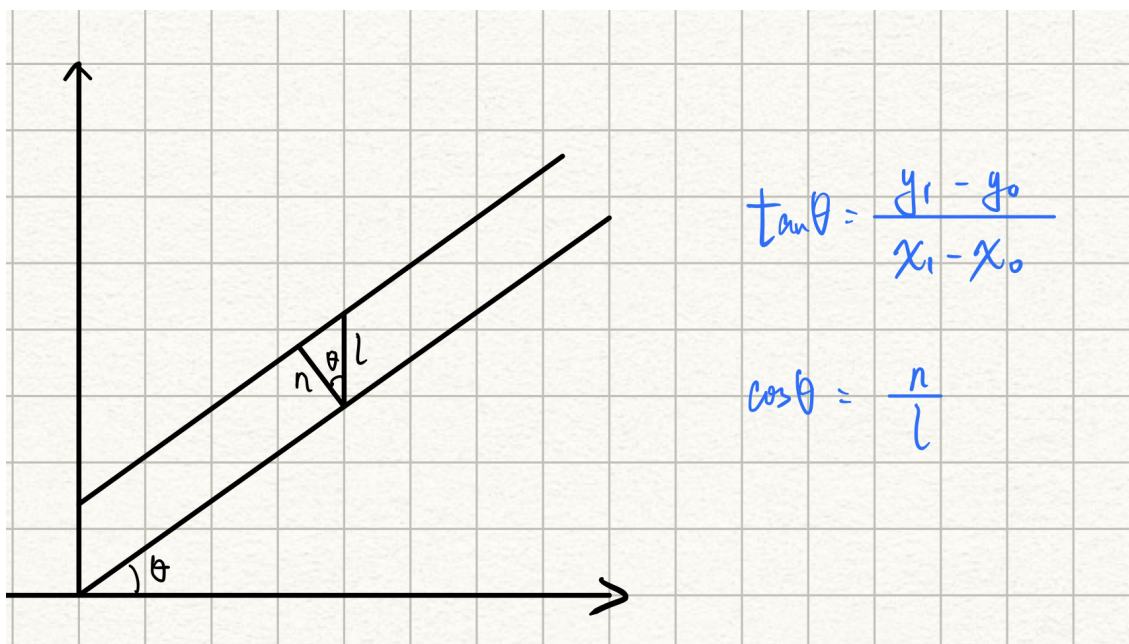


CG课堂思考题 (3.8)

1850250 赵浠明

1. 考虑直线线宽的条件下，如何使用中点画线法描绘像素点？



解法思路

假设直线（图中下方的直线）的起点、终点分别为 (x_0, y_0) ， (x_1, y_1) ，直线的线宽为 n 。

如上图所示，我认为考虑线宽条件下进行直线像素点描绘，相当于对两条无线宽的直线进行像素点描绘，并同时点亮两条直线之间的所有像素点即可。

具体解法为：

1. 使用传统的中点画线法对下方直线进行像素点描绘。
2. 由于两条直线平行，即 $l_1: ax + by + c = 0$ 和 $l_2: ax + by + c + l = 0$ ，因此中点画线法中的 $d_1 = 2a$ 和 $d_2 = 2(a + b)$ 可以重复使用，不同的只是上方直线的起点变为 $(x'_0, y'_0) = (x_0, y_0 + l)$ ，并对 y'_0 进行四舍五入取整，再对上方直线进行像素点描绘。
3. 对于所有 x_i ，设上下直线点亮的像素点分别为 (x_i, y_i) 和 (x_i, y'_i) ，将两点之间的所有像素点也全部点亮。

垂直线宽L的计算

我的思路中上述第二步中的 l 并不太好计算，而且还要作一次四舍五入，如图中的两个公式可知，根据已知条件不能通过简单计算得出 l 的值，需要使用到角度运算，这一步会比较耗费时间，也是我这个思路的缺点。

代码实现

```
void Midpoint Line (int x0,int y0,int x1, int y1,int color)
{
    int a, b, d1, d2, d;
    int x, y, y_up, l;
    a = y0 - y1;
    b = x1 - x0;
    d = 2 * a + b;
    d1 = 2 * a;
    d2 = 2 * (a + b);

    /* calculate the value of l */

    x = x0;
    y = y0;
    y_up = round(y0 + l);

    for (j: y to y_up) {
        drawpixel(x, j, color);
    }
    while (x < x1)
    {
        if (d<0) {
            x++;
            y++;
            y_up++;
            d += d2;
        }
        else {
            x++;
            d += d1;
        }

        for (j: y to y_up) {
            drawpixel(x, j, color);
        }
    }
} /* mid PointLine */
```

