Exercise Computer graphics – (till 28. September 2006)

Ultra-fast line drawing

Exercise 4: In the lecture we got to know the mid-point line-drawing algorithm which can generate line pixels with a single comparison and two or three integer additions per pixel, only.

Alter the algorithm such that points are plotted more sparsely. Therefore, the algorithm's loop should advance by two pixels in the X-direction rather than one as done in the lecture. The sketch on the right shows the cases which can evolve.

Hints:

Start with the initialization of the distance-value at $(x0+2,y0+0,5)$ and $(x0+2,y0+1,5)$ and find out the increments in the inner loop, exactly as done in the normal mid-point algorithm. Note that now you have to take more than one distance into account.
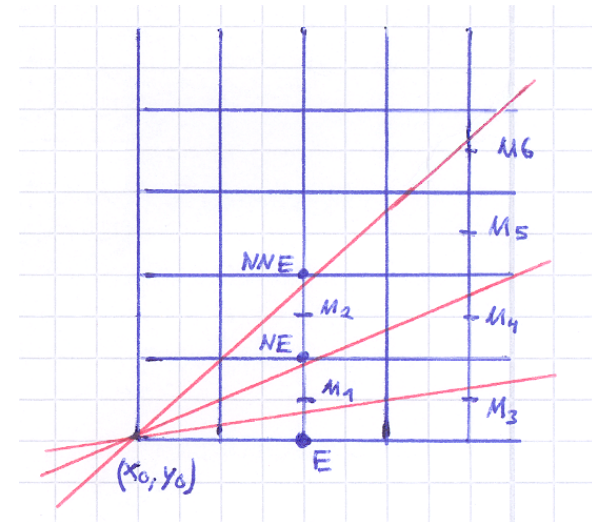
Exercise 5: The mid-point algorithm uses integer additions only. Usually, 16 bits are sufficient for representing either x- or y-components. However, modern processors are operating at 32 or even 64 bits at a time. Assuming that an integer has at least 32 bits, we exploit the lower 16 bits only because resolutions of more than 65535 x 65535 pixels are unlikely.

Change the midpoint implementation such that the most significant 16 bits are used as well. Which improvements can you achieve?