

```

1 typedef double Ct;
2 struct Pt
3 {
4     Pt(Ct _x = 0, Ct _y = 0) : x(_x), y(_y){}
5     Ct x, y;
6 };
7 struct Line
8 {
9     Line(Ct _a = 0, Ct _b = 0, Ct _c = 0) : a(_a), b(_b), c(_c){}
10    Ct online(const Pt& p) const {return a * p.x + b * p.y + c;}
11    Ct a, b, c;
12 };
13 const double pi = acos(-1);
14 typedef Pt Vec;
15 namespace S2C {
16     inline int dcmp(double x)
17 }

```

```
19     if (fabs(x) < 1e-8) return 0;
20     return x > 0 ? 1 : -1;
21
22 inline Pt operator - (const Pt& a, const Pt& b)
23 {
24     return Pt(a.x-b.x, a.y-b.y);
25 }
26 inline Pt operator + (const Pt& a, const Pt& b)
27 {
28     return Pt(a.x+b.x, a.y+b.y);
29 }
30 inline Ct dot(const Pt& a, const Pt& b)
31 {
32     return a.x*b.x + a.y*b.y;
33 }
34 inline Ct dot(const Pt& a, const Pt& b, const Pt& c)
35 {
36     return dot(b-a, c-a);
37 }
38 inline Ct cross(const Pt& a, const Pt& b)
39 {
40     return a.x * b.y - a.y * b.x;
41 }
42 inline Ct cross(const Pt& a, const Pt& b, const Pt& c)
43 {
44     return cross(b-a, c-a);
45 }
46 inline double norm(const Pt& a)
47 {
48     return sqrt(dot(a, a));
49 }
50 inline Ct norm2(const Pt& a)
51 {
52     return dot(a, a);
53 }
54 inline double p2l(const Pt& a, const Line& l)
55 {
56     return fabs(1.online(a))/norm(Pt(l.a, l.b));
57 }
58 inline Line make_line(const Pt& a, const Pt& b)
59 {
60     Line ret;
61     ret.a = b.y-a.y;
62     ret.b = -(b.x-a.x);
63     ret.c = cross(b-a, a);
64     return ret;
65 }
66 inline Pt get_symmetry(const Pt& a, const Line& l)
67 {
68     Pt x(l.a, l.b), y(dot(x, a)+2*l.c, cross(x, a));
69     double t = norm2(x);
70     return Pt(-dot(y, x)/t, -cross(y, x)/t);
71 }
72 inline Vec rotate(const Vec& v, double sita)
73 {
74     Pt r(cos(sita), -sin(sita));
75     return Pt(dot(v, r), -cross(v, r));
76 }
77 ostream& operator << (ostream& o, const Pt& p)
78 {
79     return o << "(" << p.x << ", " << p.y << ")";
80 }
81
82 Line solve(Line l, Pt p, Ct r)
83 {
84     if (dbcnp(1.online(p)) == 0 || dbcnp(r) == 0) return l; //点在直线上或者平移距离为零
85
86     double delta = sqrt(1.a * 1.a + 1.b * 1.b);
87     double dist = r * delta;
88     Line ans1(1.a, 1.b, 1.c + dist), ans2(1.a, 1.b, 1.c - dist); //求出两点可能的直线
89
90     int x1 = dbcnp(ans1.online(p)), x2 = dbcnp(ans2.online(p)); //p就在其中一条直线上
91     if (x1 == 0) return ans1;
92     if (x2 == 0) return ans2;
93
94     //求目标直线的距离近
95     double dist1 = p2l(p, ans1);
96     return dist1 > r ? ans2 : ans1;
```

回复

qinhen547 2010-08-06

[Quote=引用 10 楼 Kktemp234 的回复]

疏忽了，点的条件不多余，可以判断是加还是减，即在图像平移时该往左移还是往右移，如果这个点恰好在直线上，则左右均可
[/Quote]
楼主说了这点不在直线上...
而且方向只要直线和点是已给的不是未知数就可以知道点是在直线的上方还是下方，移动方向就只有一个...
但要是都是未知数，讨论情况就多了.....

回复

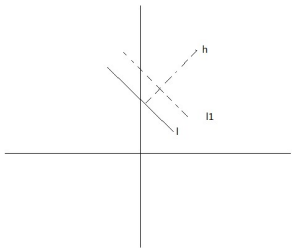
qinhen547 2010-08-06

(1) 点 (x_0, y_0) 到直线 $Ax+By+C=0$ 的距离

$$d = \frac{|Ax_0 + By_0 + C|}{\sqrt{A^2 + B^2}}$$

注意：运用本公式前把直线方程写为一般式。

(2) 两条平行线 $\begin{cases} l_1: Ax + By + C_1 = 0 \\ l_2: Ax + By + C_2 = 0 \end{cases}$ 之间的距离 $d = \frac{|C_1 - C_2|}{\sqrt{A^2 + B^2}}$



回复

ahao 2010-08-06

[Quote=引用 9 楼 Kktemp234 的回复]

另外，这个和C或C++没多大关系啊，楼主干么发这儿，应发在算法的地儿
[/Quote]

因为这个也算不上很复杂的算法，只是以前学过的东东，现在都忘了，
我知道c++版学生多，所以来这里问问:-)

上面几位的答案，我先想想

回复

最帅马老师 2010-08-06

疏忽了，点的条件不多余，可以判断是加还是减，即在图像平移时该往左移还是往右移，如果这个点恰好在直线上，则左右均可

回复

最帅马老师 2010-08-06

另外，这个和C或C++没多大关系啊，楼主干么发这儿，应发在算法的地儿

回复

最帅马老师 2010-08-06

我认为楼上的理解有问题，下面是我的理解。

已知一个直线的普通式方程 $ax+by+c=0$ 和一个点 (m,n)
并没有说这个点在直线上。

对于将直线平移的问题，都是求这个直线的平行线，设这个直线为 $ax+by+k=0$ k 为不等于 c 的任意实数。
直线斜率为 $-a/b$ ，与 x 轴夹角为 A ，则直线向此点平移等效于沿点做此直线的垂线，在垂线上求出距离为 r 的点。解三角形，由正弦定理可得直线在 x 方向平移的距离 $r \sin A$ ，设这个值为 k

对于一个面数，当图像在 x 轴作平移时，平移量为 k ，则方程为 $(x+k)$
因此新的直线方程为 $a(x-r \sin A)+by+c=0$

点的条件多余，因为总有一条经过点 (m,n) 且与原直线平行的直线存在

个人理解

qinzen547 2010-08-06
[Quote=引用 6 楼 ahao 的回复]

我想的办法是，根据直线的角度，用sin和cos计算r在两轴上的位移，对不对？
稍微得慢，解析几何已经全部还给老师了。
[/Quote]
这个方法也可以，不过要先求出已知直线与垂线的交点，然后利用你所说的方法求出要求的直线与垂线的交点，这样要求的直线已经一个点和斜率就可以求出来了...

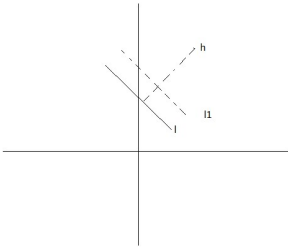
赞

ahao 2010-08-06

我想的办法是，根据直线的角度，用sin和cos计算r在两轴上的位移，对不对？
稍微得慢，解析几何已经全部还给老师了。

赞

qinzen547 2010-08-06



先利用已知直线和已知点求出过已知点直线的垂线，平移r就是按垂线的方向移动，然后再利用和已经直接的关系求出平移后的直线...可以利用点斜式求出...
坐标轴的情况就更简单了，不说了...

赞

加载更多回复

相关推荐

空间交会对接最终平移段直线制导律研究

针对空间交会对接最终平移段跟踪器制导律作采滤波误差运动问题，基于相对运动动力学特性研究V-bar、R-bar直线接近制导方法，设计了采用脉冲式匀速和有限推力式变速度直线制导律，使得了燃料最省的制导条件，最后进行了...

圆平移后的方程变化_1.1 直角坐标系中的平移变换与伸缩变换

1.1直角坐标系中的平移变换与伸缩变换目标: 平移变换与伸缩变换的应用与理解一. 直角坐标系1. 直线上，取定一个点为原点，规定一个长度为半单位长度，规定直线的方向为正方向。这样我们就建立了直线上的坐标系(图...)

圆平移后的方程变化_将圆x^2+y^2=1沿x轴正向平移1个单位后所得圆C，则圆C的方程是()...

若点M(x0,y0)在圆x^2+y^2=Dx+Ey+F=0上，则过点M的切线方程为x0 x + y0 y + D'(x-x0)/2 + E'(y-y0)/2 + F =0或表述为: 若点M(x0,y0)在圆(x-a)^2+(y-b)^2=r^2上，则过点M的切线方程为(x-a)(x0-a)+(y-b)(y-b)=r^2...

圆平移后的方程变化_直角坐标系中的平移变换与伸缩变换

※图下注1.1 直角坐标系中的平移变换与伸缩变换目标: 平移变换与伸缩变换的应用与理解一. 直角坐标系1. 直线上，取定一个点为原点，规定一个长度为半单位长度，规定直线的方向为正方向。这样我们就建立了直线上的坐标...

matlab求空间直线公垂线_大一的求两直线公垂线方程

大一的求两直线公垂线方程以下文字资料由(写文章知识网www.bahairzhi.com)小编为大家整理整理后发布的内容，让我们赶快一起来看一下吧！大一的求两直线公垂线方程公垂线是无数条的，公垂线所在的直线是唯...一条...

一条直线平行与斜率方向平移距离d后的坐标

##只能输入一条直线==两个点坐标,得到直线方程 y=kx+b def Equation(A, B): # points list 坐标[x,y] # B[0]-A[0] == 0: raise ("数据错误！points", points) res = [0, 0] # 计算直线方程K:=y1-y0/x1-x0

c语言直线和圆的方程结论_直线和圆的方程知识点总结

1. 直线与圆的方程方程一. 直线方程1. 直线的斜截角2. 直线方程的几种形式: 点斜式、截距式、两点式、斜切式3. 两条直线平行: 推论: 如果两条直线的斜截角为两, 两条直线垂直: 两条直线垂直的条件: 设两条直线和的...

两直线平行交叉相乘_两条直线方程相乘的几何意义_是不是说两相交直线的点的轨迹...

底开全部两条直线方程相乘的几何意义: 1. 两条直线方程相乘的几何意义就是两相交直线的点的轨迹84323131335323631343130323136353331333431356639的轨迹; 2. 两条直线方程相乘仅表示满足两直线方程的点的自变量的平方...

已知两点坐标求直线的一般表达式

假设有两个点x1,y1,x2,y2 一般式Ax+By+C=0 A=y2-y1 B=x1-y2 g=___gcd(A,B); A/=g,B/=g; 那么C就 C=-(Ax+By) 随便带一个点进去就好了 转载于https://www.cnblogs.com/mch5201314/p/10844958.html...

线性代数笔记6——直线和曲线的参数方程

线性方程的几何意义二元线性方程 该方程是一个二元线性方程，包含两个方程，每个方程都是一条直线，两条直线的交点就是该方程有唯一解，这就是二元线性方程的几何意义，平面方程 空间内不在同一直线上的三点构成一...

直线的倾斜角斜率和直线方程

前言: 直线方程 典例例1 直线的方向向量 例1 直线3x + 4y + 5 = 0 的方向向量是共线的一个单位向量是[] A.(3, 4) B.(4, -3) C.(4/5, -3/5) D.(4/5, 3/5) D (4/5, -3/5) ...

数学基础一: 直线方程Ax+By+C=0

一、什么是直线及其表示方法: 首先我们来看直线，不同两点可以确定一条直线，然后从直线寻找规律，在平面直角坐标系中任意不同两点设为(x1,y1)和(x2,y2)，再画一些平行X轴和Y轴的辅助线，然后在这条直线再取一个点...

java两条直线交点_求两条直线的交点坐标

晋UdaCity机器学习大纲，发现入门课中有一条: 编写算法计算一值直线或...记录下思考过程版本一: /求直角坐标系中直线2x-y=0 和 4x-5y=9的交点坐标x = 0.5\$olution = array(1)while(true){/ 2*\$x-\$y==0; \$这样...

判断点在三角形区域内: 求空间直线与平面的交点:

A 求空间直线与平面的交点: 若直线不与平面平行，将存在交点，如下图所示，已知直线L过点m(m1, m2, m3)，且方向向量为vL(v1, v2, v3)，平面P过点n(n1, n2, n3)，且法线方向向量为VP(vp1, vp2)...

直线的参数方程ABCF基础中档高阶辅导

可以类比成数值的原点，点A, B, C分别是点P沿垂直方向斜向上平移了1, 2, 3个单位得到的，点D, E分别是点P沿垂直方向斜向下平移了1, 2个单位得到的，点M对应参数t，那么当t = 1, 2, 3 时，分别对应点A...

计算机图形学综合(画图形、旋转、缩放等)实验

计算机图形学的综合实验，但是制作得比较和平，能实现所要求的基本功能，如画直线、椭圆、矩形、多边形，能对图形进行旋转、平移、缩小、放大，能够实现对多变形的跟踪。

8月8日程序备份——找挡板角度及平移误差

//求直线方程 float A = 0, B = 0, C = 0; A = pointA.y - pointB.y; B = pointB.x - pointA.x; C = pointA.x*pointB.y - pointA.y*pointB.x; //代入点到直线距离公式 float d = distance = 0; ...

comsol移动网络_借助变形网格接口模拟平移运动

本篇博客中，我们将介绍何时使用这些接口，以及如何通过它们来高效模拟平移运动，使用变形网格的代价假使我们希望通过 COMSOL Multiphysics 模型来描述一个在较大域内移动的固体对象，域内充满诸如空气等的流体...

氯气报警设定值原理知识.pdf

氯气报警设定值原理知识.pdf

《2021年品牌CDP与营销数字化转型报告》(185页).pdf

《2021年品牌CDP与营销数字化转型报告》(185页).pdf

关于我们 招贤纳士 商务合作 寻求报道 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00

公安备案号11010502030143 京ICP备19004658号 京网文[2020]1039-165号 经营性网站备案信息 北京互联网违法和不良信息举报中心 家长监护 网络110报警服务 中国互联网举报中心 Chrome商店下载 ©1999-2022北京创新信知网络技术有限公司 版权所有 极识号免流声明 极识号广告 出版物许可证 京企头照