indul és a végpontba érkezik)
  - Ha ez nem teljesül, akkor poligon rajzolásnál "lukak" jelennek meg
  - Később a görbéket is szakaszokkal közelítjük
- Fedettsége legyen állandó
- Legyen gyors

Egyszerű növekményes algoritmus

Induljunk ki az egyenes explicit egyenletéből

$$y = mx + b$$

Ekkor az y tengellyel vett metszéspontja

$$(0, b)$$

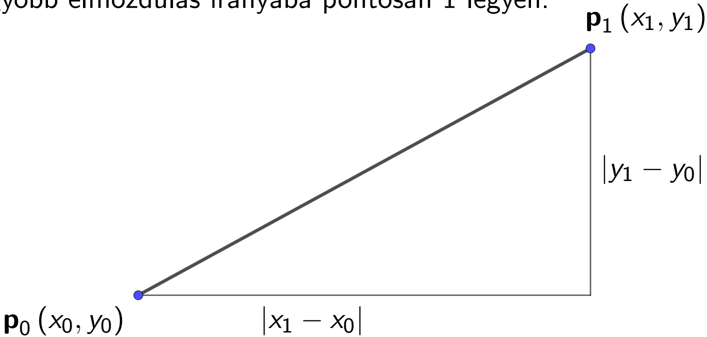
A meredekség segítségével meghatározhatjuk, hogy mennyit lépünk vízszintesen, illetve függőlegesen a következő pontig

$$m = \frac{d_y}{d_x}$$

Az eredmény nem szép, mert az kerekítések miatt lukak lehetnek benne.

DDA

Lényegege, hogy d_x és d_y értékét úgy választjuk meg, hogy a nagyobb elmozdulás irányába pontosan 1 legyen.



DDA

Lényege, hogy d_x és d_y értékét úgy választjuk meg, hogy a nagyobb elmozdulás irányába pontosan 1 legyen.
Legyenek a megrajzolandó szakasz végpontjai $\mathbf{p}_0\,(x_0, y_0)$ és $\mathbf{p}_1\,(x_1, y_1)$. Ekkor

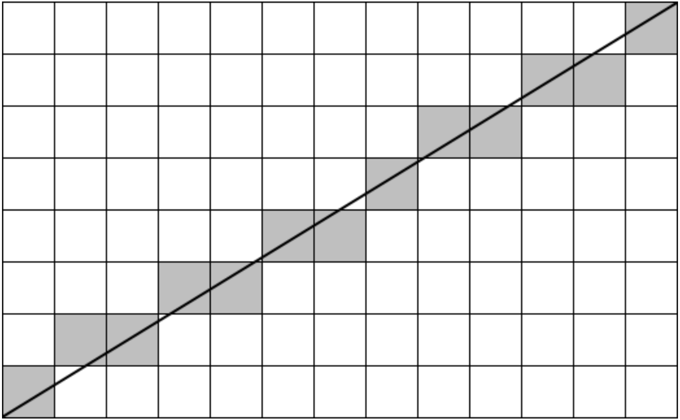
$$d_x = \frac{d_x}{\max\left(|x_1 - x_0|, |y_1 - y_0|\right)}$$
$$d_y = \frac{d_y}{\max\left(|x_1 - x_0|, |y_1 - y_0|\right)}$$

DDA

```
ELJÁRÁS DDA(SZIN S, EGÉSZ X0, EGÉSZ Y0, EGÉSZ X1, EGÉSZ Y1);
    VÁLTOZÓK
        VALÓS: DX, DY, HOSSZ, NX, NY, X, Y;
        EGÉSZ: I;
    ALGORITMUS
        DX <- X1 - X0; DY <- Y1 - Y0;
        HOSSZ <- ABS(DX);
        HA (HOSSZ < ABS(DY)) AKKOR
            HOSSZ <- ABS(DY);
        HA_VÉGE;
        NX <- DX / HOSSZ; X <- X0;
        NY <- DY / HOSSZ; Y <- Y0;
        PIXEL(KEREKÍT(X), KEREKÍT(Y), S);
        CIKLUS I <- 1..HOSSZ
            X <- X + XN;
            Y <- Y + YN;
        PIXEL(KEREKÍT(X), KEREKÍT(Y), S);
        CIKLUS_VÉGE;
ELJÁRÁS_VÉGE;
```

DDA

$\mathbf{p}_1\,(x_1, y_1)$



$\mathbf{p}_0\,(x_0, y_0)$

DDA

Probléma: az algoritmus valós aritmetikát és kerekítést alkalmaz.

MidPoint algoritmus

MidPoint algoritmus

Jack Elton Bresenham 1962-ben az IBM-nél kifejlesztett egy algoritmust, mely javítja a DDA hibáját.

MidPoint algoritmus

Jack Elton Bresenham 1962-ben az IBM-nél kifejlesztett egy algoritmust, mely javítja a DDA hibáját.

Később Pitteway, M.L.W. megalkotta a MidPoint algoritmust, mely bizonyítottan ugyanazokat a pontokat állítja elő, mint az 1962-es Bresenham algoritmus.

MidPoint algoritmus

Jack Elton Bresenham 1962-ben az IBM-nél kifejlesztett egy algoritmust, mely javítja a DDA hibáját.

Később Pitteway, M.L.W. megalkotta a MidPoint algoritmust, mely bizonyítottan ugyanazokat a pontokat állítja elő, mint az 1962-es Bresenham algoritmus.

Az algoritmus lényege, hogy a lehetséges képernyőpontok közül mindig azt választjuk ki, amelyik közelebb helyezkedik el az elméleti egyeneshez.

MidPoint algoritmus

A szakasz legyen adott a $\mathbf{p}_1(x_1; y_1), \mathbf{p}_2(x_2; y_2)$, és induljunk ki az egyenes $y = mx + b$ egyenletéből! Ekkor

$$d_x = x_2 - x_1$$
$$d_y = y_2 - y_1.$$

Ezek alapján

$$m = \frac{d_y}{d_x} = \frac{y_2 - y_1}{x_2 - x_1}.$$

Az egyenesnek az x tengellyel bezárt szögét jelölje α, és tegyük fel, hogy $0 \leq m < 1$. Ekkor tudjuk, hogy $0^\circ \leq \alpha < 45^\circ$, mivel $m = \tan\alpha$.

MidPoint algoritmus

Mivel $0^\circ \leq \alpha < 45^\circ$, ezért tudjuk, hogy $d_x > d_y$.

MidPoint algoritmus

Mivel $0^\circ \leq \alpha < 45^\circ$, ezért tudjuk, hogy $d_x > d_y$.
A növekményes algoritmus minden x irányú lépés után kiszámolta az y irányú növekedést, és egész túlcsordulás esetén kigyújtotta a pontot.

MidPoint algoritmus

Mivel $0^\circ \leq \alpha < 45^\circ$, ezért tudjuk, hogy $d_x > d_y$.
A növekményes algoritmus minden x irányú lépés után kiszámolta az y irányú növekedést, és egész túlcsordulás esetén kigyújtotta a pontot.
Az ideális egyenes azonban valahol két y koordináta között található. A feladatunk az, hogy megkeressük, melyik van kisebb távolságra az egyenestől.

MidPoint algoritmus

Mivel $0° \leq \alpha < 45°$, ezért tudjuk, hogy $d_x > d_y$.
A növekményes algoritmus minden x irányú lépés után kiszámolta az y irányú növekedést, és egész túlcsordulás esetén kigyújtotta a pontot.
Az ideális egyenes azonban valahol két y koordináta között található. A feladatunk az, hogy megkeressük, melyik van kisebb távolságra az egyenestől.
Ahhoz, hogy az egyeneshez való viszonyt tisztázzuk, írjuk fel az egyenest implicit módon!

MidPoint algoritmus

Mivel $0^\circ \leq \alpha < 45^\circ$, ezért tudjuk, hogy $d_x > d_y$.
A növekményes algoritmus minden x irányú lépés után kiszámolta az y irányú növekedést, és egész túlcsordulás esetén kigyújtotta a pontot.
Az ideális egyenes azonban valahol két y koordináta között található. A feladatunk az, hogy megkeressük, melyik van kisebb távolságra az egyenestől.
Ahhoz, hogy az egyeneshez való viszonyt tisztázzuk, írjuk fel az egyenest implicit módon! Mit jelent ez?

MidPoint algoritmus

Mivel $0^\circ \leq \alpha < 45^\circ$, ezért tudjuk, hogy $d_x > d_y$.
A növekményes algoritmus minden x irányú lépés után kiszámolta az y irányú növekedést, és egész túlcsordulás esetén kigyújtotta a pontot.
Az ideális egyenes azonban valahol két y koordináta között található. A feladatunk az, hogy megkeressük, melyik van kisebb távolságra az egyenestől.
Ahhoz, hogy az egyeneshez való viszonyt tisztázzuk, írjuk fel az egyenest implicit módon! Mit jelent ez?

- Explicit: $y = f(x)$

MidPoint algoritmus

Mivel $0° \leq \alpha < 45°$, ezért tudjuk, hogy $d_x > d_y$.
A növekményes algoritmus minden x irányú lépés után kiszámolta az y irányú növekedést, és egész túlcsordulás esetén kigyújtotta a pontot.
Az ideális egyenes azonban valahol két y koordináta között található. A feladatunk az, hogy megkeressük, melyik van kisebb távolságra az egyenestől.
Ahhoz, hogy az egyeneshez való viszonyt tisztázzuk, írjuk fel az egyenest implicit módon! Mit jelent ez?

- Explicit: $y = f(x)$
- Implicit: $F(x, y) = 0$,

MidPoint algoritmus

Mivel $0° \leq \alpha < 45°$, ezért tudjuk, hogy $d_x > d_y$.
A növekményes algoritmus minden x irányú lépés után kiszámolta az y irányú növekedést, és egész túlcsordulás esetén kigyújtotta a pontot.
Az ideális egyenes azonban valahol két y koordináta között található. A feladatunk az, hogy megkeressük, melyik van kisebb távolságra az egyenestől.
Ahhoz, hogy az egyeneshez való viszonyt tisztázzuk, írjuk fel az egyenest implicit módon! Mit jelent ez?

- Explicit: $y = f(x)$
- Implicit: $F(x, y) = 0$, vagy $F(x, y) = c$

MidPoint algoritmus

Mivel $0^\circ \leq \alpha < 45^\circ$, ezért tudjuk, hogy $d_x > d_y$.
A növekményes algoritmus minden x irányú lépés után kiszámolta az y irányú növekedést, és egész túlcsordulás esetén kigyújtotta a pontot.
Az ideális egyenes azonban valahol két y koordináta között található. A feladatunk az, hogy megkeressük, melyik van kisebb távolságra az egyenestől.
Ahhoz, hogy az egyeneshez való viszonyt tisztázzuk, írjuk fel az egyenest implicit módon! Mit jelent ez?

- Explicit: $y = f(x)$
- Implicit: $F(x, y) = 0$, vagy $F(x, y) = c$
- Paraméteres: $\mathbf{r}(t) = \mathbf{p} + t\mathbf{v}$

MidPoint algoritmus

Az egyenes implicit alakja $F(x, y) = Ax + By + C = 0$

MidPoint algoritmus

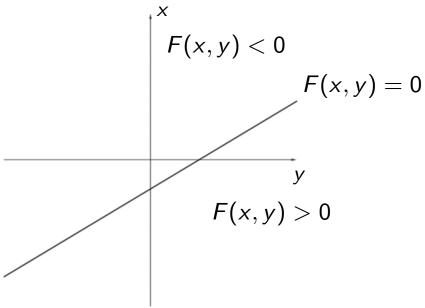
Az egyenes implicit alakja $F(x, y) = Ax + By + C = 0$, ahol

$$A = d_y$$
$$B = -d_x$$
$$C = bd_x$$

MidPoint algoritmus

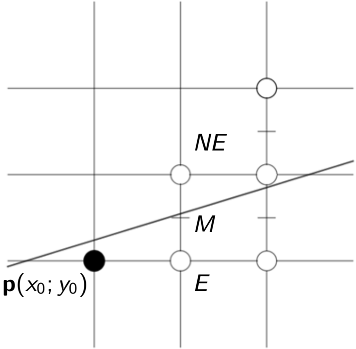
Az egyenes implicit alakja $F(x, y) = Ax + By + C = 0$, ahol

$$A = d_y$$
$$B = -d_x$$
$$C = bd_x$$



MidPoint algoritmus

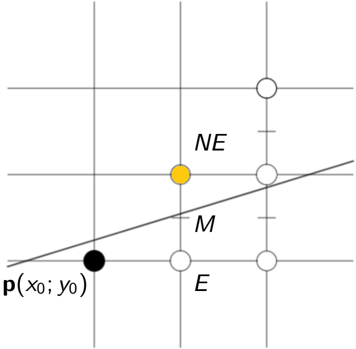
Mivel $0 \leq m < 1$, így amennyiben $d_x = 1$, úgy $d_y < 1$.



MidPoint algoritmus

Mivel $0 \leq m < 1$, így amennyiben $d_x = 1$, úgy $d_y < 1$.

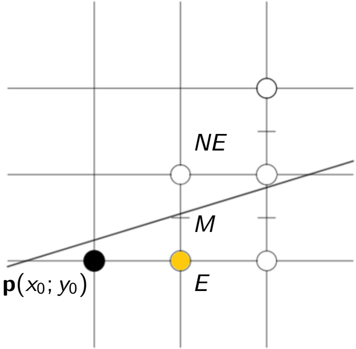
Ha $d_{p+1} = F\left(x_p + 1, y_p + \frac{1}{2}\right) > 0 \rightarrow NE$



MidPoint algoritmus

Mivel $0 \leq m < 1$, így amennyiben $d_x = 1$, úgy $d_y < 1$.

Ha $d_{p+1} = F\left(x_p + 1, y_p + \frac{1}{2}\right) \leq 0 \rightarrow E$



MidPoint algoritmus

Ha minden x szerinti lépés után kiszámítjuk $F(x_{p+1}; y_{p+1})$ értékét, akkor a sok művelet miatt lassú lesz az algoritmus.

MidPoint algoritmus

Ha minden x szerinti lépés után kiszámítjuk $F(x_{p+1}; y_{p+1})$ értékét, akkor a sok művelet miatt lassú lesz az algoritmus.
Megoldás: d_{p+1} ismeretében határozzuk meg d_{p+2} értékét!

MidPoint algoritmus

Ha minden x szerinti lépés után kiszámítjuk $F(x_{p+1}; y_{p+1})$ értékét, akkor a sok művelet miatt lassú lesz az algoritmus.
Megoldás: d_{p+1} ismeretében határozzuk meg d_{p+2} értékét!
Ha x_{p+1} esetén E-t választottuk, akkor

$$d_{p+2} = F\left(x_p + 2, y_p + \frac{1}{2}\right) = A(x_p + 2) + B(y_p + \frac{1}{2}) + C$$

$$d_{p+1} = F\left(x_p + 1, y_p + \frac{1}{2}\right) = A(x_p + 1) + B(y_p + \frac{1}{2}) + C$$

$$d_E = d_{p+2} - d_{p+1} = A = d_y$$

MidPoint algoritmus

Ha minden x szerinti lépés után kiszámítjuk $F(x_{p+1}; y_{p+1})$ értékét, akkor a sok művelet miatt lassú lesz az algoritmus.
Megoldás: d_{p+1} ismeretében határozzuk meg d_{p+2} értékét!
Ha x_{p+1} esetén NE-t választottuk, akkor

$$d_{p+2} = F\left(x_p + 2, y_p + \frac{3}{2}\right) = A(x_p + 2) + B(y_p + \frac{3}{2}) + C$$

$$d_{p+1} = F\left(x_p + 1, y_p + \frac{1}{2}\right) = A(x_p + 1) + B(y_p + \frac{1}{2}) + C$$

$$d_E = d_{p+2} - d_{p+1} = A + B = d_y - d_x$$

MidPoint algoritmus

Meghatároztuk, hogy x szerint lépve hogyan változik a d az előző d_x-re adott döntés alapján. A következő feladat d kezdeti értékének meghatározása.

MidPoint algoritmus

Meghatároztuk, hogy x szerint lépve hogyan változik a d az előző d_x-re adott döntés alapján. A következő feladat d kezdeti értékének meghatározása.

Az első képpont a szakasz egyik végpontja $\mathbf{p}_1(x_1; y_1)$
Az első középpont koordinátái

$$M\left(x_1 + 1; x_1 + \frac{1}{2}\right)$$

MidPoint algoritmus

Meghatároztuk, hogy x szerint lépve hogyan változik a d az előző d_x-re adott döntés alapján. A következő feladat d kezdeti értékének meghatározása.

Az első képpont a szakasz egyik végpontja $\mathbf{p}_1(x_1; y_1)$
Az első középpont koordinátái

$$M\left(x_1 + 1; x_1 + \frac{1}{2}\right)$$

Ez alapján

$$d_0 = F\left(x_1 + 1; y_1 + \frac{1}{2}\right) = Ax_1 + By_1 + C + A + \frac{B}{2} = F(x_1; y_1) + A + \frac{B}{2}$$

MidPoint algoritmus

Mivel $(x_1; y_1)$ rajta van a vonalon, így $F(x_1; y_1) = 0$, tehát

$$d_0 = A + \frac{B}{2} = d_y - \frac{d_x}{2}$$

A törtek elkerüléséhez egyszerűen megszorozzuk a kezdőértéket 2-vel

$$d_0 = 2d_y - d_x$$

MidPoint algoritmus

```
ELJÁRÁS SZAKASZ_MIDPOINT(SZIN S, EGÉSZ X0, EGÉSZ Y0,
                                 EGÉSZ X1, EGÉSZ Y1);
    VÁLTOZÓK
        EGÉSZ: D, DY, DX, X, Y, I;
    ALGORITMUS
        D <- 2 * DY - DX;
        X <- X0;
        Y <- Y0;
        CIKLUS I <- 1..DX
            PIXEL(X, Y, S);
            HA (D > 0) AKKOR
                Y <- Y + 1;
                D <- D + 2 * (DY - DX);
            KÜLÖNBEN
                D <- D + 2 * DY;
            HA_VÉGE;
            X <- X + 1;
        CIKLUS_VÉGE;
ELJÁRÁS_VÉGE;
```

MidPoint algoritmus

Az eljárás jelenleg csak olyan egyenesek esetén működik, melyek x tengellyel bezárt szöge $0^\circ \leq \alpha \leq 45^\circ$ közé esik.

MidPoint algoritmus

Az eljárás jelenleg csak olyan egyenesek esetén működik, melyek x tengellyel bezárt szöge $0^\circ \leq \alpha \leq 45^\circ$ közé esik.

A megoldás technikai jellegű.

MidPoint algoritmus

Az eljárás jelenleg csak olyan egyenesek esetén működik, melyek x tengellyel bezárt szöge $0° \leq \alpha \leq 45°$ közé esik.

A megoldás technikai jellegű. Az alapötlet az, hogy a tárgyalt egyenes alapján 8 különböző helyzetű egyenes különböztethető meg a szakaszok végpontjai alapján.

MidPoint algoritmus

Az eljárás jelenleg csak olyan egyenesek esetén működik, melyek x tengellyel bezárt szöge $0° \leq \alpha \leq 45°$ közé esik.

A megoldás technikai jellegű. Az alapötlet az, hogy a tárgyalt egyenes alapján 8 különböző helyzetű egyenes különböztethető meg a szakaszok végpontjai alapján.

Ebből következik, hogy az egyes koordinátákat esetenként nem növelni, hanem csökkenteni kell 1-gyel. Ezt általában két segédváltozó (SX, SY) bevezetésével oldjuk meg, melyek értéke $0, 1$, vagy -1 lehet.

MidPoint algoritmus

Az eljárás jelenleg csak olyan egyenesek esetén működik, melyek x tengellyel bezárt szöge $0^\circ \leq \alpha \leq 45^\circ$ közé esik.

A megoldás technikai jellegű. Az alapötlet az, hogy a tárgyalt egyenes alapján 8 különböző helyzetű egyenes különböztethető meg a szakaszok végpontjai alapján.

Ebből következik, hogy az egyes koordinátákat esetenként nem növelni, hanem csökkenteni kell 1-gyel. Ezt általában két segédváltozó (SX, SY) bevezetésével oldjuk meg, melyek értéke $0, 1$, vagy -1 lehet.

Továbbá, ha a $m > 1$, akkor d_x és d_y szerepe felcserélendő.

Áttekintés

Körrajzoló algoritmus

Tekintsük az Origó középpontú, R sugarú kör implicit egyenletét

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

Körrajzoló algoritmus

Tekintsük az Origó középpontú, R sugarú kör implicit egyenletét

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

Ebből explicit egyenletet csak úgy kaphatunk, ha két félkörre bontjuk

$$y = \pm\sqrt{R^2 - x^2}$$

Körrajzoló algoritmus

Tekintsük az Origó középpontú, R sugarú kör implicit egyenletét

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

Ebből explicit egyenletet csak úgy kaphatunk, ha két félkörre bontjuk

$$y = \pm\sqrt{R^2 - x^2}$$

Ez alapján elegendő egy félkör pontjait meghatároznunk, a többi tükrözéssel megkapható.

Körrajzoló algoritmus

Tekintsük az Origó középpontú, R sugarú kör implicit egyenletét

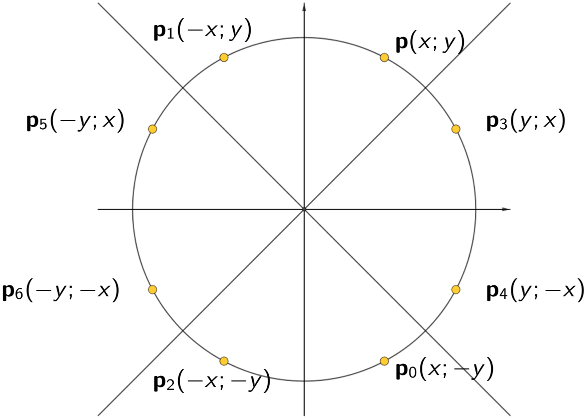
$$x^2 + y^2 = R$$

Ebből explicit egyenletet csak úgy kaphatunk, ha két félkörre bontjuk

$$y = \pm\sqrt{R^2 - x^2}$$

Ez alapján elegendő egy félkör pontjait meghatároznunk, a többi tükrözéssel megkapható.
Ezt tovább gondolva elegendő egy negyed- vagy nyolcad kör pontjait meghatároznunk.

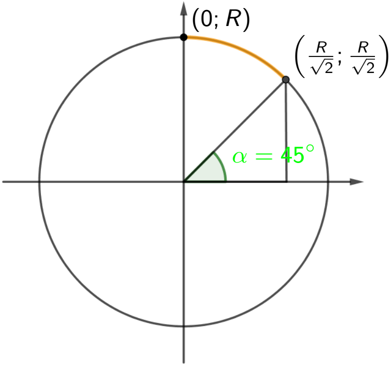
Körrajzoló algoritmus



MidPoint körrajzoló algoritmus

Szintén Bresenham nevéhez fűződik, elve hasonló a szakaszrajzoló algoritmuséhoz.

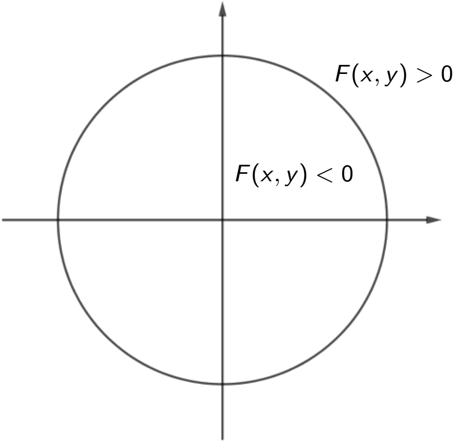
Tekintsük azt az esetet, amikor $x \in \left[0; \frac{R}{\sqrt{2}}\right]$ és $y \in \left[\frac{R}{\sqrt{2}}; R\right]$.



MidPoint körrajzoló algoritmus

Az *Origo* középpontú R sugarú kör implicit egyenlete

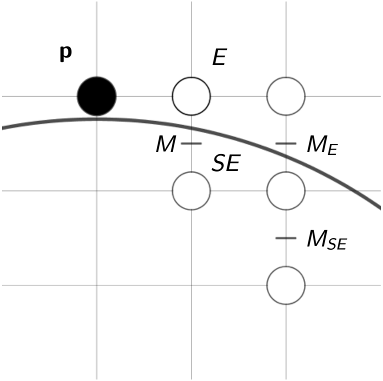
$$F(x, y) = x^2 + y^2 - R^2$$



MidPoint körrajzoló algoritmus

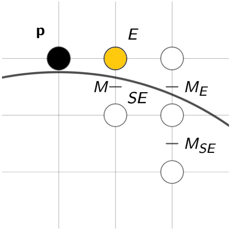
Tegyük fel, hogy egy előző lépésben a **p** $(x_p; y_p)$ pontot választottuk. Ekkor a d_p döntési változó az M pontban

$$d_p = F\left(x_p + 1; y_p - \frac{1}{2}\right) = (x_p + 1)^2 + \left(y_p - \frac{1}{2}\right)^2 - R^2$$



MidPoint körrajzoló algoritmus

Ha $d_p < 0$, tehát az E pontot választottuk, akkor M_E értéke a következőképpen számolható.



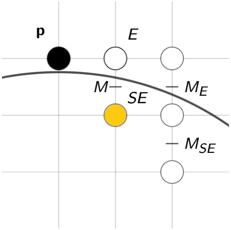
$$d_{p+1} = F\left(x_p + 2, y_p - \frac{1}{2}\right) = (x_p + 2)^2 + \left(y_p - \frac{1}{2}\right)^2 - R^2$$

$$d_{p+1} = d_p + 2x_p + 3$$

Tehát $d_{M_E} = 2x_p + 3$

MidPoint körrajzoló algoritmus

Ha $d_p \leq 0$, tehát az SE pontot választottuk, akkor M_{SE} értéke a következőképpen számolható.



$$d_{p+1} = F\left(x_p + 2, y_p - \frac{3}{2}\right) = (x_p + 2)^2 + \left(y_p - \frac{3}{2}\right)^2 - R^2$$

$$d_{p+1} = d_p + 2x_p - 2y_p + 5$$

Tehát $d_{M_{SE}} = 2x_p - 2y_p + 5$

MidPoint körrajzoló algoritmus

A szakaszrajzoló algoritmushoz hasonlóan már csak a kezdőértéket kell meghatároznunk.
Ehhez d_P egyenletébe helyettesítsük be a kezdőpont $(0; R)$ koordinátáit!

MidPoint körrajzoló algoritmus

A szakaszrajzoló algoritmushoz hasonlóan már csak a kezdőértéket kell meghatároznunk.
Ehhez d_P egyenletébe helyettesítsük be a kezdőpont $(0; R)$ koordinátáit!

$$
\begin{aligned}
d_1 = F\left(1, R - \frac{1}{2}\right) &= 1 + \left(R - \frac{1}{2}\right)^2 - R^2 \\
&= 1 + \left(R^2 - R + \frac{1}{4}\right) - R^2 = \frac{5}{4} - R
\end{aligned}
$$

MidPoint algoritmus

```
ELJÁRÁS MP_KOR_V1(EGÉSZ R, SZIN S);
    VÁLTOZÓK
        EGÉSZ: X, Y;
        VALÓS: D;
    ALGORITMUS
        X <- 0;
        Y <- R;
        D <- 5 / 4 - R;
        KORPONT(X, Y, S);
        CIKLUS_AMÍG (Y > X)
            HA (D < 0) AKKOR
                D <- D + 2 * X + 3;
            KÜLÖNBEN
                D <- D + 2 * (X - Y) + 5;
                Y <- Y - 1;
            HA_VÉGE;
            X <- X + 1;
            KORPONT(X, Y, S);
        CIKLUS_VÉGE;
ELJÁRÁS_VÉGE;
```

MidPoint körrajzoló algoritmus

Sajnos a döntési változó valós.

MidPoint körrajzoló algoritmus

Sajnos a döntési változó valós. A probléma kiküszöbölése egyszerű programtranszformációval történik.

MidPoint körrajzoló algoritmus

Sajnos a döntési változó valós. A probléma kiküszöbölése egyszerű programtranszformációval történik.

- Vezessük be a $H \leftarrow D - 1/4$ döntési változót!

MidPoint körrajzoló algoritmus

Sajnos a döntési változó valós. A probléma kiküszöbölése egyszerű programtranszformációval történik.

- Vezessük be a $H \leftarrow D–1/4$ döntési változót!
- Ezzel D értékét $H + 1/4$–re cseréljük.

MidPoint körrajzoló algoritmus

Sajnos a döntési változó valós. A probléma kiküszöbölése egyszerű programtranszformációval történik.

- Vezessük be a $H \leftarrow D$–$1/4$ döntési változót!
- Ezzel D értékét $H + 1/4$–re cseréljük.
- Az inicializáló érték $H \leftarrow 1$–R lesz.

MidPoint körrajzoló algoritmus

Sajnos a döntési változó valós. A probléma kiküszöbölése egyszerű programtranszformációval történik.

- Vezessük be a $H \leftarrow D$–$1/4$ döntési változót!
- Ezzel D értékét $H + 1/4$–re cseréljük.
- Az inicializáló érték $H \leftarrow 1$–R lesz.
- Ezzel együtt a ciklusmagban a feltétel $H < -1/4$–re változik.

MidPoint körrajzoló algoritmus

Sajnos a döntési változó valós. A probléma kiküszöbölése egyszerű programtranszformációval történik.

- Vezessük be a $H \leftarrow D$–$1/4$ döntési változót!
- Ezzel D értékét $H + 1/4$–re cseréljük.
- Az inicializáló érték $H \leftarrow 1$–R lesz.
- Ezzel együtt a ciklusmagban a feltétel $H < -1/4$–re változik.

Mivel H egész értékről indul, és egész értékkel növekszik, így ezt a feltételt biztonsággal cserélhetjük $H < 0$–ra.

MidPoint algoritmus

```
ELJÁRÁS MP_KOR_V2(EGÉSZ R, SZIN S);
    VÁLTOZÓK
        EGÉSZ: X, Y, H;
    ALGORITMUS
        X <- 0;
        Y <- R;
        H <- 1 - R;
        KORPONT(X, Y, S);
        CIKLUS_AMÍG (Y > X)
            HA (H < 0) AKKOR
                H <- H + 2 *X + 3;
            KÜLÖNBEN
                H <- H + 2 * (X - Y) + 5;
                Y <- Y - 1;
            HA_VÉGE;
            X <- X + 1;
            KORPONT(X, Y, S);
        CIKLUS_VÉGE;
ELJÁRÁS_VÉGE;
```

Köszönöm a figyelmet!