

VÁGÁS

A vágásról általában

Pontok vágása

Vonalak, szakaszok vágása egyenletrendszer megoldásával

A COHEN - SUTHERLAND -féle vonal vágás

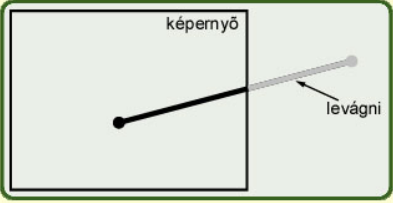
Parametrikus vonal vágó algoritmus

Körök és ellipszisek vágása

Poligonok vágása

VÁGÁS

A primitívekből csak annyit szabad mutatni, amennyi látszik belőlük (takarás, kilógás a képből)



VÁGÁS

Módszerek:

1. Vágjuk le a megjelenítés előtt, azaz számítsuk ki a metszéspontokat és az új végpontokkal rajzoljunk
2. Pásztázzuk a teljes primitívet, de csak a látható képpontokat jelenítjük meg: minden (x, y) -ra ellenőrzés
3. A teljes primitívet munkaterületre rajzoljuk, majd innen átmásoljuk a megfelelő darabot

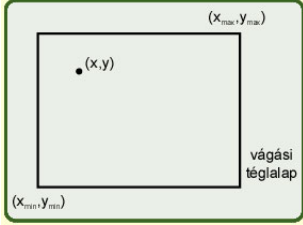
VÁGÁS

Pontok vágása:

(x, y) belül van, ha

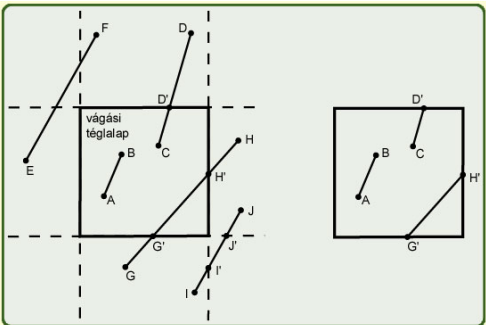
$$x_{min} \leq x \leq x_{max}$$

és

$$y_{min} \leq y \leq y_{max}$$


VÁGÁS

Szakaszok vágása egyenletrendszer megoldásával



Szakaszok vágása egyenletrendszer megoldásával

Elég a végpontokat vizsgálni:

- a) Ha mindkét végpont belül van, akkor a teljes vonal belül van, nincs vágás;
- b) Ha pontosan egy végpont van belül, akkor metszéspontot kell számolni és vágni;
- c) Ha mindkét végpont kívül van, akkor további vizsgálat szükséges: lehet, hogy nincs közös része a vágási téglalappal, de lehet, hogy van.

Szakaszok vágása egyenletrendszer megoldásával

A vágási téglalap minden élére megvizsgáljuk: van-e az élnek közös része a szakasszal
Egyenesek metszéspontjának meghatározása, és az élen belül van-e a metszéspont

Problémák:

Egyenesek (nem szakaszok!) metszéspontjai,
Speciális esetek (vízszintes, függőleges egyenesek)

Szakaszok vágása egyenletrendszer megoldásával

Javítás: parametrikus alak

$$x = x_0 + t \cdot (x_1 - x_0)$$

$$t \in [0, 1] \text{ (szakaszt ír le)}$$

$$y = y_0 + t \cdot (y_1 - y_0)$$

Metszéspont:

$t_{él}$: a metszéspont paramétere az élen

t_{vonal} : a metszéspont paramétere a vonalon

Ha $t_{él}, t_{vonal} \in [0, 1]$, akkor belül van

Még így sem hatékony a módszer, mert sokat kell ellenőrizni és számolni

Szakaszok vágása egyenletrendszer megoldásával

COHEN - SUTHERLAND -féle szakasz vágás

(x_1, y_1) és (x_2, y_2) a szakasz két végpontja.

A végpontok kódolása: Minden végpont annak megfelelő kódot ($code_1, code_2$) kap, hogy melyik tartományban van.

tartományok:	1001	1000	1010	
outkód	0001	0000	0010	
pont	0101	0100	0110	

4 bit

--	--	--	--

$y > y_{max}$ $x > x_{max}$
 $y < y_{min}$ $x < x_{min}$

|| ||
 $y_{max} - y$ $y - y_{min}$ $x_{max} - x$ $x - x_{min}$
 előjele előjele előjele előjele

COHEN - SUTHERLAND -féle szakasz vágás

Előzetes vizsgálatok:

- Ha a végpontok belül vannak, akkor nincs mit vágni (triviális elfogadás)

$$code_1 = code_2 = 0000$$

tartományok:	1001	1000	1010	
outkód	0001	0000	0010	
pont	0101	0100	0110	

4 bit

--	--	--	--

$y > y_{max}$ $x > x_{max}$
 $y < y_{min}$ $x < x_{min}$

|| ||
 $y_{max} - y$ $y - y_{min}$ $x_{max} - x$ $x - x_{min}$
 előjele előjele előjele előjele

COHEN - SUTHERLAND -féle szakasz vágás

különben:

- ha $x_1, x_2 < x_{min}$ (...1.)
 vagy $x_1, x_2 > x_{max}$ (.1..) minden kívül van
 vagy $y_1, y_2 < y_{min}$ (.1..) (triviális elvetés)
 vagy $y_1, y_2 > y_{max}$ (1...)

$$code_1 \text{ AND } code_2 = \text{TRUE} \text{ (bitenként)}$$

1001	1000	1010
0001	0000	0010
0101	0100	0110

COHEN - SUTHERLAND -féle szakasz vágás

különben:

- az $(x_1, y_1) - (x_2, y_2)$ szakasz metszi valamelyik élet. Vegyünk egy külső végpontot (legalább egyik az; ha több van, akkor válasszuk közülük felülről lefelé és jobbról balra haladva az elsőt), számítsuk ki a metszéspontot.
 A két részre vágott szakasz egyik fele a 2. pont alapján triviálisan elvethető.

COHEN - SUTHERLAND - féle szakasz vágás

Interaktív módon is használható

Hatékony, mert gyakori, hogy sok vagy kevés szakasz van belül

A legáltalánosabban használt eljárás

Parametrikus szakasz vágó algoritmus

$P(t) = P_0 + \frac{(P_1 - P_0)}{D} t = P_0 + D t$

A metszéspontra (skalárszorzat):

$$N_i (P(t) - P_{Ei}) = 0$$

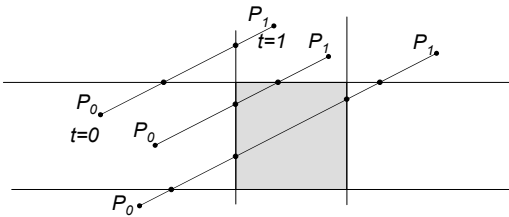
$$N_i (P_0 + D t - P_{Ei}) = 0$$

$$t = \frac{N_i (P_0 - P_{Ei})}{-N_i D}$$

Ha $N_i D$ $\begin{cases} < 0, \text{ akkor belép a félsíkba,} \\ = 0, \text{ akkor párhuzamos a félsík élével,} \\ > 0, \text{ akkor kilép a félsíkból.} \end{cases}$

Parametrikus szakasz vágó algoritmus

Meghatározható az egyenesnek a téglalap 4 élével való 4 metszéspontja (4 db t érték).
Melyik két t a megfelelő?



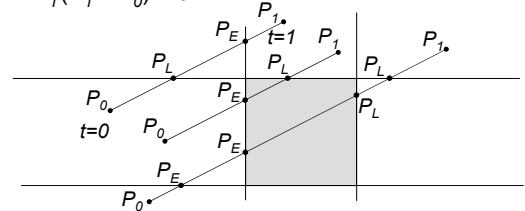
Parametrikus szakasz vágó algoritmus

P_E olyan pont, ahol P_0 -ból P_1 -felé haladva belépünk egy belső félsíkba, ekkor

$$N_i (P_1 - P_0) < 0$$

P_L olyan, ahol kilépünk egy belső félsíkból, ekkor

$$N_i (P_1 - P_0) > 0$$

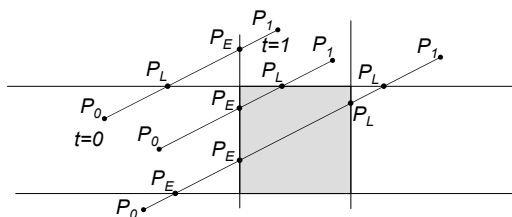


Parametrikus szakasz vágó algoritmus

Legyen

$$t_E = \max \{0, \max\{t_{PE}\}\},$$

$$t_L = \min \{1, \min\{t_{PL}\}\}$$



- ♦ Ha $t_E > t_L$, akkor nincs belső metszés
- ♦ Ha $t_E, t_L \in [0, 1]$, akkor ez belső szakasz

Parametrikus szakasz vágó algoritmus számítása

A metszéspontok számítása:

él _i vágás	N_i	P_{Ei}	$P_0 - P_{Ei}$	$t = \frac{N_i \cdot (P_0 - P_{Ei})}{-N_i \cdot D}$
bal $x = x_{min}$	(-1, 0)	(x_{min}, y)	$(x_0 - x_{min}, y_0 - y)$	$-\frac{(x_0 - x_{min})}{(x_1 - x_0)}$
jobb $x = x_{max}$	(1, 0)	(x_{max}, y)	$(x_0 - x_{max}, y_0 - y)$	$\frac{(x_0 - x_{max})}{-(x_1 - x_0)}$
lent $y = y_{min}$	(0, -1)	(x, y_{min})	$(x_0 - x, y_0 - y_{min})$	$-\frac{(y_0 - y_{min})}{(y_1 - y_0)}$
fent $y = y_{max}$	(0, 1)	(x, y_{max})	$(x_0 - x, y_0 - y_{max})$	$\frac{(y_0 - y_{max})}{-(y_1 - y_0)}$

Parametrikus szakasz vágó algoritmus

```

begin
   $N_i$  kiszámítása,  $P_{E_i}$  kiválasztása minden élre;
  for szakaszokra
    if  $P_1 = P_0$  then pont vágása;
    else begin
       $t_E = 0$ ;  $t_L = 1$ ;  $D = P_1 - P_0$ ;
      for él és szakasz párokra
        if  $N_i * D < 0$  then begin
           $t$  kiszámítása.  $N_i * D < 0$ : PE,  $> 0$ : PL;
          if PE then  $t_E = \max(t_E, t)$ ;
          if PL then  $t_L = \min(t_L, t)$ ;
        end;
      if  $t_E > t_L$  then nincs belső metszés
      else  $P(t_E)$ -től  $P(t_L)$ -ig belső metszés
    end
  end
end
    
```

Körök és ellipszisek vágása

Körök és ellipszisek vágása

Triviális vizsgálat:
 Ha a keret belül van, akkor a kör is belül van, nincs mit vágni;
 Ha a keret kívül van, akkor a kör is kívül van, nincs mit vágni.

Különben:
 Körnegyedekre (nyolcadokra) kiszámítjuk a kör és a téglalap élének metszéspontját, utána pásztázás
 Ha a kör nem nagy, akkor pixelenként dönthetünk.

Ellipszis: hasonlóan.

Poligonok vágása

Sok eset lehet:

Általában minden éllel vágni kell

Poligonok vágása

SUTHERLAND, HODGMAN:

vágjunk egyenként az összes éllel

(V_1, V_2, \dots, V_n) csúcspontok $\xrightarrow{\text{algoritmus}}$ $(V'_1, V'_2, \dots, V'_m)$ új csúcspontok