

Decuparea vectorilor

Prof. univ. dr. ing. Florica Moldoveanu

Decuparea primitivelor grafice

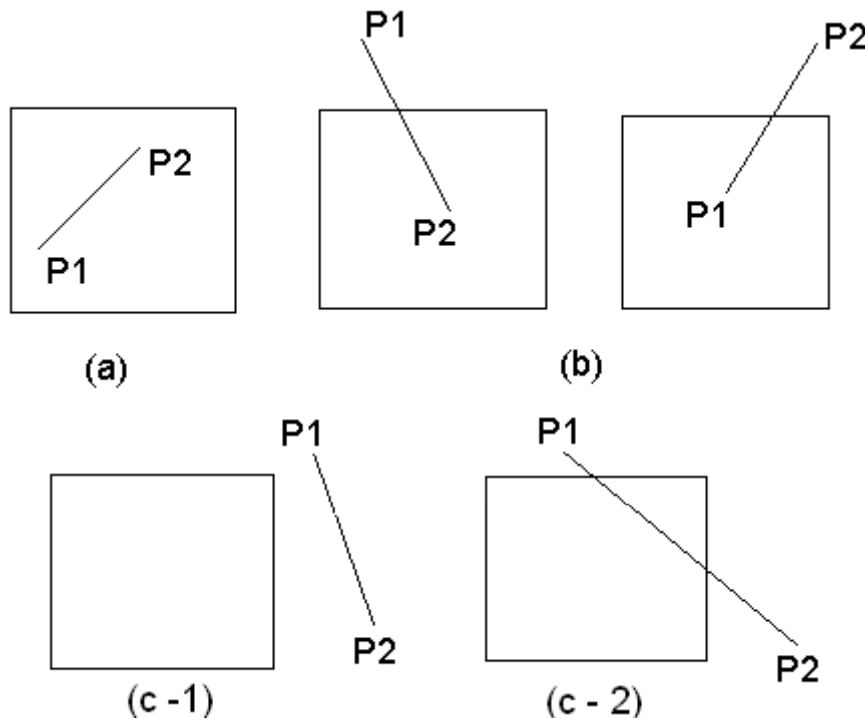
Intr-un sistem grafic 2D:

- Decupare la frontiera unui dreptunghi din spatiul de afisare, avand laturile paralele cu cele ale sistemului de coordonate carteziane 2D (poarta de afisare)
- Decupare la frontiera unui poligon oarecare din spatiul de afisare
- Decuparea se poate efectua:
 - Analitic, prin intersectia primitivelor grafice cu frontiera dreptunghiului/poligonului de decupare, inainte de rasterizare
 - La momentul rasterizarii primitivelor, prin testul de apartenenta a fiecarui fragment la dreptunghiul/poligonul de decupare
- Pentru linii si poligoane, decuparea se efectueaza analitic (mai eficient).

Intr-un sistem grafic 3D:

- Decupare la frontiera volumului vizual canonic, dupa transformarea de proiectie.

Decuparea vectorilor 2D(1)



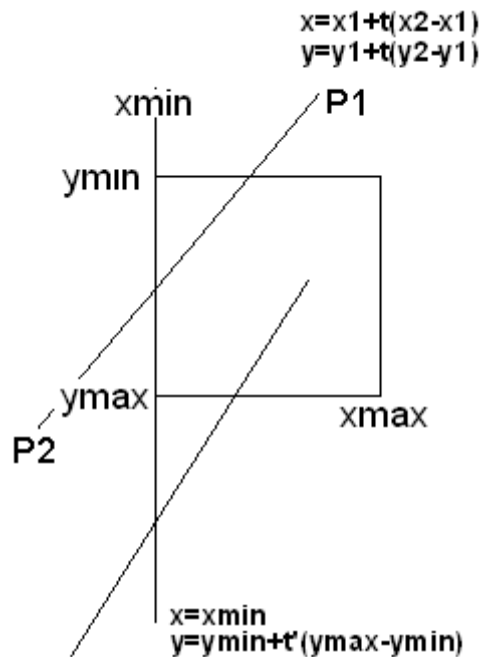
Pozitia vectorului fata de dreptunghiul de decupare

Optimizari: evitarea calculelor de intersectie inutile

- Evitarea calculelor de intersectie in cazurile (a) si (c-1)
- Reducerea calculelor de intersectie in celelalte cazuri, prin evitarea calculelor de intersectie cu laturile neintersectate.

Decuparea vectorilor 2D(2)

Calculul intersecțiilor cu laturile dreptunghiului de decupare



Intersecția segmentului P1-P2 cu latura $x = x_{min}$ a dreptunghiului:

1) $x_{min} = x_1 + t(x_2 - x_1) \rightarrow t = (x_{min} - x_1) / (x_2 - x_1)$

2) este $0 \leq t \leq 1$???

da: punctul de intersecție se afla pe segmentul P1-P2

- se calculează $y_i = y_1 + (x_{min} - x_1) * (y_2 - y_1) / (x_2 - x_1)$

- este $y_{min} \leq y_i \leq y_{max}$??

da: există intersecție între vectorul P1-P2 și latura $x = x_{min}$

Decuparea vectorilor 2D(3)

Algoritmul Cohen-Sutherland(1)

1001	1000	1010
0001	0000	0010
0101	0100	0110

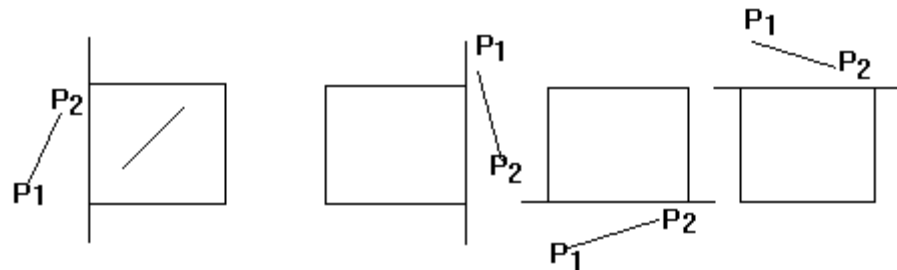
- Punctele din plan sunt codificate prin 4 biti, in functie de pozitia lor fata de dreptunghiul de decupare.

De exemplu: b3b2b1b0

b0=1: stanga, b1=1: dreapta,

b2=1: sub dreptunghi, b3=1: deasupra dreptunghiului

P1: cod1, P2: cod2



- 1) $\text{cod1} == 0 \ \&\& \ \text{cod2} == 0$: segment acceptat trivial

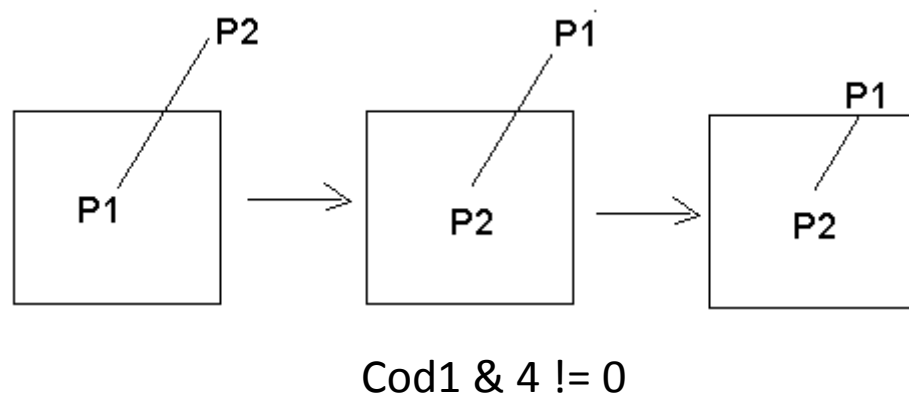
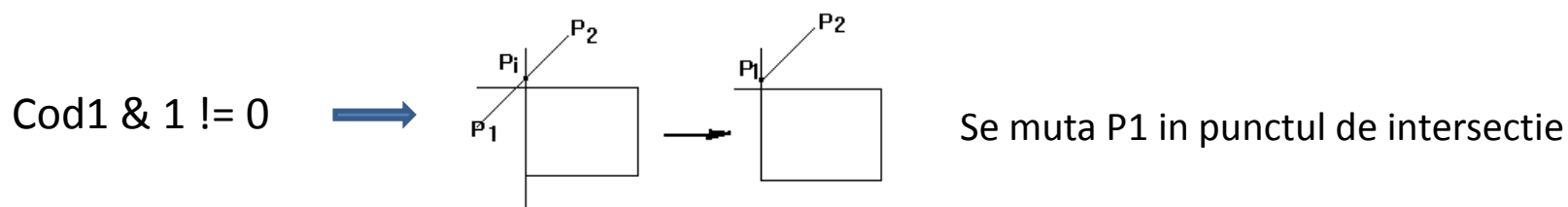
- 2) $(\text{cod1} \ \& \ \text{cod2}) \neq 0$: segment rejectat trivial

- 3) altfel: se intersecteaza segmentul P1-P2 cu dreptunghiul de decupare

Decuparea vectorilor 2D(4)

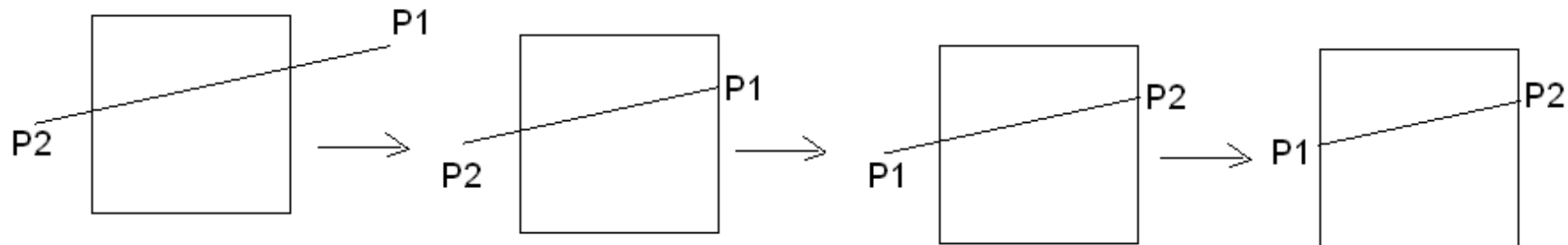
Algoritmul Cohen-Sutherland(2)

- Alegerea laturii cu care se intersecteaza segmentul P1-P2: pe baza codului varfului P1, care trebuie sa fie situat in afara dreptunghiului de decupare



Decuparea vectorilor 2D(5)

Algoritmul Cohen-Sutherland(3)



Calculul punctelor de intersectie

Intersectia cu latura situata pe dreapta $y = y_{\max}$

$y_{\max} = y_1 + t(y_2 - y_1)$ de unde, $t = (y_{\max} - y_1) / (y_2 - y_1)$

$x = x_1 + t(x_2 - x_1) \rightarrow x_i = x_1 + (y_{\max} - y_1) / m$

Intersectia cu latura situata pe dreapta $x = x_{\min}$

$x_{\min} = x_1 + t(x_2 - x_1)$, deci $t = (x_{\min} - x_1) / (x_2 - x_1)$

$y = y_1 + t(y_2 - y_1) \rightarrow y_i = y_1 + (x_{\min} - x_1) * m$

Decuparea vectorilor 2D(6)

Algoritmul Cohen-Sutherland(4)

```
int xmin, ymin, xmax, ymax; // colturile dreptunghiului de decupare
```

```
// functia calculeaza codul binar al unui punct din plan
```

```
int codif(int x, int y)
```

```
{
```

```
    int cod;
```

```
    cod=(x < xmin) ? 1:0;
```

```
    if(x > xmax) cod=2;
```

```
    if(y < ymin) cod |=4;
```

```
    if(y > ymax) cod |=8;
```

```
    return cod;
```

```
}
```


Decuparea vectorilor 2D(7)

Algorimul Cohen-Sutherland (5)

```
int Cohen_Suth(int x1i,int y1i,int x2i,int y2i, int* x1d,int* y1d,int* x2d,int* y2d)  
{ // primeste coord. capetelor vectorului de decupat si intoarce coord. partii incluse in  
    dreptunghi, daca exista  
    int acceptat,rejectat,vertical;  
    float m, x1=x1i, y1=y1i, x2=x2i, y2=y2i;  
    int cod1, cod2, t;  
    acceptat=rejectat=vertical=0;  
    if (x1i != x2i)  
        m=(y2-y1)/(x2-x1);  
    else  
        vertical=1;  
    do  
    { cod1=codif(x1,y1); cod2=codif(x2,y2);  
      if(cod1==0 && cod2==0) { acceptat=1; break;}  
      if((cod1 & cod2) !=0 ) { rejectat=1; break;}
```

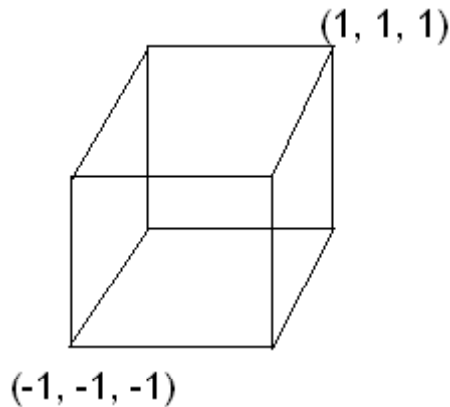
Decuparea vectorilor 2D(8)

```
if(cod1==0) //inversare capete
    {t=x1;x1=x2;x2=t; t=y1;y1=y2;y2=t; cod1=cod2;}
if(cod1 & 1)
    { y1+=(xmin-x1)*m; x1=xmin;}
else
    if(cod1 & 2)
        { y1+=(xmax-x1)*m; x1=xmax; }
    else
        if(cod1 & 4)
            { if (!vertical) x1+=(ymin-y1)/m;
              y1=ymin;
            }
        else
            {if (!vertical) x1+=(ymax-y1)/m;
              y1=ymax;
            }
    }
} while ( acceptat==rejectat);
```

Decuparea vectorilor 2D(9)

```
if(acceptat)
{
    *x1d=x1;*y1d=y1;*x2d=x2;*y2d=y2;
    return 1;
}
return 0;
}
}
```

Generalizarea algoritmului Cohen-Sutherland pentru decuparea vectorilor 3D(1)



Fata de volumul vizual canonic, un punct din spatiu se poate afla: in interior, in stanga, in dreapta, sub volum, deasupra volumului, in fata, in spate:

→ pentru codificarea pozitiei unui punct din spatiu fata de volumul canonic sunt necesari 6 biti.

De ex. se poate face conventia:

$b_0 = 1$ pentru puncte (x,y,z) cu $x < -1$

$b_1 = 1$ pentru puncte (x,y,z) cu $x > 1$

$b_2 = 1$ pentru puncte (x,y,z) cu $y < -1$

$b_3 = 1$ pentru puncte (x,y,z) cu $y > 1$

$b_4 = 1$ pentru puncte (x,y,z) cu $z < -1$

$b_5 = 1$ pentru puncte (x,y,z) cu $z > 1$

daca: $\text{cod}(P_1) == 0 \ \&\& \ \text{cod}(P_2) == 0$
segment acceptat trivial

daca $(\text{cod}(P_1) \ \& \ \text{cod}(P_2)) \neq 0$
segment rejectat trivial

altfel

se intersecteaza segmentul cu volumul

Generalizarea algoritmului Cohen-Sutherland pentru decuparea vectorilor 3D(2)

Intersecțiile segmentului cu volumul vizual

Ec. parametrice ale segmentului

$$x = x_1 + t(x_2 - x_1)$$

$$y = y_1 + t(y_2 - y_1)$$

$$z = z_1 + t(z_2 - z_1)$$

daca $\text{cod}(P_1) \& 8 \neq 0$

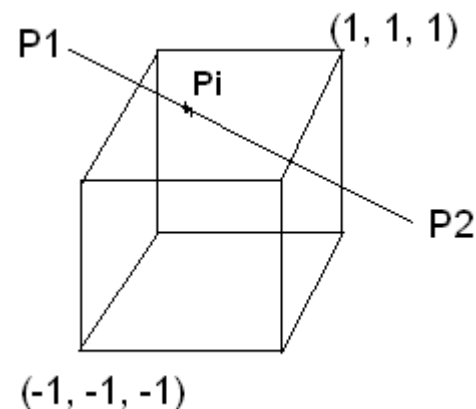
se efectueaza intersectia cu planul $y = 1$:

$$1 = y_1 + t(y_2 - y_1) \rightarrow t_{\text{inters}} = (1 - y_1) / (y_2 - y_1)$$

$$x_i = x_1 + t_{\text{inters}}(x_2 - x_1)$$

$$y_i = 1;$$

$$z_i = z_1 + t_{\text{inters}}(z_2 - z_1)$$



Generalizarea algoritmului Cohen-Sutherland pentru decuparea vectorilor 3D(3)

