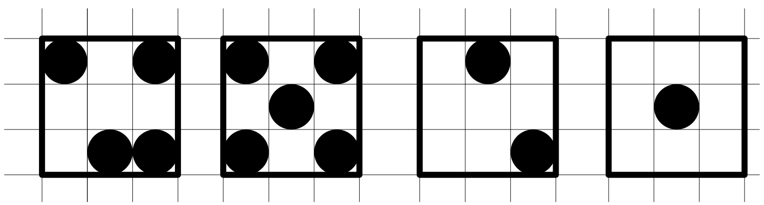
Exercise Computer graphics – (till October 30, 2007)

Ordered dithering

Exercise 12: Ordered dithering meant that a gray value is approximated by different patterns like these:



For imaging devices being able to display a small number of gray levels, an extension was proposed that does not only consist of black and white pixels like shown above but that consists of shades of gray.

0	0	1
0	0	0
0	1	0

1	1	1
1	1	1
1	1	1

1	1	2
1	2	1
1	2	1

2	3	2
2	2	2
2	2	3

Explain how to use these gray-level patterns if the resolution of an image must not be increased.

Solution:

Like in the dithering example which only took black and white into account, our new dithering function expects the (x, y) coordinates and a gray value.

The gray value is used as an index that addresses one of the patterns shows in the lower left row. Gray value 20 would e. g., address the rightmost grid because its sum equals to 20.

The (x, y) coordinate addresses a specific element in the grid, e.g., (x % 3, y % 3) for 3x3 grids. The value in the grid cell will be returned and displayed. For gray value = 20 and (7, 6) the grid element (7 % 3, 6 % 3) = (1, 0) is addressed and its contained value 3 is returned.

Exercise Computer graphics – (till October 30, 2007)

Ordered dithering

Exercise 13: Color dithering

(a) In the lecture we have seen the Floyd-Steinberg dithering algorithm for gray-scale images. How does the approach have to be extended in order to handle color images?

Solution:

Each color component is quantized separately. In the sample implementation, each component is reduced to two values only (which results in a total amount of only 8 colors). Then, the error between the desired color and the one actually set is calculated for each component. For each component, the error is distributed over neighboring pixels as was done in the gray-scale example.

In short: The Floyd-Steinberg implementation is applied on each color channel separately.

(b) Change our example program such that color images are quantized and dithered.

Solution: (See source code)

(c) If you did not yet finish (b) do so now. It can be done making only a few changes in the code.

Exercise Computer graphics

Line clipping according to Cyrus Beck

Exercise 14: Line clipping

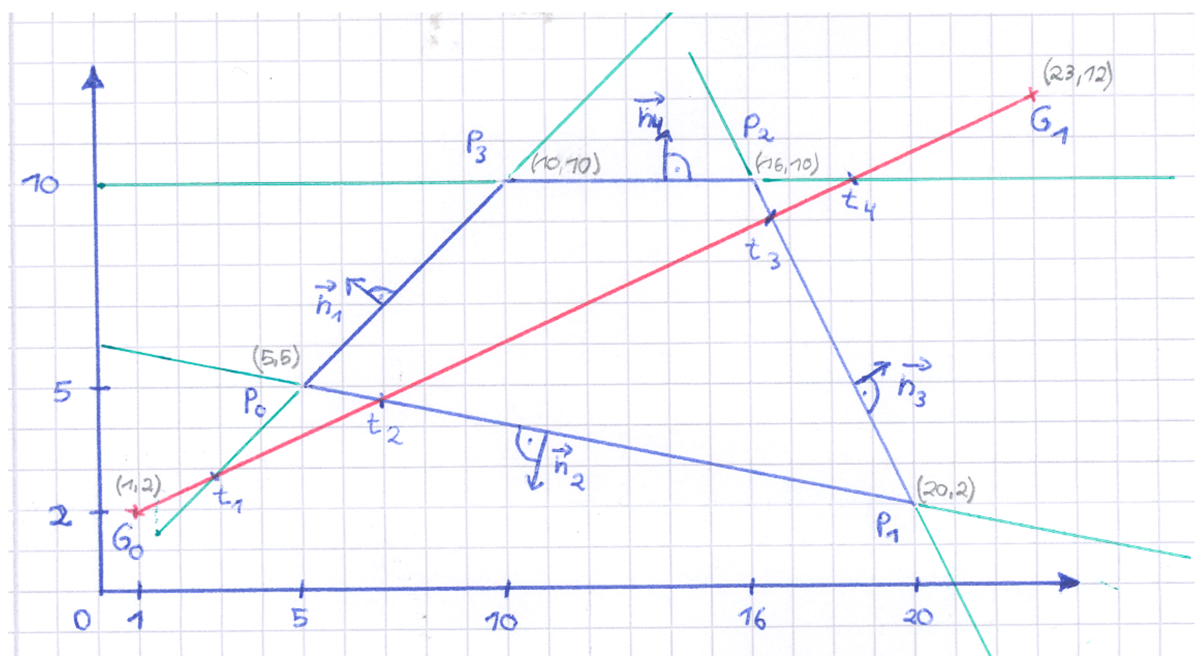
Let a clipping polygon be defined by the vertices (5, 5), (20, 2), (16, 10), (10, 10)

and a line between (1,2) and (23, 12)

(a) Perform the Cyrus Beck clipping algorithm. Find out for each intersection parameter t whether it is "entering" or "leaving" and finally determine which parameters for t are of interest only.

(b)

In the general case of an n-sided polygon: How many intersections have to be performed at most for every line to be displayed?



Exercise Computer graphics

Line clipping according to Cyrus Beck

Exercise 14: Line clipping

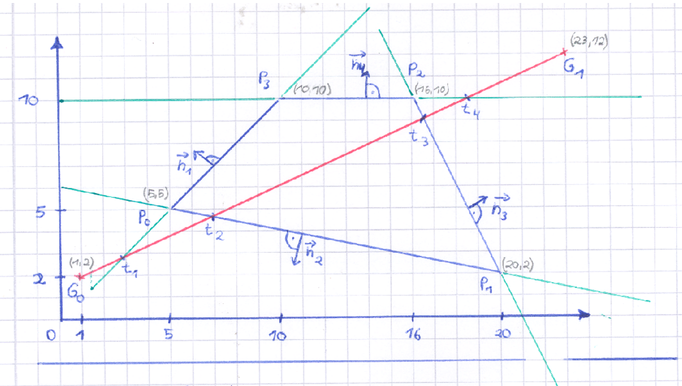
Let a clipping polygon be defined by the vertices
(5, 5), (20, 2), (16, 10), (10, 10)

and a line between
(1,2) and (23, 12)

(a) Perform the Cyrus Beck clipping algorithm. Find out for each intersection parameter t whether it is "entering" or "leaving" and finally determine which parameters for t are of interest only.

Solution (1):

LINE CLIPPING according to CYRUS BECK



Calculate normal vectors:

$\vec{n}_1 = \begin{pmatrix}-1\\1\end{pmatrix}$; $\vec{n}_4 = \begin{pmatrix}0\\1\end{pmatrix}$

$\vec{n}_2 (P_1 - P_0) \stackrel{!}{=} 0 \Rightarrow \begin{pmatrix}n_{2x}\\n_{2y}\end{pmatrix}\begin{pmatrix}20-5\\2-5\end{pmatrix} = 0 \Rightarrow 15 n_{2x} - 3 n_{2y} = 0$

e.g.: $15(-1) - 3(-5) = 0$

$\Rightarrow n_2 = \begin{pmatrix}-1\\-5\end{pmatrix}$

Hint: The normal vectors do not require $|n| = 1$

$\vec{n}_3 (P_2 - P_1) \stackrel{!}{=} 0 \Rightarrow \begin{pmatrix}n_{3x}\\n_{3y}\end{pmatrix}\begin{pmatrix}16-20\\10-2\end{pmatrix} = 0 \Rightarrow -4 n_{3x} + 8 n_{3y} = 0$

e.g.: $-4 \cdot (+2) + 8 \cdot 1 = 0$

$\Rightarrow n_3 = \begin{pmatrix}2\\1\end{pmatrix}$

Intersection: $[G_0 + t(G_1 - G_0) - P_0]\,\vec{n}_1 \stackrel{!}{=} 0$

$$\Rightarrow t = \frac{P_0 \vec{n}_1 - G_0 \vec{n}_1}{\vec{n}_1 (G_1 - G_0)} = \frac{\vec{n}_1 (P_0 - G_0)}{\vec{n}_1 (G_1 - G_0)}$$

$$t_1 = \frac{\begin{pmatrix}-1\\1\end{pmatrix}\left[\begin{pmatrix}5\\5\end{pmatrix} - \begin{pmatrix}1\\2\end{pmatrix}\right]}{\begin{pmatrix}-1\\1\end{pmatrix}\left[\begin{pmatrix}23\\12\end{pmatrix} - \begin{pmatrix}1\\2\end{pmatrix}\right]} = \frac{-4+3}{-22+10} = \frac{-1}{-12}$$

← denominator < 0 ⇒ „entering"

$$t_2 = \frac{\begin{pmatrix}-1\\-5\end{pmatrix}\left[\begin{pmatrix}5\\5\end{pmatrix} - \begin{pmatrix}1\\2\end{pmatrix}\right]}{\begin{pmatrix}-1\\-5\end{pmatrix}\begin{pmatrix}22\\10\end{pmatrix}} = \frac{-4-15}{-22-50} = \frac{-19}{-72}$$

← „entering"

Exercise Computer graphics

Line clipping according to Cyrus Beck

Exercise 14: Line clipping

Let a clipping polygon be defined by the vertices
(5, 5), (20, 2), (16, 10), (10, 10)

and a line between
(1,2) and (23, 12)

(a) Perform the Cyrus Beck clipping algorithm. Find out for each intersection parameter t whether it is "entering" or "leaving" and finally determine which parameters for t are of interest only.

Solution (2):

$$t_3 = \frac{\begin{pmatrix}2\\1\end{pmatrix}\left[\begin{pmatrix}20\\2\end{pmatrix} - \begin{pmatrix}1\\2\end{pmatrix}\right]}{\begin{pmatrix}2\\1\end{pmatrix}\begin{pmatrix}22\\10\end{pmatrix}} = \frac{38}{44+10} = \frac{19}{27}$$ ← denominator > 0 ⇒ „leave"

$$t_4 = \frac{\begin{pmatrix}0\\1\end{pmatrix}\left[\begin{pmatrix}16\\10\end{pmatrix} - \begin{pmatrix}1\\2\end{pmatrix}\right]}{\begin{pmatrix}0\\1\end{pmatrix}\begin{pmatrix}22\\10\end{pmatrix}} = \frac{8}{10}$$ ← denominator > 0 ⇒ „leave"

We are interested in the largest entering value and the smallest leaving value

$$\Rightarrow t_E = \frac{19}{72}; \quad t_L = \frac{19}{27}$$

$$\Rightarrow P_E = \begin{pmatrix}1\\2\end{pmatrix} + \frac{19}{72}\begin{pmatrix}22\\10\end{pmatrix} \approx \begin{pmatrix}6,8\\4,6\end{pmatrix}$$

$$P_L = \begin{pmatrix}1\\2\end{pmatrix} + \frac{19}{27}\begin{pmatrix}22\\10\end{pmatrix} \approx \begin{pmatrix}16,5\\9\end{pmatrix}$$

Exercise Computer graphics

Line clipping according to Cyrus Beck

Exercise 14: Line clipping

Let a clipping polygon be defined by the vertices
(5, 5), (20, 2), (16, 10), (10, 10)

and a line between
(1,2) and (23, 12)

(b)

In the general case of an n-sided polygon: How many intersections have to be performed at most for every line to be displayed?

Solution:

Every line of the polygon can potentially intersect a line to be drawn, unless they are parallel.