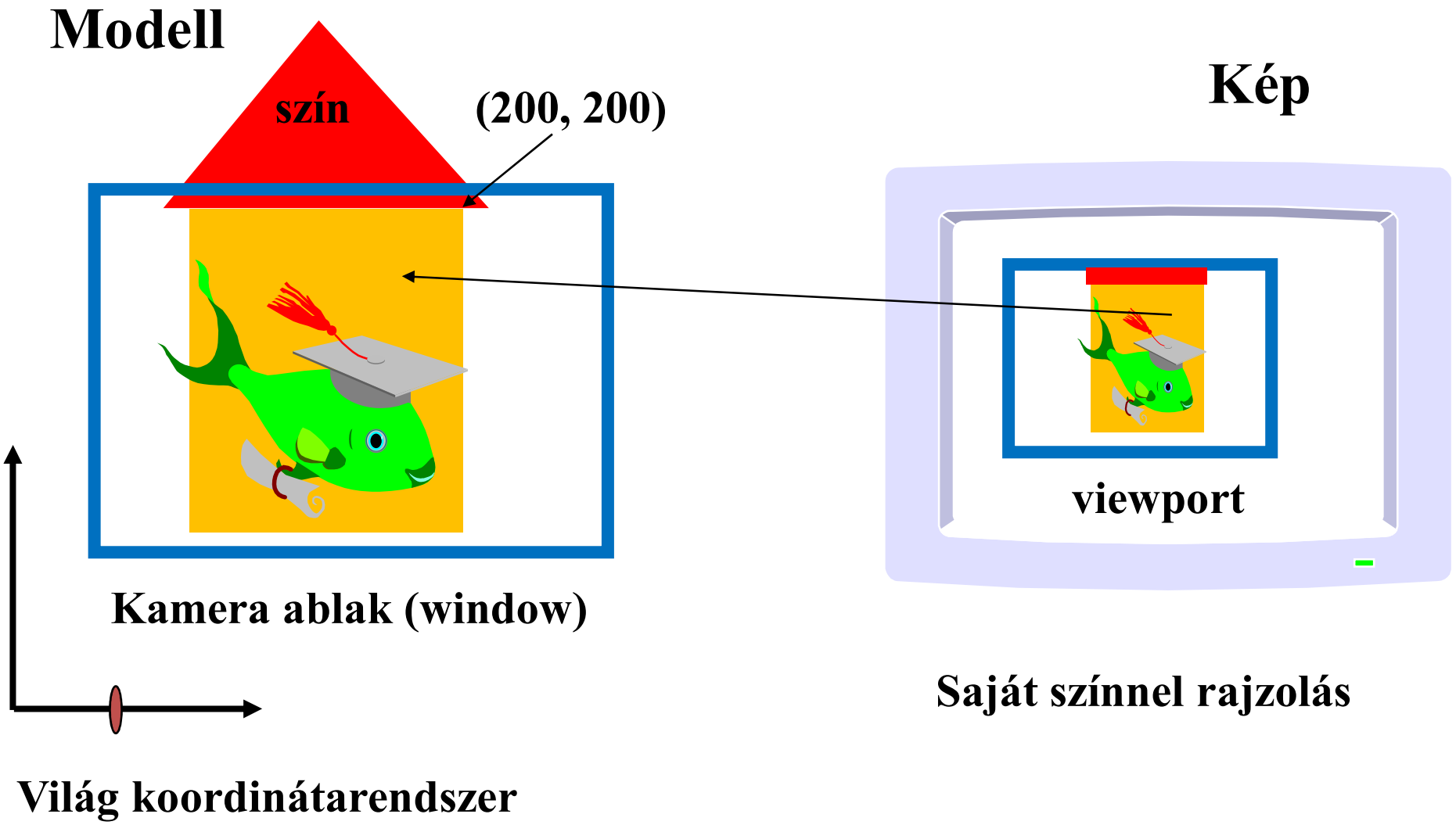
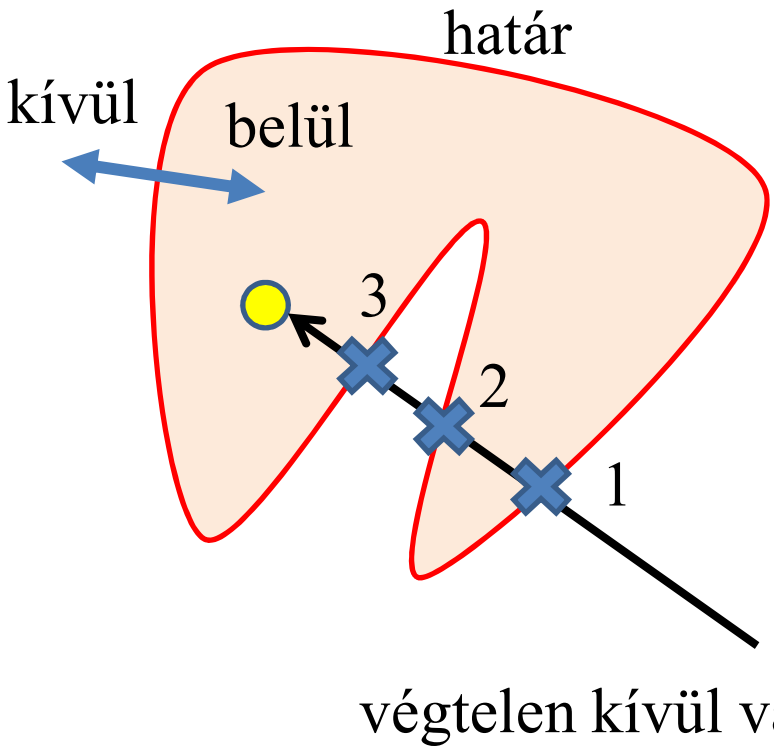
2D képszintézis

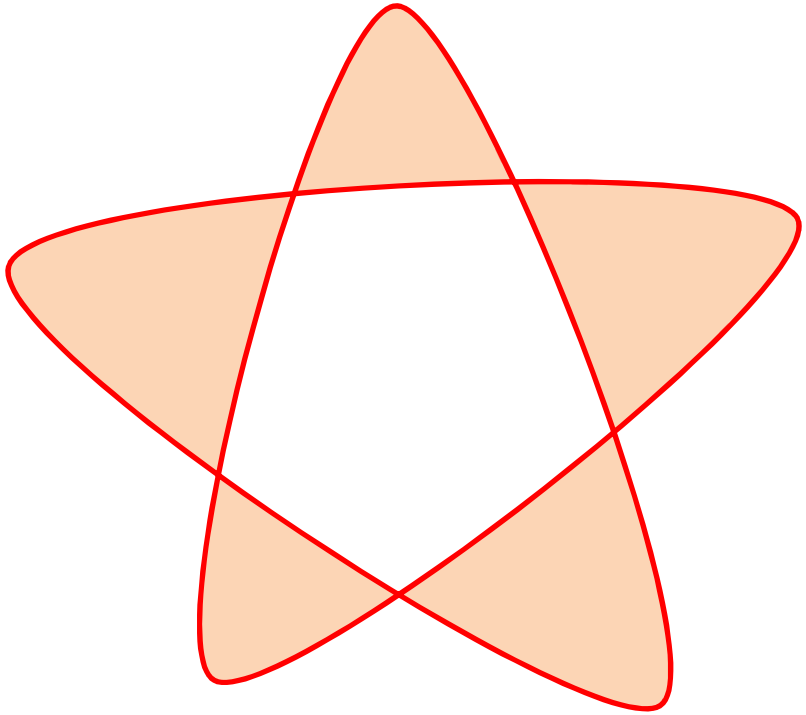
Szirmay-Kalos László

2D képszintézis

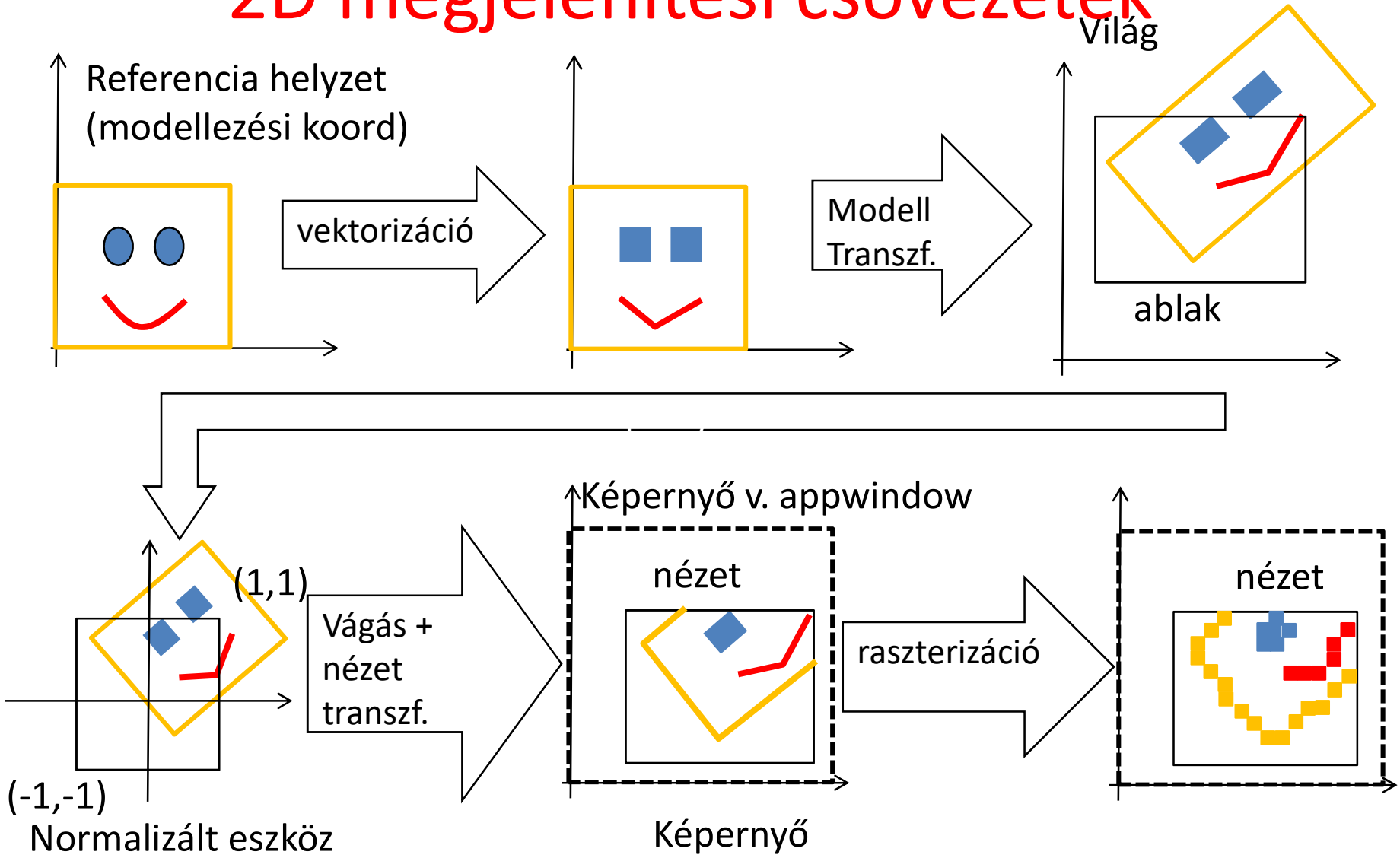


Pixel vezérelt megközelítés: Tartalmazás (objektum, pont)

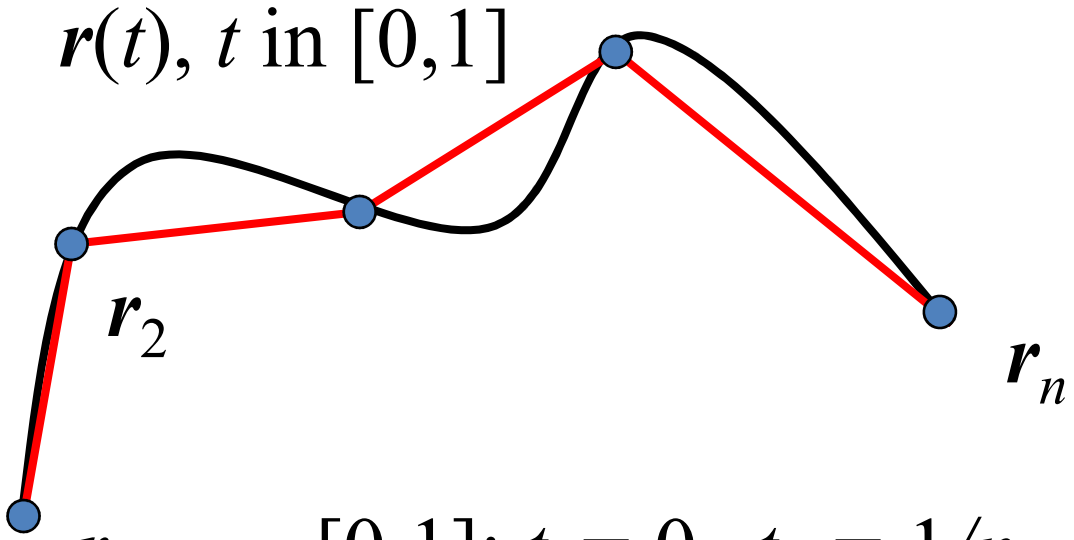




Világ vezérelt megközelítés: 2D megjelenítési csővezeték



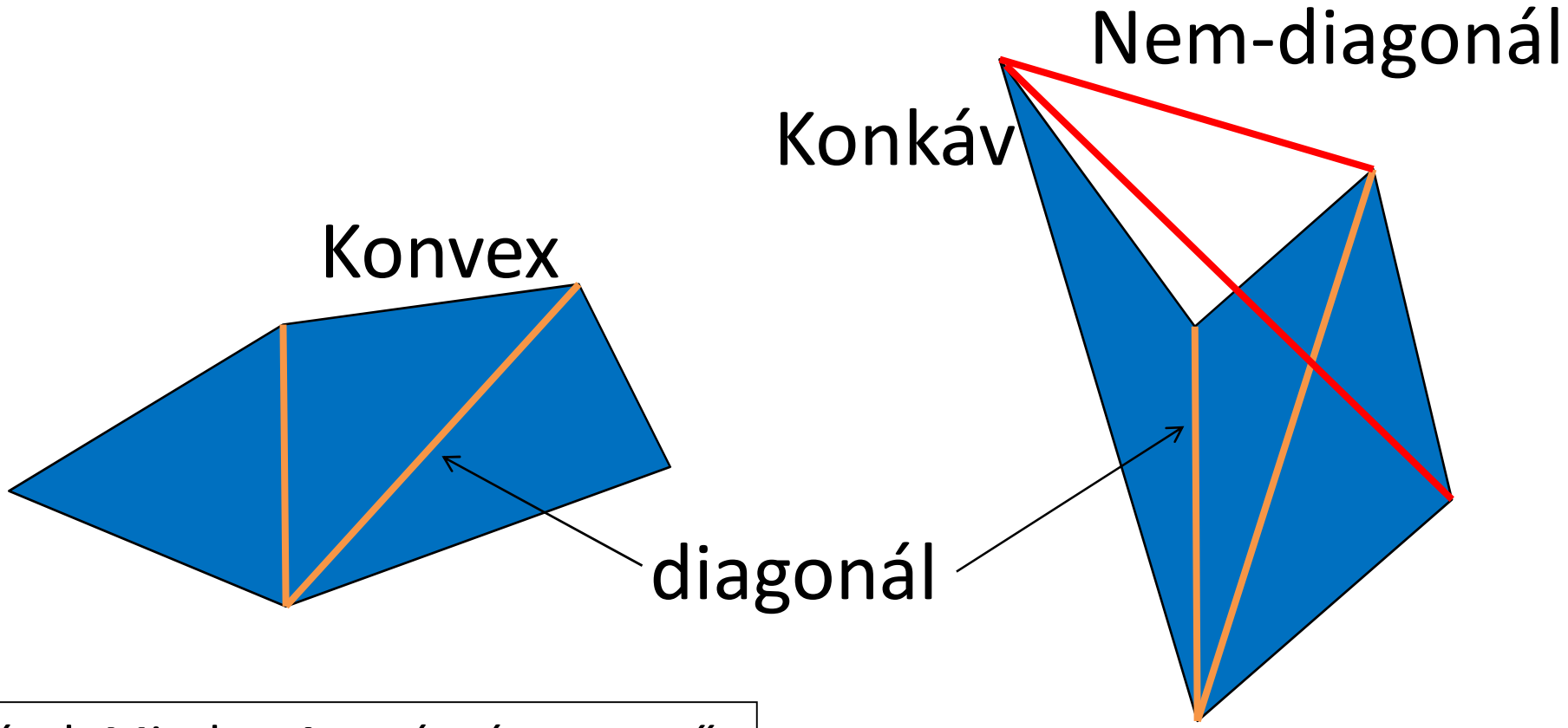
Vektorizáció



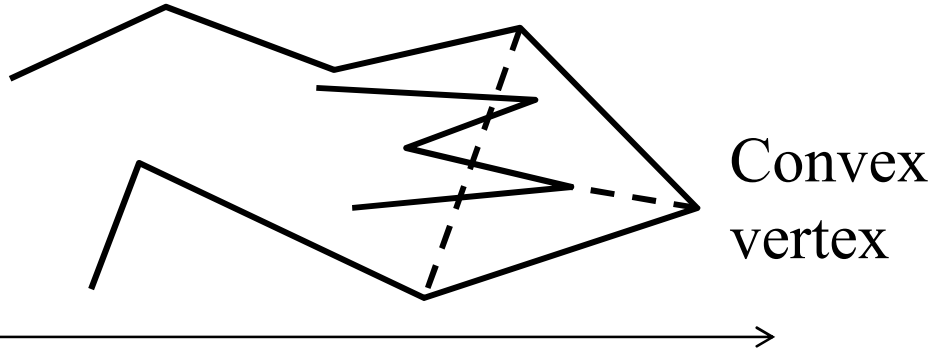
$[0,1]: t_1 = 0,\ t_2 = 1/n, \ldots, t_n = 1$

$\boldsymbol{r}_1 = \boldsymbol{r}(0), \boldsymbol{r}_2 = \boldsymbol{r}(t_2), \ldots, \boldsymbol{r}_n = \boldsymbol{r}(1)$

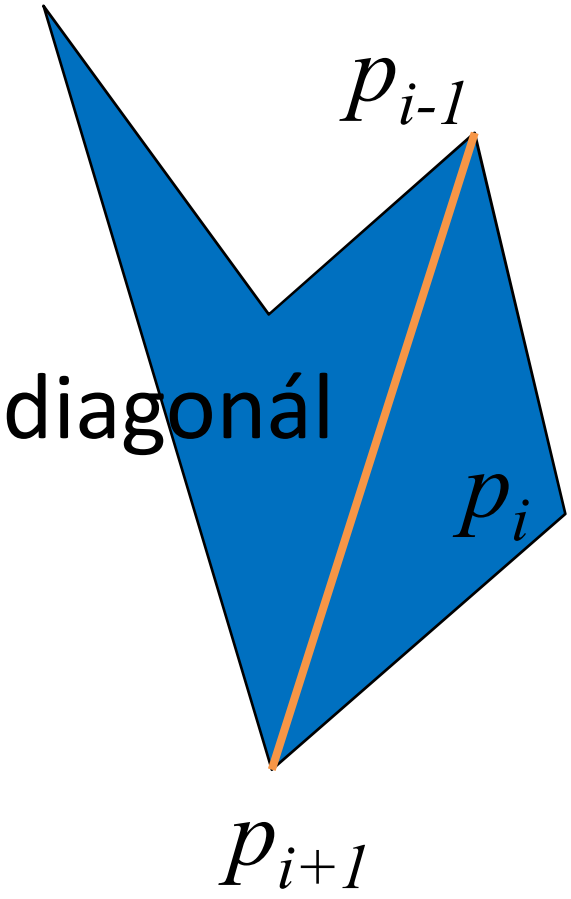
Poligon háromszögekre bontása



Tétel: Minden 4+ csúcsú egyszerű sokszögnek van diagonálja, azaz mindegyik felbontható diagonálok mentén.

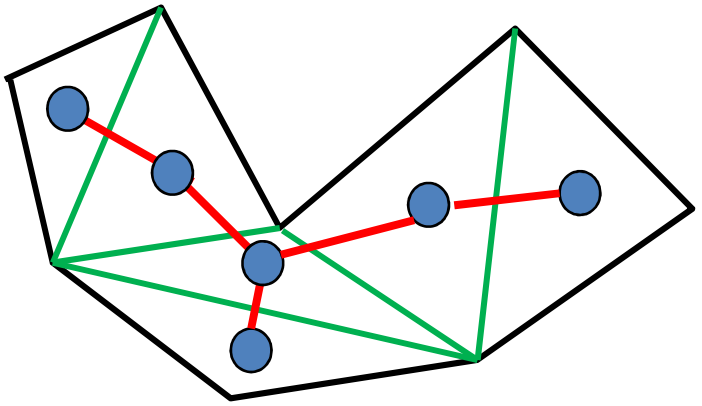


Fül



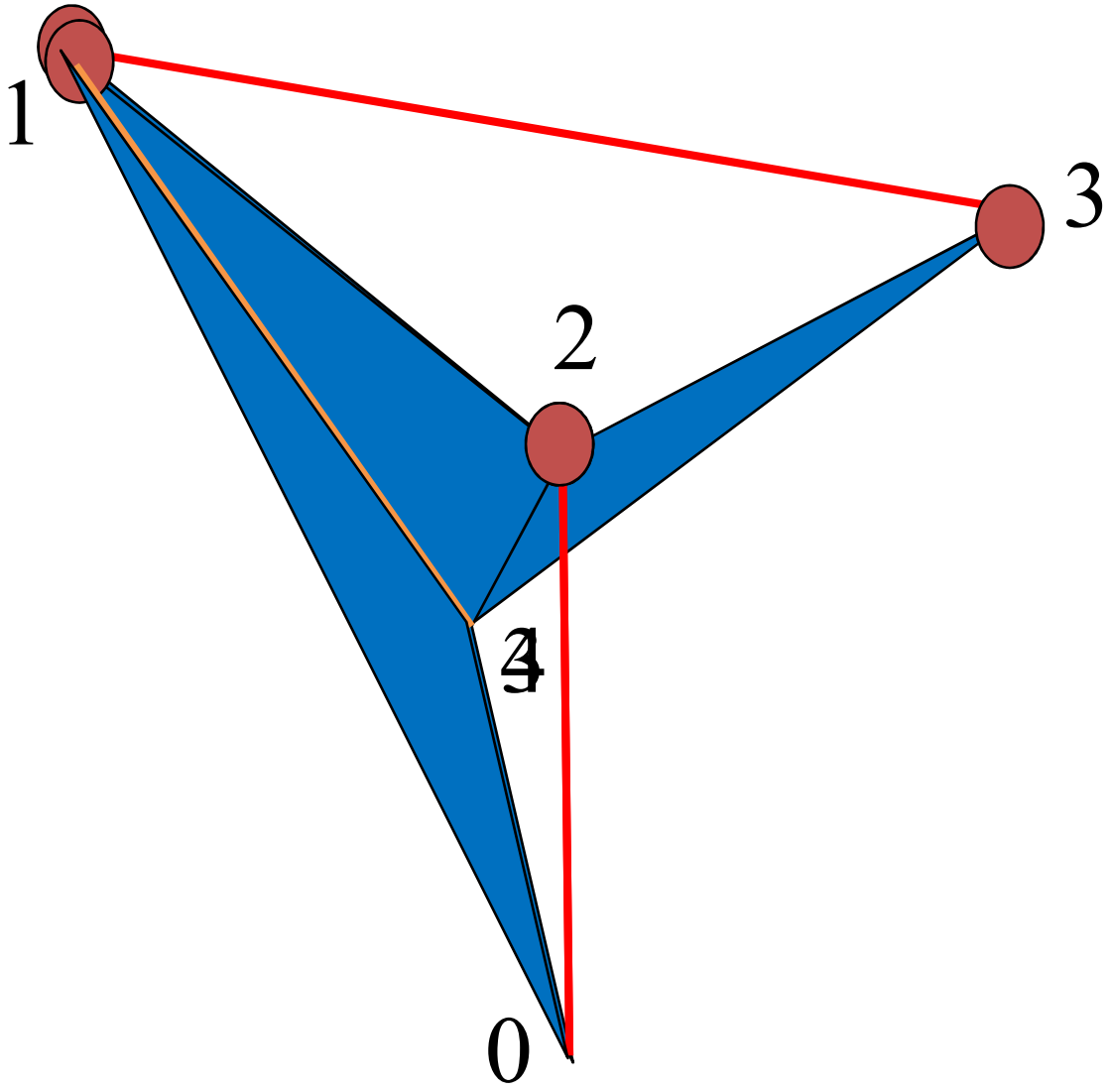
- p_i fül, ha $p_{i-1} - p_{i+1}$ diagonál
- Fül levágható!
- **Fülvágás**:

  keress fület és nyissz!



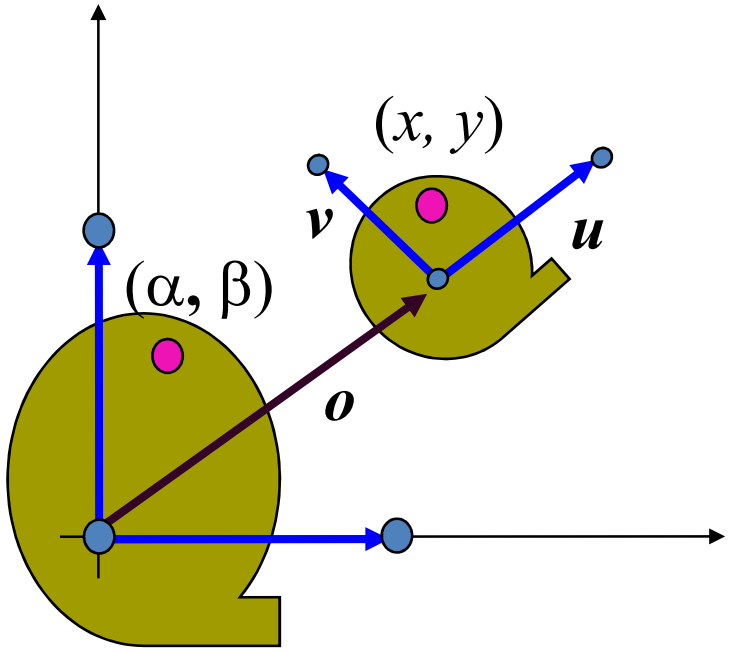
Két fül tétel: Minden legalább 4 csúcsú egyszerű sokszögnek van legalább 2 füle.

Fülvágó algoritmus



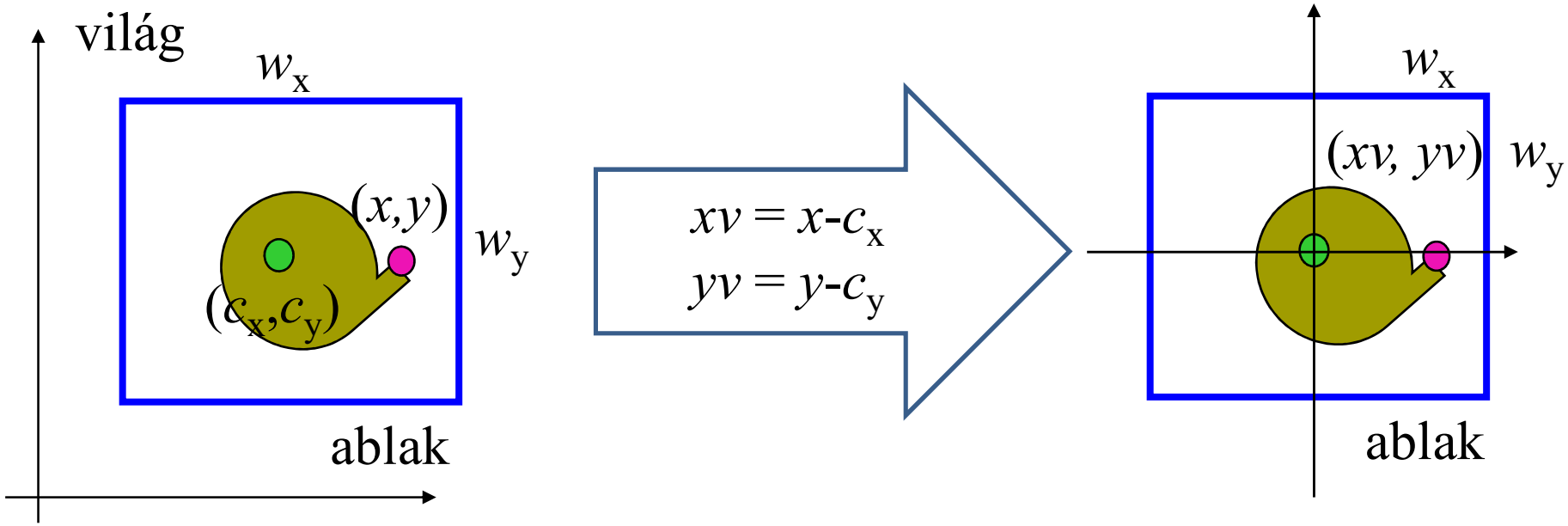
Modellezési transzformáció

$$\begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ p_x & p_y & 1 \end{bmatrix}$$



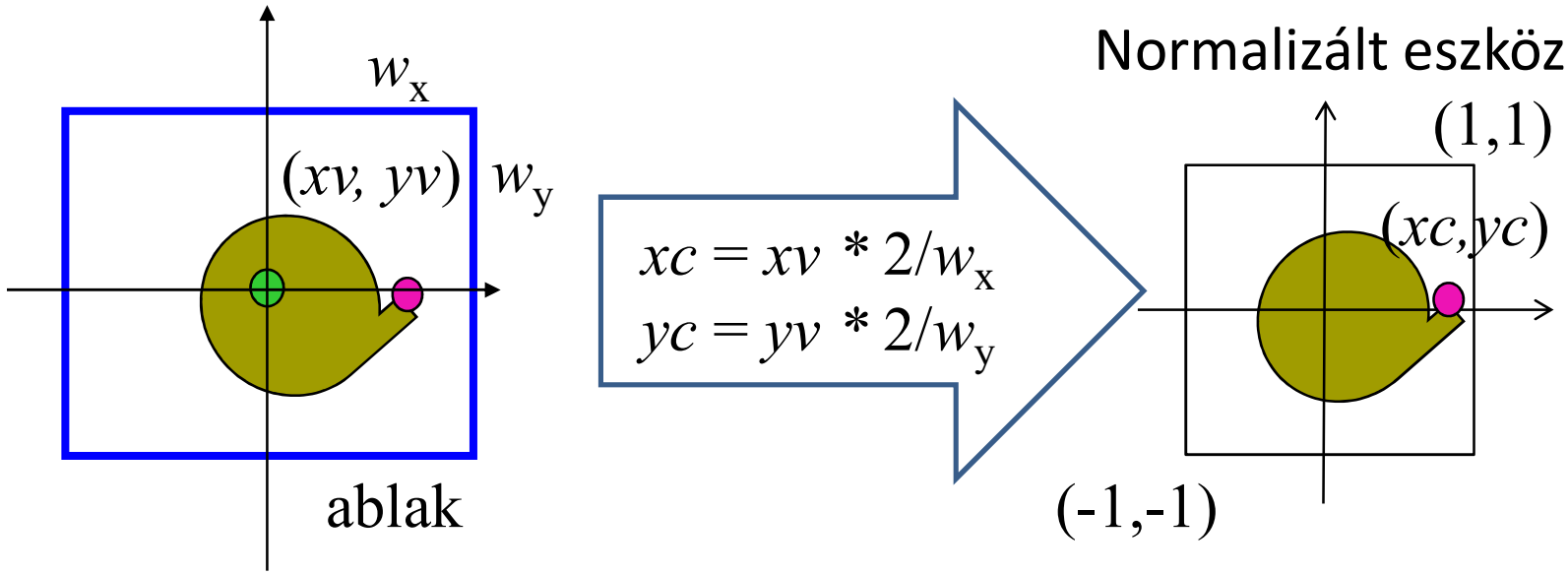
$$(x, y) = \boldsymbol{o} + \alpha\boldsymbol{u} + \beta\boldsymbol{v}$$

View transzformáció: kameraablak közepe az origóba



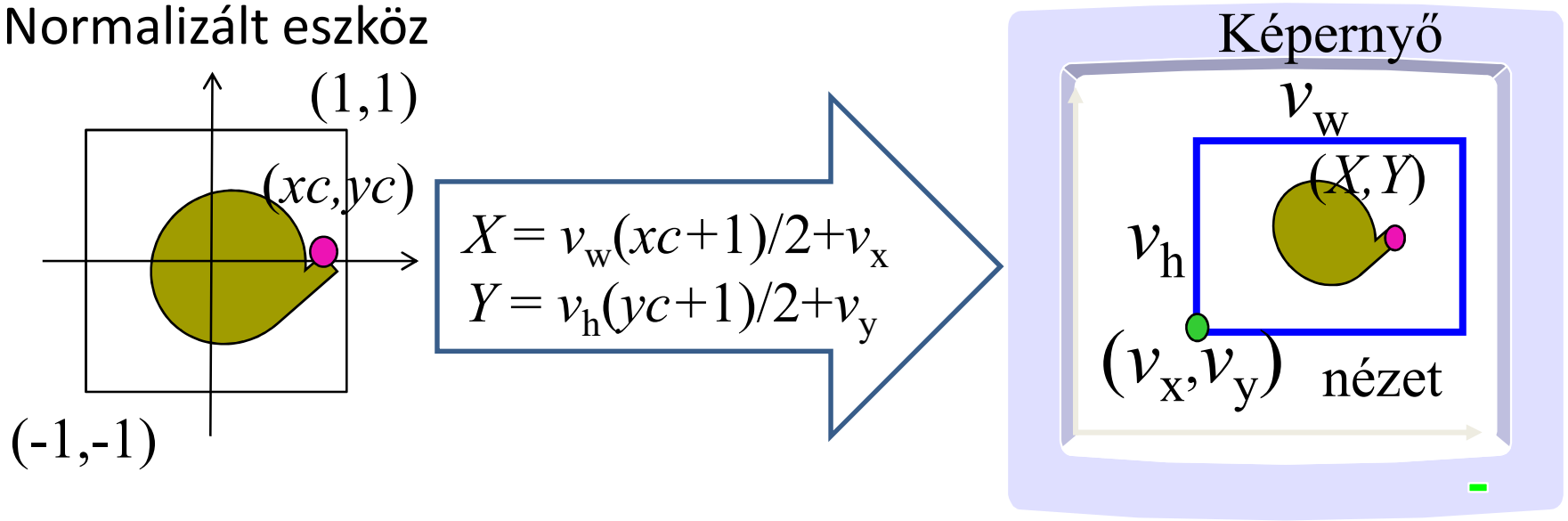
$$[xv, yv, 1] = [x, y, 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -c_x & -c_y & 1 \end{bmatrix}$$

Projekció:
kameraablak a (-1, -1)-(1, 1) négyzetbe



$$[xc, yc, 1] = [xv, yv, 1] \begin{bmatrix} 2/w_x & 0 & 0 \\ 0 & 2/w_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Viewport transzformáció: Normalizáltból képernyő koordinátákba

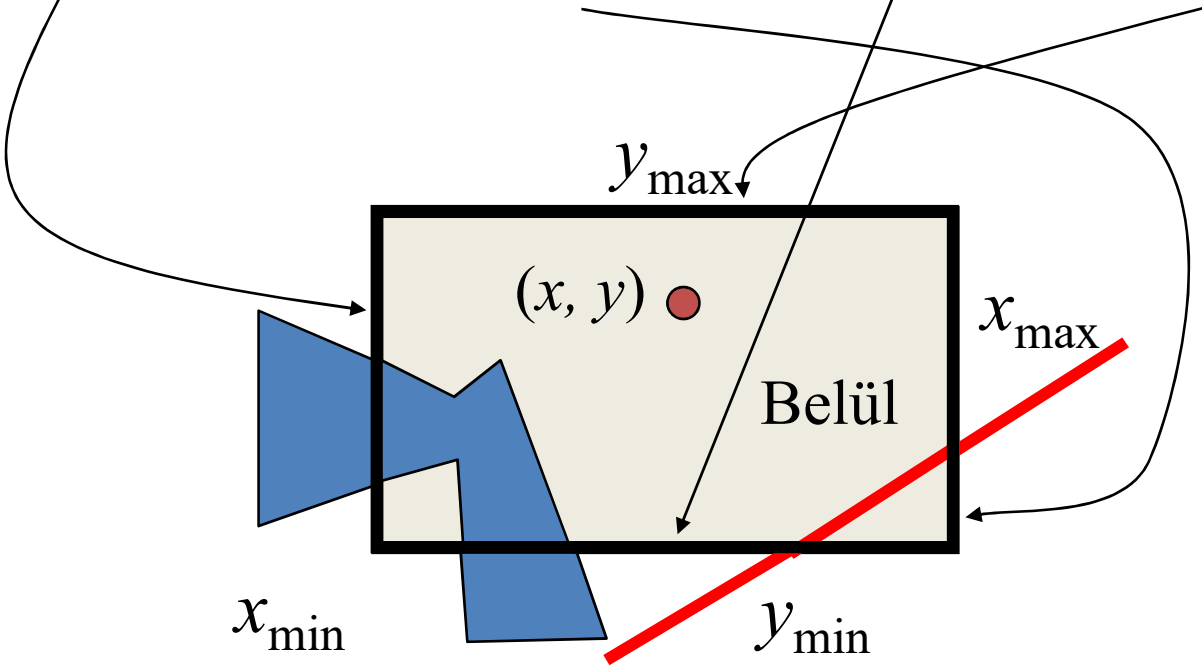



```
glViewport(vx, vy, vw, vh);
```

Vágás

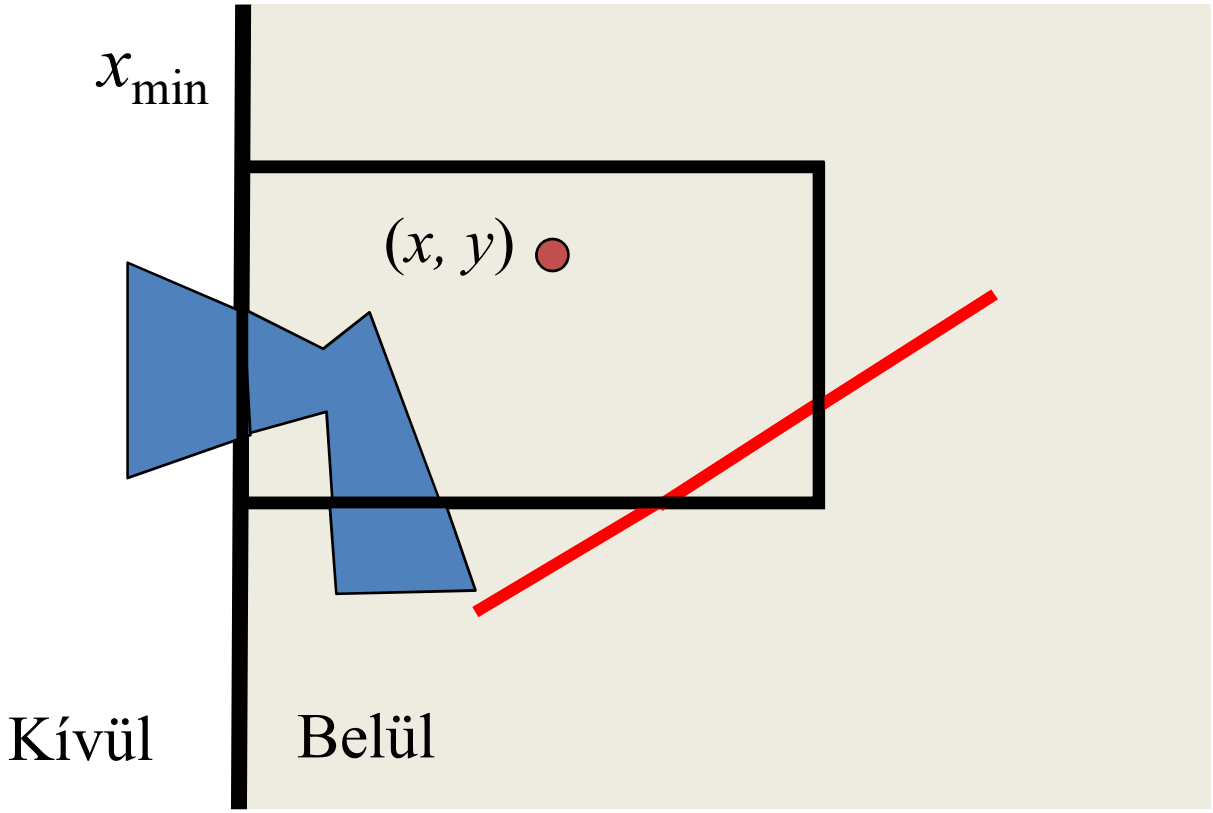
Pont vágás:

$x > x_{min} = -1 \quad x < x_{max} = +1 \quad y > y_{min} = -1 \quad y < y_{max} = +1$

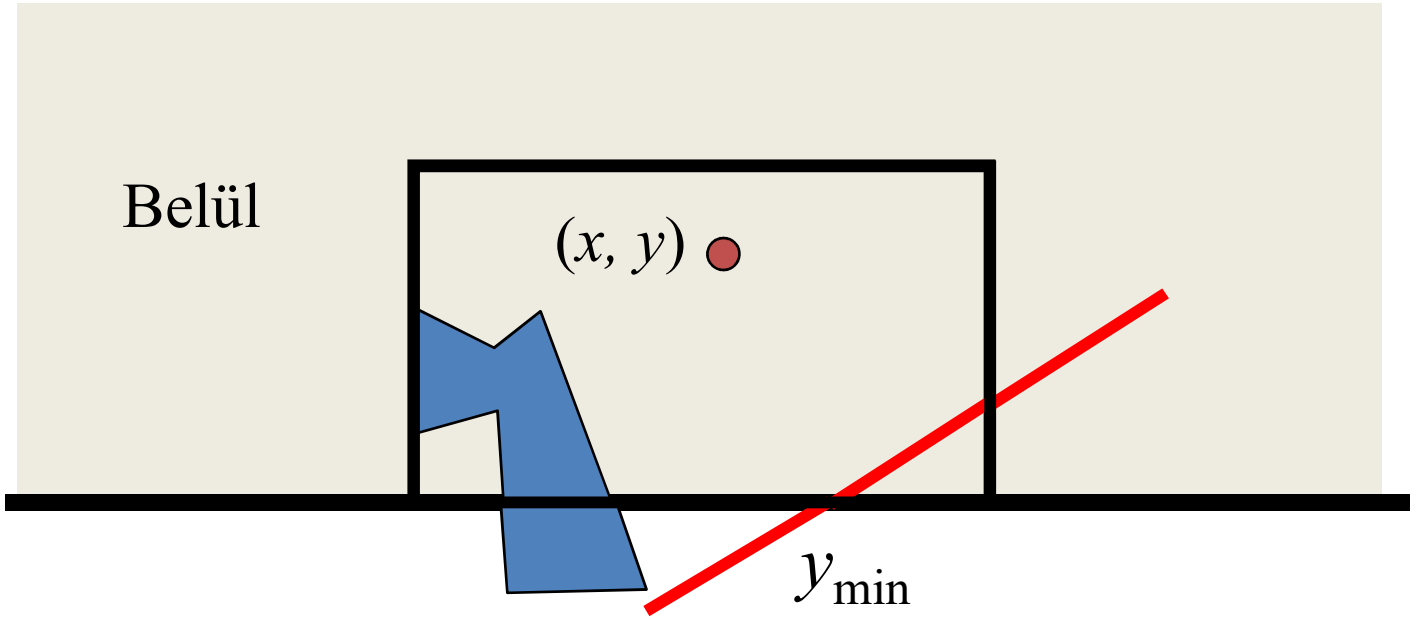


Kívül

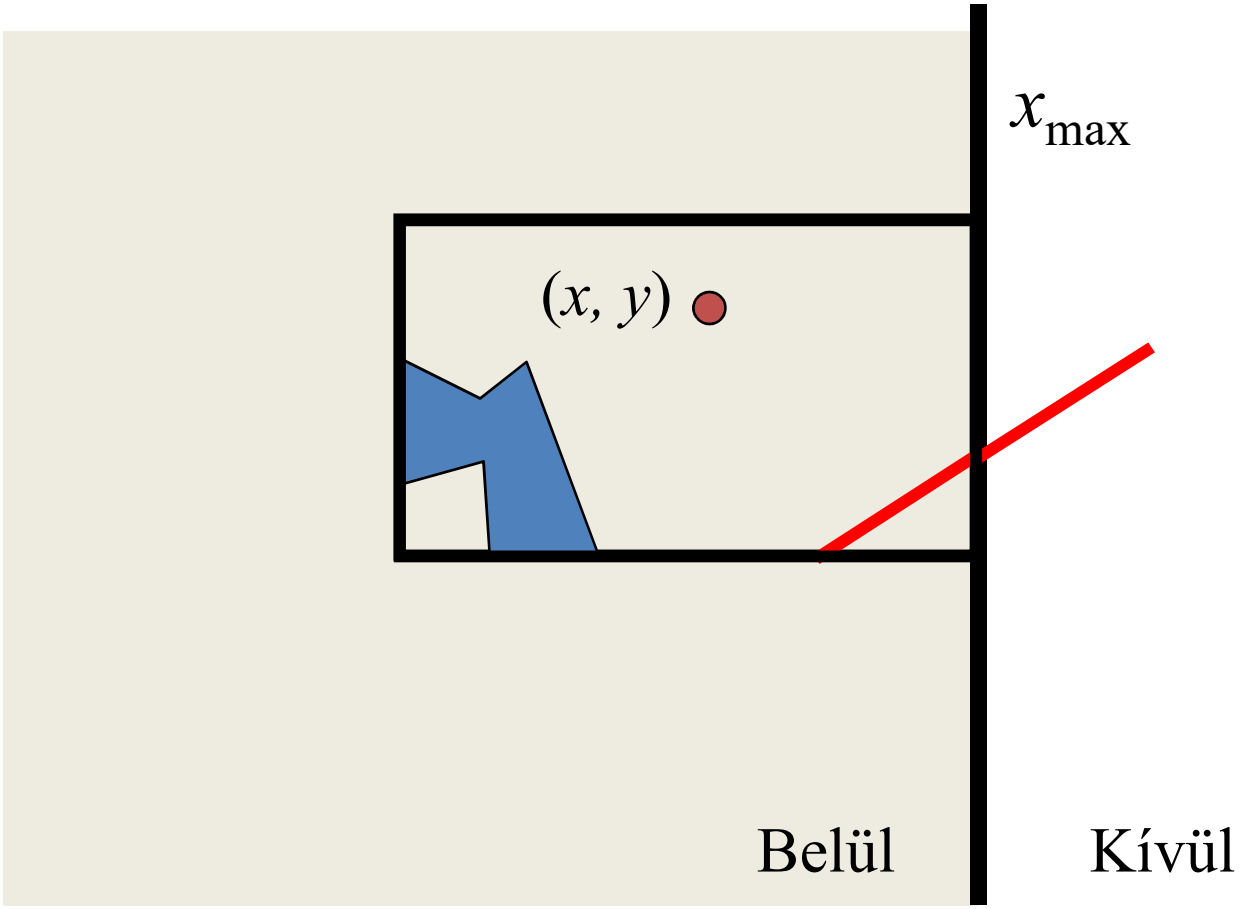
Vágás



Vágás

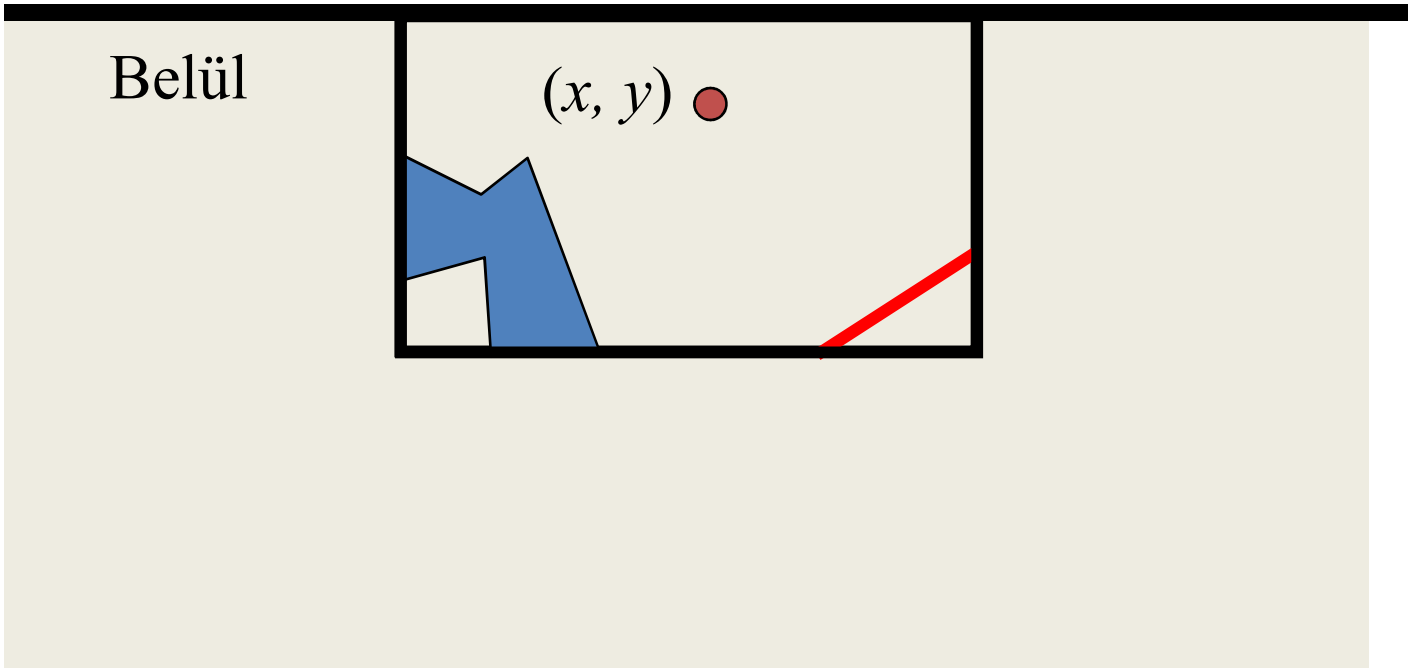


Vágás

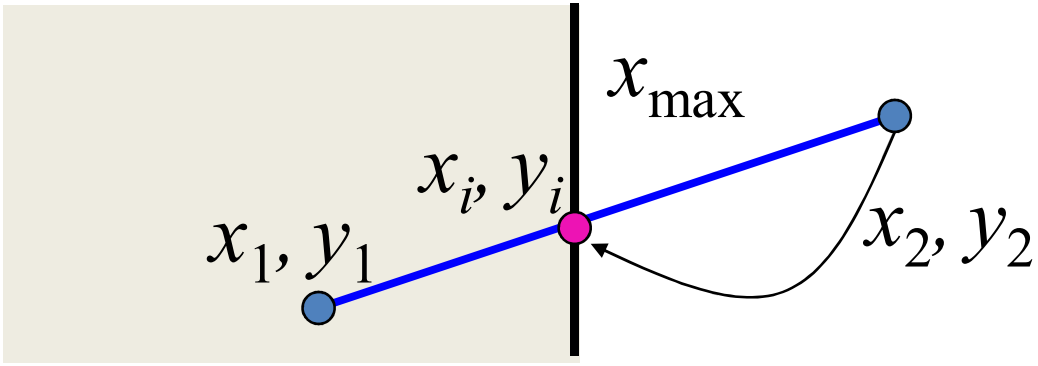


Vágás

Kívül



Szakasz vágás



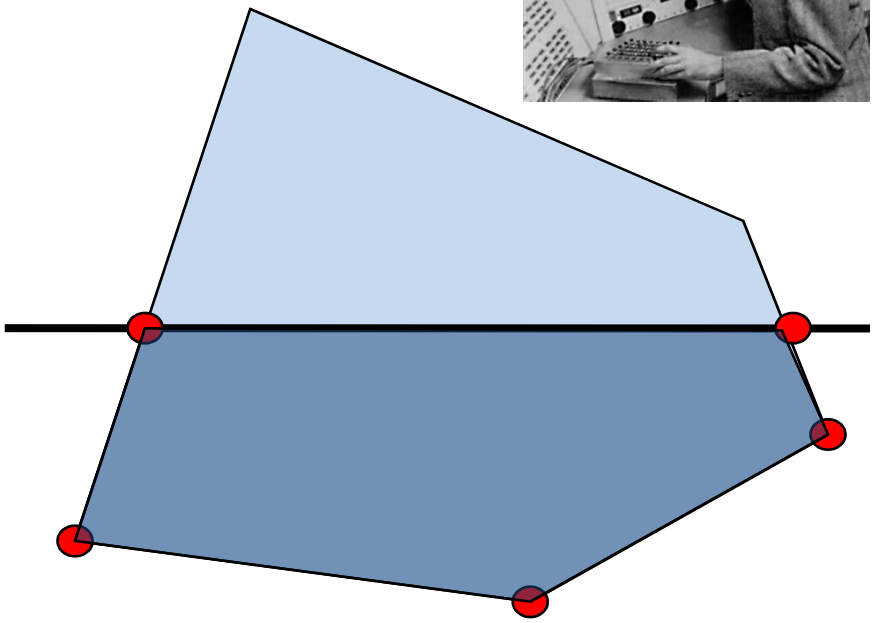
$$x(t) = x_1 + (x_2 - x_1)t, \quad y(t) = y_1 + (y_2 - y_1)t$$

$$x = x_{max}$$

Metszés: $x_{max} = x_1 + (x_2 - x_1)t \Rightarrow t = (x_{max}-x_1)/(x_2-x_1)$

$$x_i = x_{max} \qquad y_i = y_1 + (y_2 - y_1)\,(x_{max}-x_1)/(x_2-x_1)$$

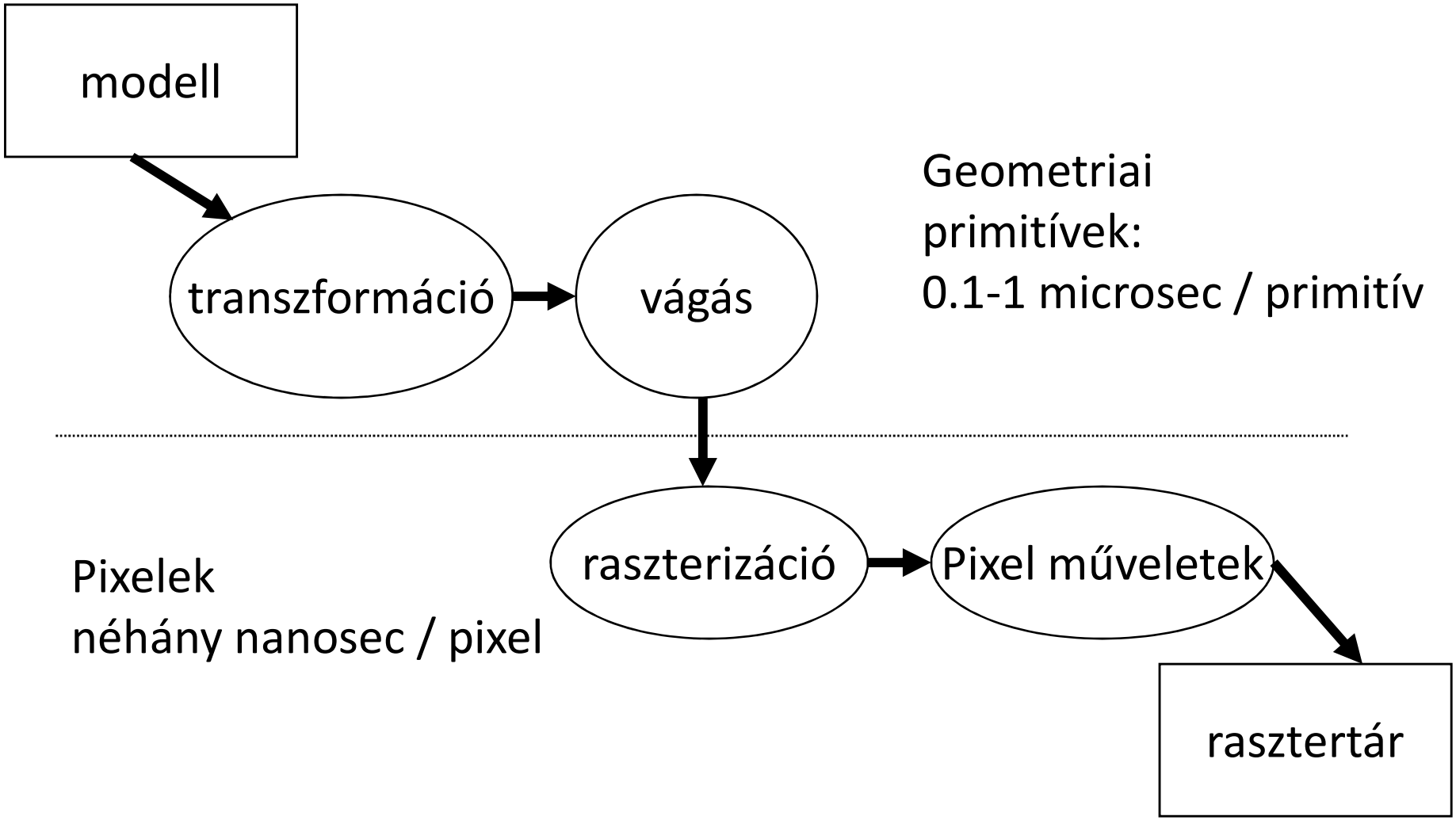
Sutherland-Hodgeman poligonvágás



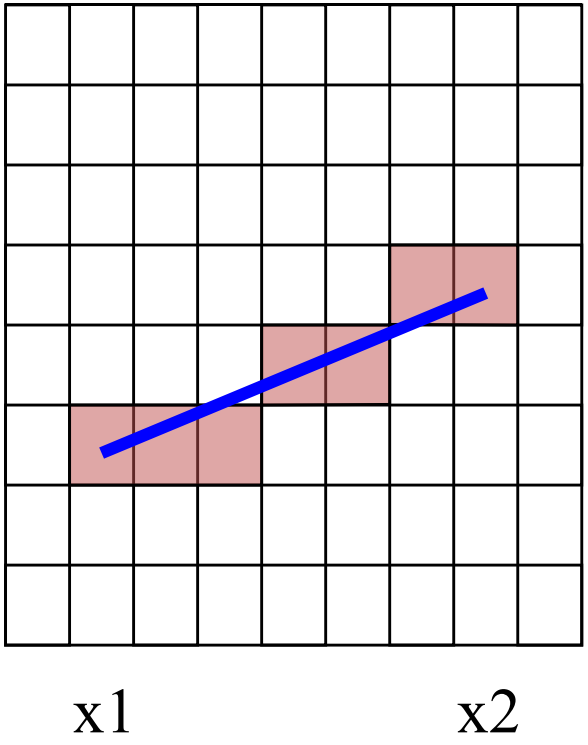
```
PolygonClip(p[n] ⇨ q[m])
    m = 0;
    for( i=0; i < n; i++) {
        if (p[i] belső) {
            q[m++] = p[i];
            if (p[i+1] külső)
                q[m++] = Intersect(p[i], p[i+1], vágóegyenes);
        } else {
            if  (p[i+1] belső)
                q[m++] = Intersect(p[i], p[i+1], vágóegyenes);
        }
    }
}
```

Első pontot még egyszer
a tömb végére

Raszterizáció



Szakasz rajzolás



Egyenes egyenlete:

$$y = mx + b$$

Egyeneshúzás

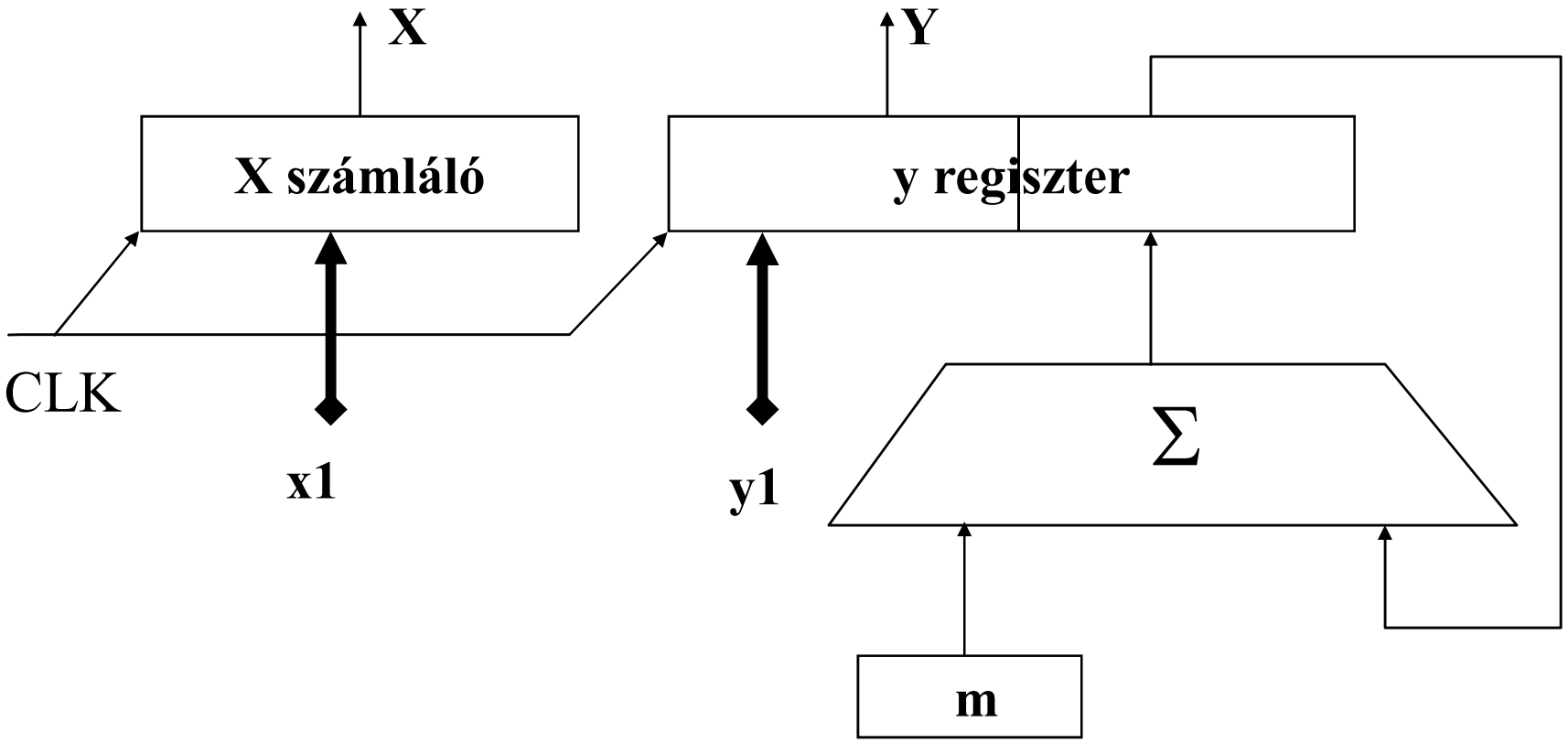
```
for( x = x1; x <= x2; x++) {
        Y = m*x + b;
        y = Round( Y );
        write( x, y );
}
```

Inkrementális elv

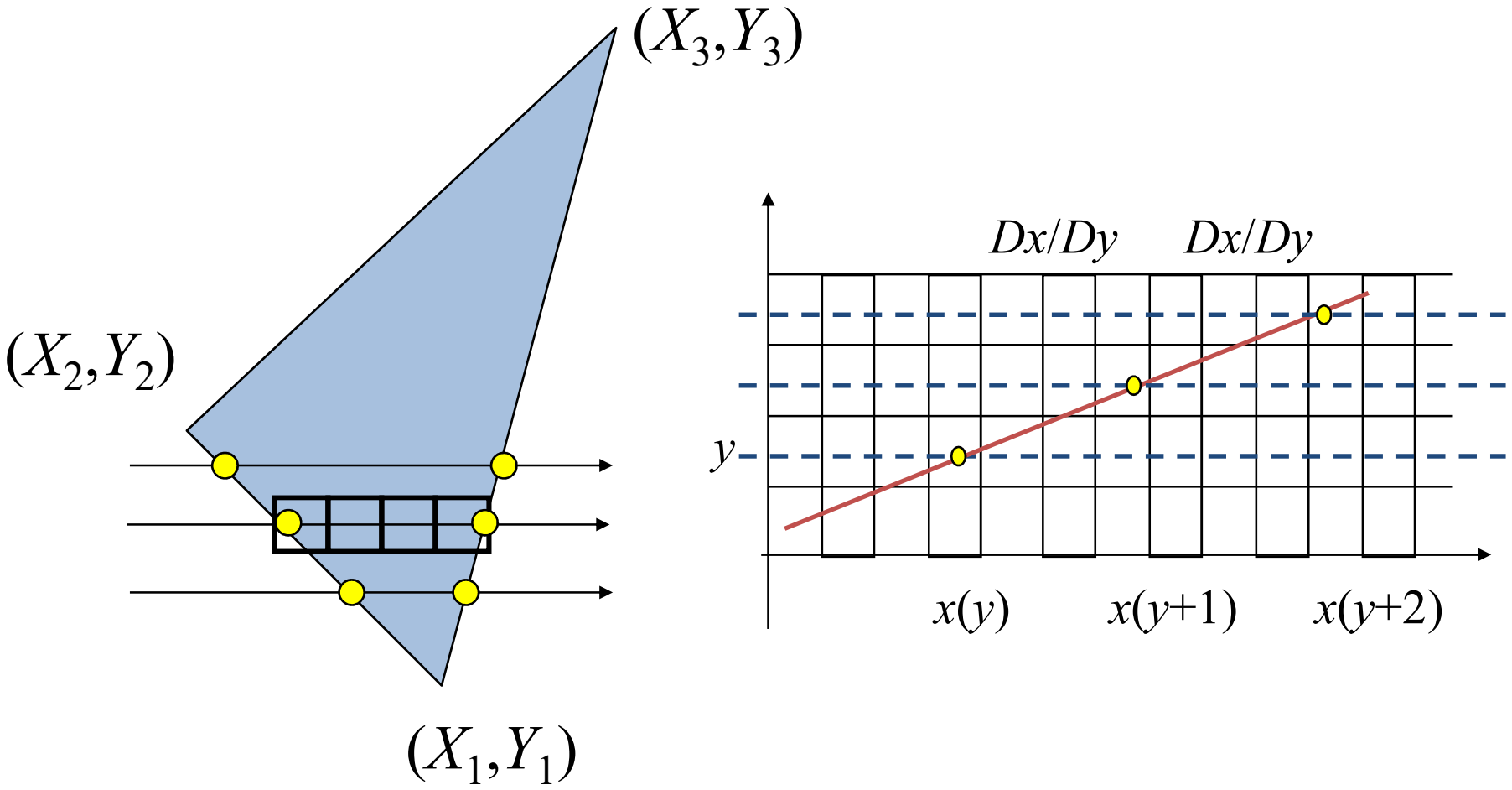
- Egyenlet: $Y(X) = mX + b = Y(X\text{-}1) + m$

```
DDADrawLine(int x1, int y1, int x2, int y2) {
        float m = (y2 - y1)/(x2 - x1);
        float y = y1;
        for(int x = x1; x <= x2; x++) {
                int Y = round(y);
                WRITE(x, Y, color);
                y = y+m;
        }
}
```

DDA szakaszrajzoló hardver



Háromszög kitöltés



Ellenőrző kérdések

- Bizonyítsa be, hogy bármely 4+ csúcsu sokszögnek van diagonálja!
- Bizonyítsa be a kétfül tételt!
- Van értelme a kört raszterizáló algoritmusnak (van ilyen)?
- Írjon sokszögkitöltő algoritmust, amely nem egyszerű (határ önmagát metszi és több határ is van) sokszögeket is ki tud tölteni.
- Implementálja a vágás és raszterizálás algoritmusait!
- Írjon programot, amely eldönti, hogy egy koordináta-tengelyekkel párhuzamos téglalap tartalmaz-e egy szakaszból vagy egy sokszögből valamennyit?
- Adja meg egy 2D szerkesztő (pl. egyszerűsített Powerpoint) osztálydiagramját.
- Mi az értelme a normalizált eszköz-koordinátarendszer bevezetésének?