Algoritm de decupare a liniilor parametrice

Marian Ioan MUNTEANU

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

3 Decembrie 2012

Cuprins

Am văzut că algoritmul Cohen - Sutherland determină intersecțiile segmentului cu dreptele suport ale laturilor dreptunghiului de clipping, substituind valorile uneia dintre extremitățile segmentului cu coordonatele astfel obținute.

Algoritmul Liang - Barsky determină valoarea parametrului t din reprezentarea parametrică a segmentului de dreaptă pentru punctele în care dreapta suport a segmentului intersectează dreptele suport ale laturilor ferestrei, în total $\mathbf{n} = \mathbf{4}$ valori.

Am văzut că algoritmul Cohen - Sutherland determină intersecţiile segmentului cu dreptele suport ale laturilor dreptunghiului de clipping, substituind valorile uneia dintre extremităţile segmentului cu coordonatele astfel obţinute.

Algoritmul Liang - Barsky determină valoarea parametrului t din reprezentarea parametrică a segmentului de dreaptă pentru punctele în care dreapta suport a segmentului intersectează dreptele suport ale laturilor ferestrei, în total $\mathbf{n} = \mathbf{4}$ valori.

Fie P_0, P_1 capetele segmentului cu care dorim să lucrăm.
Un punct oarecare de pe dreapta P_0P_1 se scrie

$$P(t) = P_0 + (P_1 - P_0)t$$

Fie L_i, $(i = 1, 2, 3, 4)$ o latură a dreptunghiului de clipping și N_i normala sa exterioară (îndreptată spre exteriorul dreptunghiului).

Fie P_{L_i} un punct aparținând laturii L_i.
Fie $P(t)$ un punct de pe dreapta P_0P_1.

Fie P_0, P_1 capetele segmentului cu care dorim să lucrăm.
Un punct oarecare de pe dreapta P_0P_1 se scrie

$$P(t) = P_0 + (P_1 - P_0)t$$

Fie L_i, $(i = 1, 2, 3, 4)$ o latură a dreptunghiului de clipping și N_i normala sa exterioară (îndreptată spre exteriorul dreptunghiului).

Fie P_{L_i} un punct aparținând laturii L_i.
Fie $P(t)$ un punct de pe dreapta P_0P_1.

Numim *semiplan interior* corespunzător laturii L_i, acel semiplan determinat de dreapta suport a laturii L_i și care conține fereastra de clipping. Astfel, dacă $P(t)$ este interior (aparține semiplanului interior corespunzător laturii L_i) atunci avem

$$\langle N_i,\ P(t) - P_{L_i} \rangle \ < 0$$

unde $\langle\ ,\ \rangle$ este produsul scalar uzual din $\mathbf{R} \times \mathbf{R}$. Relația are loc întrucât, în acest caz, unghiul dintre cei doi vectori este obtuz. Prin urmare avem:

$$\begin{aligned} P(t) \text{ interior } &\iff \langle N_i, P(t) - P_{L_i} \rangle \ < 0 \\ P(t) \text{ exterior } &\iff \langle N_i, P(t) - P_{L_i} \rangle \ > 0 \\ P(t) \in L_i &\iff \langle N_i, P(t) - P_{L_i} \rangle \ = 0. \end{aligned}$$

Vom nota cu t_i soluția ecuației (ultime). Avem

$$\langle N_i, P(t_i) - P_{L_i}\rangle = 0 \iff \langle N_i, P_0 + (P_1 - P_0)t_i - P_{L_i}\rangle = 0 \iff$$
$$\iff \langle N_i, P_0 - P_{L_i}\rangle \ + \ t_i\langle N_i, P_1 - P_0\rangle \ = 0 \iff$$
$$\iff \mathbf{t_i} = -\frac{\langle \mathbf{N_i}, \mathbf{P_0} - \mathbf{P_{L_i}}\rangle}{\langle \mathbf{N_i}, \mathbf{P_1} - \mathbf{P_0}\rangle}\ .$$

Să analizăm pe rând cele 4 cazuri:

1. L_1 = latura de jos: $y = y_{\min} \rightarrow N_1 = (0, -1)$
Un punct P_{L_1} are coordonatele $(x, y_{\min})$, cu x arbitar
$P_0 - P_{L_1} = (x_0, y_0) - (x, y_{\min}) = (x_0 - x, y_0 - y_{\min})$
$P_1 - P_0 = (x_1 - x_0, y_1 - y_0)$

$$\mathbf{t_1} = -\frac{\langle (0,-1),(x_0 - x, y_0 - y_{\min})\rangle}{\langle (0,-1),(x_1 - x_0, y_1 - y_0)\rangle} = -\frac{(y_{\min} - y_0)}{(y_0 - y_1)} = \frac{(\mathbf{y}_{\min} - \mathbf{y_0})}{(\mathbf{y_1} - \mathbf{y_0})} .$$

2. L_2 = latura din dreapta: $x = x_{\max} \rightarrow N_2 = (1, 0)$
Un punct P_{L_2} are coordonatele $(x_{\max}, y)$, cu y arbitrar
$P_0 - P_{L_2} = (x_0 - x_{\max}, y_0 - y)$

$$\mathbf{t_2} = -\frac{\langle (1,0), (x_0 - x_{\max}, y_0 - y) \rangle}{\langle (1,0), (x_1 - x_0, y_1 - y_0) \rangle} = -\frac{(x_0 - x_{\max})}{(x_1 - x_0)} = \frac{(\mathbf{x_{\max} - x_0})}{(\mathbf{x_1 - x_0})} .$$

3. L_3 = latura de sus: $y = y_{\max} \to N_3 = (0, 1)$
Un punct P_{L_3} are coordonatele $(x, y_{\max})$, cu x arbitrar
$P_0 - P_{L_3} = (x_0 - x, y_0 - y_{\max})$

$$\mathbf{t_3} = -\frac{\langle (0,1), (x_0 - x, y_0 - y_{\max}) \rangle}{\langle (0,1), (x_1 - x_0, y_1 - y_0) \rangle} = -\frac{(y_0 - y_{\max})}{(y_1 - y_0)} = \frac{(\mathbf{y}_{\max} - \mathbf{y_0})}{(\mathbf{y_1} - \mathbf{y_0})}\ .$$

4. L_4 = latura din stânga: $x = x_{\min} \to N_4 = (-1, 0)$
Un punct P_{L_4} are coordonatele $(x_{\min}, y)$, cu y arbitrar
$P_0 - P_{L_4} = (x_0 - x_{\min}, y_0 - y)$

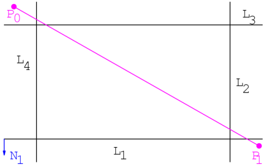
$$\mathbf{t_4} = -\frac{\langle(-1,0),(x_0 - x_{\min}, y_0 - y)\rangle}{\langle(-1,0),(x_1 - x_0, y_1 - y_0)\rangle} = -\frac{(x_{\min} - x_0)}{(x_0 - x_1)} = \frac{(\mathbf{x_{\min} - x_0})}{(\mathbf{x_1 - x_0})} .$$

Se adaugă în listă valorile $t = 0$ și $t = 1$ și se scot din lista valorile lui t care nu aparțin $[0, 1]$ deoarece nu conduc la puncte din interiorul segmentului $(P_0 P_1)$.

Valorile rămase sunt sortate (maxim șase puncte).

Se determină punctele $\mathbf{t_I}$ în care segmentul "intră" în interiorul unui semiplan (impunând condiția $\langle N_i, P_1 - P_0 \rangle < 0$) și punctele $\mathbf{t_E}$ în care segmentul iese în exterior (din condiția $\langle N_i, P_1 - P_0 \rangle > 0$).

Exemplu.



Avem relațiile:
$\langle N_1, P_1 - P_0 \rangle = y_0 - y_1 < 0 \rightarrow L_1 \equiv$ punct de intrare
$\langle N_2, P_1 - P_0 \rangle = x_1 - x_0 < 0 \rightarrow L_2 \equiv$ punct de intrare
$\langle N_3, P_1 - P_0 \rangle = y_1 - y_0 > 0 \rightarrow L_3 \equiv$ punct de ieșire
$\langle N_4, P_1 - P_0 \rangle = x_0 - x_1 > 0 \rightarrow L_4 \equiv$ punct de ieșire.
Se consideră $t = 0$ cel mai mic t_I, iar $t = 1$ cel mai mare t_E.

Segmentul interior ferestrei este definit de cel mai mare t_I și cel mai mic t_E.

Am văzut în exemplul prezentat că produsul $\langle N_i, P_1 - P_0 \rangle$, care determină dacă intersecția este o intrare sau o ieșire din semiplan, se reduce la semnele diferențelor $(x_1 - x_0)$ și $(y_1 - y_0)$; astfel, dacă $dx = (x_1 - x_0)$ este pozitiv, linia trasată de la stânga la dreapta are o intersecție de tip I cu latura stânga și de tip E cu latura din dreapta ferestrei. Analog, dacă $dy > 0$ linia este trasată de jos în sus și prin urmare are o intersecție de tip I cu latura de jos și o intersecție de tip E cu latura de sus.

Înainte de a prezenta funcția C care realizează algoritmul de mai sus, să observăm că numitorii fracțiilor care dau valorile parametrilor t pentru punctele de intersecție sunt exact dx și dy. Mai mult, facem următoarea remarcă ce se va dovedi utilă în contrucția funcției: (în notațiile de mai înainte)

- dacă $(dx > 0)$: L_4 este de tip I, L_2 este de tip E
  dacă $(-dx > 0)$: L_2 este de tip I, L_4 de tip E
  dacă $(dx = 0)$ segmentul este paralel cu Oy
- dacă $(dy > 0)$: L_1 este de tip I, L_3 este de tip E
  dacă $(-dy > 0)$: L_3 este de tip I, L_1 de tip E
  dacă $(dy = 0)$ segmentul este paralel cu Ox.

Definim constantele **TRUE** și **FALSE** (având valorile 1 și respectiv 0) și considerăm variabilele globale (de tip double) tI și tE.

Construim mai întâi o funcție care calculează valoarea parametrului t al punctului de intersecție cu o latură a dreptunghiului, reactualizând valorile tI și tE (ținând cont că $t \in (tI,\ tE)$). Funcția va returna o variabilă booleană (TRUE sau FALSE definite anterior) în funcție de poziția parametrului t față de tI și tE.

```
int Clip_ t (double denom, double num)
{                                   /* denom va fi +/− dx sau +/− dy */
  double t;
  if (denom > 0)
   {                               /* suntem in cazul intersectiilor de tip I */
    t = num / denom;
    if (t > tE) return FALSE;              /* t trebuie sa fie intre tI si tE */
      else if (t > tI) tI = t;                     /* se reactualizeaza tI */
   }
```

```
    else if (denom < 0)
     {                        /* suntem in cazul intersectiilor de tip E */
       t = num / denom;
       if (t < tI) return FALSE;    /* la fel, t trebuie sa fie intre tI si tE */
         else if (t < tE) tE = t;               /* se reactualizeaza tE */
     }
     else     /* cazul denom = 0 inseamna linie paralela cu una din axe */
       if (num > 0) return FALSE;
  return TRUE;
}     /* sfarsit functie Clip_ t */
```

Prezentăm în continuare o funcție care face clipping pentru un punct.

```
int Clip_ point(int x, int y, int xmin, int ymin, int xmax, int ymax)
  {
   if ((x < xmin) || (x > xmax) || (y < ymin) || (y > ymax))
        return FALSE;
    else return TRUE;
  }
```

În final, vom prezenta secvența din **main** care realizează clipping:

```
void main ( )
  {
   float x0, y0, x1, y1;                    /* capetele segmentului P0-P1 */
   float dx, dy;
   float float xmin, ymin, xmax, ymax;
                                  /* coordonatele dreptunghiului de clipping */
   int visible;
   dx = x1 − x0;
   dy = y1 − y0;
   visible = FALSE;
   if ((dx==0) && (dy==0) && Clip_ point(x0,y0,
xmin,ymin,xmax,ymax))
         visible = TRUE;
    else {
      tI = 0;
      tE = 1;
```

```
if (Clip_ t(dx, xmin − x0))
 if (Clip_ t(−dx, x0 − xmax))
  if (Clip_ t(dy, ymin − y0))
   if (Clip_ t(−dy, y0 − ymax))
    {
     visible = TRUE;
     if (tE < 1)                              /* initial este 1 */
      {                /* se calculeaza intersectia de tip E */
       x1 = x0 + tE * dx;
       y1 = y0 + tE * dy;
      }
```

```
        if (tI > 0)                                         /* initial este 0 */
          {                              /* se calculeaza intersectia de tip I */
           x0 += tI * dx;
           y0 += tI * dy;
          }
        }       /* end if - if - if - if */
    }       /* end else */
 if (visible)                         /* se traseaza segmentul intre tI si tE */
  line(x0, y0, x1, y1);
}         /* end main */
```

Atenție! Dacă mai întâi se pune `if (tI>0)...` și apoi `if (tE<1)...` rezultatul va fi greșit. `De ce?` Pentru că în cazul lui `if (tI > 0)` valorile lui x0 și y0 se modifică, după care în corpul lui `if (tE < 1)` s-ar folosi noile valori, ceea ce este greșit.

SFÂRȘITUL PRIMEI PĂRȚI

Atenție! Dacă mai întâi se pune `if (tI>0)...` și apoi `if (tE<1)...` rezultatul va fi greșit. De ce? Pentru că în cazul lui `if (tI > 0)` valorile lui x0 și y0 se modifică, după care în corpul lui `if (tE < 1)` s-ar folosi noile valori, ceea ce este greșit.

SFÂRȘITUL PRIMEI PĂRȚI