定义点的结构

Struct Point（X,Y,Z,Distance）

假定四个已知点的坐标如下：

P1(x1,y1,z1,d1)

P2(x2,y2,z2,d2)

P3(x3,y3,z3,d3)

P4(x4,y4,z4,d4)

P0(x0,y0,z0) \*\*\*P0点是待测点

所以可得4个方程：

(x0-x1)²+(y0-y1)²+(z0-z1)²=d1² ……………..式1

(x0-x2)²+(y0-y2)²+(z0-z2)²=d2² ……………..式2

(x0-x3)²+(y0-y3)²+(z0-z3)²=d3² ……………..式3

(x0-x4)²+(y0-y4)²+(z0-z4)²=d4² ……………..式4

由式1和式2可得：

2\*(x2-x1)\*x0+2\*(y2-y1)\*y0+2\*(z2-z1)\*z0=d1²-d2²+x2²-x1²+y2²-y1²+z2²-z1²; ……………..式5

同理可得：

2\*(x3-x1)\*x0+2\*(y3-y1)\*y0+2\*(z3-z1)\*z0=d1²-d3²+x3²-x1²+y3²-y1²+z3²-z1²; ……………..式6

2\*(x4-x1)\*x0+2\*(y4-y1)\*y0+2\*(z4-z1)\*z0=d1²-d4²+x4²-x1²+y4²-y1²+z4²-z1²; ……………..式7

**Solution：**根据式5 、6、 7可以直接联立方程求出x0,y0,z0

令：

2\*(x2-x1)=M11，2\*(y2-y1)=M12，2\*(z2-z1)=M13，d1²-d2²+x2²-x1²+y2²-y1²+z2²-z1²=N1；

2\*(x3-x1)=M21，2\*(y3-y1)=M22，2\*(z3-z1)=M23，d1²-d3²+x3²-x1²+y3²-y1²+z3²-z1²=N2；

2\*(x4-x1)=M31，2\*(y4-y1)=M32，2\*(z4-z1)=M33，d1²-d4²+x4²-x1²+y4²-y1²+z4²-z1²=N3；

M12\*M21-M22\*M11=K11，M13\*M21-M23\*M11=K12，N1\*M21-N2\*M11=T1;

M12\*M31-M32\*M11=K21，M13\*M31-M33\*M11=K22，N1\*M31-N3\*M11=T2;

**则结果为：**

**z0** = (T1\*K21-T2\*K11) / (K12\*K21-K22\*K11);

**y0** = (T1\*K21-K12\*K21\*z0) / (K11\*K21);

**x0** = (N1-M12\*Y0-M13\*z0) / M11;

附参考代码：

Int main()

{

double x1 = 107.67, y1 = -38.75, z1 = 127.67, d1 = 145.4541113,

x2 = 140.75, y2 = 68.22, z2 = 2.91, d2 = 123.9999927,

x3 = 269.29, y3 = 41.87, z3 = 269.91, d3 = 347.6197113,

x4 = 203.96, y4 = 163.75, z4 = 148.77, d4 = 261.2238496;

//赋值操作

//计算过程

double m11 = 2 \* (x2 - x1), m12 = 2 \* (y2 - y1), m13 = 2 \* (z2 - z1),

m21 = 2 \* (x3 - x1), m22 = 2 \* (y3 - y1), m23 = 2 \* (z3 - z1),

m31 = 2 \* (x4 - x1), m32 = 2 \* (y4 - y1), m33 = 2 \* (z4 - z1);

double n1 = d1 \* d1 - d2 \* d2 + x2 \* x2 - x1 \* x1 + y2 \* y2 - y1 \* y1 + z2 \* z2 - z1 \* z1,

n2 = d1 \* d1 - d3 \* d3 + x3 \* x3 - x1 \* x1 + y3 \* y3 - y1 \* y1 + z3 \* z3 - z1 \* z1,

n3 = d1 \* d1 - d4 \* d4 + x4 \* x4 - x1 \* x1 + y4 \* y4 - y1 \* y1 + z4 \* z4 - z1 \* z1;

double k11 = m12 \* m21 - m22 \* m11, k12 = m13 \* m21 - m23 \* m11,

k21 = m12 \* m31 - m32 \* m11, k22 = m13 \* m31 - m33 \* m11;

double t1 = n1 \* m21 - n2 \* m11, t2 = n1 \* m31 - n3 \* m11;

//结果

double z = (t1 \* k21 - t2 \* k11) / (k12 \* k21 - k22 \* k11);

double y = (t1 \* k21 - k12 \* k21 \* z) / (k11 \* k21);

double x = (n1 - m12 \* y - m13 \* z) / m11;

}

附（根据6个距离值求得4个点的坐标）

double GetCos(double LengthA, double LengthO, double LengthB)

{

return (LengthA \* LengthA + LengthB \* LengthB - LengthO \* LengthO) /

(2 \* LengthA \* LengthB);

}

Int main()

{

**//已知6个距离值**

double s\_12 = 140.75, s\_13 = 272.525596962928,

s\_23 = 135.187382917194, s\_41 = 117.261489842147,

s\_42 = 65.544484893849, s\_43 = 173.322982319137;

//得到P1和P2的坐标值

double p1\_x = 0, p1\_y = 0, p1\_z = 0,

p2\_x = 140.75, p2\_y = 0, p2\_z = 0;

double Cos213 = GetCos(s\_12, s\_23, s\_13);

double Sin213 = Math.Sqrt(1 - Cos213 \* Cos213);

//算得P3的坐标值

double p3\_x = s\_13 \* Cos213, p3\_y = s\_13 \* Sin213, p3\_z = 0;

//接下来计算P4的坐标

double p4\_x, p4\_y, p4\_z;

double CosMAS = GetCos(s\_41, s\_42, s\_12);

double CosMAC = GetCos(s\_41, s\_43, s\_13);

double CosCAB=Cos213;

double TanW = (CosMAC / CosMAS - CosCAB) / Sin213;

double cos\_ = CosMAS / (Math.Cos(Math.Atan(TanW)));

if (Math.Abs(cos\_) >= 1)

{

p4\_x = 0;

p4\_y = 0;

p4\_z = s\_41;

}

else

{

double AO = s\_41 \* cos\_;

p4\_x = AO \* Math.Cos(Math.Atan(TanW));

p4\_y = AO \* Math.Sin(Math.Atan(TanW));

p4\_z = Math.Sqrt(s\_41 \* s\_41 - AO \* AO);

}

}