

百科知识

比克曼

October 30, 2015

Contents

1	GPS	2
1.1	GPS 组成	2
1.1.1	空间段	2
1.1.2	控制段	2
1.1.3	用户段	2
1.1.4	GPS 通信信号	3
1.2	时间系统	3
1.3	经纬度转换	3
1.4	经纬度转换米	3
1.5	任意两点的经纬度计算两点距离	4

1 GPS

- **地球经纬线:**地轴是一根通过地球南北极和地球中心的假想线,在地球中腰画一个与地轴垂直的大圆圈,使圆圈的每个点都和南北极的距离相等,这个圆圈就叫做“赤道”,在赤道的南北两边,画出许多和赤道平行的圆圈叫纬线,赤道定为纬度零度,向南向北各 90° ,北极就是北纬 90° ,南极就是南纬 90° ;从北极点到南极点,画出许多南北方向的与地球赤道垂直的大圆圈,叫做经度,国际上规定以通过英国伦敦近郊的格林尼治天文台的经线作为计算经度的起点,即经度零度零分零秒,也称“本初子午线”;在它东面的为东经,共 180° ,在它西面的为西经,共 180° ;因为地球是圆的,所以东经 180° 度和西经 180° 度的经线是同一条经线;规定 180° 度经线为“国际日期变更线”。为了避免同一地区使用两个不同的日期,国际日期变更线在遇陆地时略有偏离;每一经度和纬度还可以再细分为 60 分,每一分再分为 60 秒以及秒的小数。利用经纬线,我们就可以确定地球上每一个地方的具体位置,并且把它在地图或地球仪上表示出来;经度和纬度都是一种角度。
- **经度:** 是个两面角,是两个经线平面的夹角。因所有经线都一样长,为度量经度选取一个起点面(本初子午线),本初子午线平面是起点面,终点面是本地经线平面,某一点的经度,就是该点所在的经线平面与本初子午线平面间的夹角。
- **纬度:**是个线面角。起点面是赤道平面,线是本地的地面法线。所谓法线,即垂直于参考扁球体表面的线;某地的纬度就是该地的法线与赤道平面之间的夹角。纬度数值在 0° 至 30° 之间的地区称为低纬地区,纬度数值在 30° 至 60° 之间的地区称为中纬地区,纬度数值在 60° 至 90° 之间的地区称为高纬地区。
- **ECI 坐标系:**地心惯性坐标系,earth centered inertial,用于描述卫星的位置,坐标系固定不变;
- **ECEF 坐标系:**地心地球固连坐标系,earth centered,earth fixed,用于描述地面接收器的位置,随着地球自转而旋转,在具体定位时,需要将卫星的 ECI 坐标系的位置转换成它在 ECEF 坐标系的位置;
- **星历表:**本来用来记录天体特定时刻的位置。而在 GNSS 中,星历表则记录了卫星的运行参数,它使我们通过星历表就可以计算出任意时刻的导航卫星的位置和速度,其数据量较大,传输时间长,也有简版星历表叫历书;
- **历书:**包含卫星的位置等信息;

1.1 GPS 组成

1.1.1 空间段

空间段由 GPS 卫星组成;

1.1.2 控制段

控制段用来控制和监视 GPS 的运行,控制段包含

- 主控站;
- 监控站;
- 地面控制站;
- 地面天线

1.1.3 用户段

- 民用用户:提供标准定位服务;
- 军用用户:提供精密定位服务;

1.1.4 GPS 通信信号

GPS 通信频率都位于 L 频段,共有 3 个频率;

- L1 频率:中心频率为 1575.42MHz,传输民用的 CA 码,军用的 P(Y) 码;
- L2 频率:中心频率为 1227.6MHz;
- L5 频率:中心频率为 1176.45MHz, 主要为航空安全服务;

GPS 信号包含几个部分

- 测距码:用来测量卫星和接收器间距离的信号,其本质是精心设计的伪随机噪声,也就是上面提到的民用 CA 码和军用 P(Y) 码;

1. CA 码:频率 $F=1.023\text{MHz}$,周期 $T=1\text{ms}$,码长 $N=1023$,码元宽度 $L=293.05\text{m}$,测距精度 2m 到 3m,光速为 C ,码长表示周期范围内有多少个脉冲码元, $N=F*T$,码元宽度计算公式:

$$L = T * C / N$$

2. P(Y) 码:频率 10.23MHz ,周期 7 天,码长 6.1871×10^{12} ,码元周期 $0.097752\mu\text{s}$,码元宽度为 29.3m,测距精度为 0.3m,对 P 码加密后就是 Y 码;

- D 码:也叫导航电文,星历和时间信息就封装在导航电文中,其传输速率为 50bit/s

1.2 时间系统

- 国际原子时钟:IAT,international atomic time;
- 协调世界时钟:UTC,coordinated universal time,互联网使用的网络时间协议 (NTP),获取的时间就是 UTC,格式年月日时分秒;
- GPS 时间:GPST,gps time,以 IAT 时钟为准,时间原点定于 1980 年 1 月 6 日 0 时;
- 本地时间:local time, 基于 UTC 时间,将全球分为 24 个时区,每个时区的中心为相隔 15 度经线,第一时区的中心位于子午线;

1.3 经纬度转换

- 度转度分秒如何将度 (DDD): 108.90593 度换算成“度分秒 (DMS)“,东经 E 108 度 54 分 22.2 秒? 转换方法是将 108.90593 整数位不变取 108(度), 用 $0.90593*60=54.3558$, 取整数位 54(分), $0.3558*60=21.348$ 再取整数位 21(秒), 故转化为 108 度 54 分 21 秒.
- 度分秒转度同样将度分秒 (DMS): 东经 E 108 度 54 分 22.2 秒换算成度 (DDD) 的方法如下: $108 \text{ 度 } 54 \text{ 分 } 22.2 \text{ 秒} = 108 + (54/60) + (22.2/3600) = 108.90616$ 度;因为计算时小数位保留的原因,导致正反计算存在一定误差,但误差影响不是很大。1 秒的误差就是几米的样子。

1.4 经纬度转换米

- 子午线总长度大约 $L=40008\text{km}$;
- 赤道半径 = 大约 6378.140km ;
- 极半径 = 大约 6356.755km ;
- 地球平均半径 = 大约 6371.004km ;
- 纬度 1 度 = $(L/2)/180 =$ 大约 111km ;
- 纬度 1 分 = $111\text{km}/60 =$ 大约 1.85km ;
- 纬度 1 秒 = $1.85\text{km}/60 =$ 大约 30.9m ;

1.5 任意两点的经纬度计算两点距离

地球是一个近乎标准的椭球体,它的赤道半径为 6378.140 千米,极半径为 6356.755 千米,平均半径 6371.004 千米。如果我们假设地球是一个完美的球体,那么它的半径就是地球的平均半径,记为 R。如果以 0 度经线为基准,那么根据地球表面任意两点的经纬度就可以计算出这两点间的地表距离(这里忽略地球表面地形对计算带来的误差,仅仅是理论上的估算值)。设第一点 A 的经纬度为 (LonA, LatA),第二点 B 的经纬度为 (LonB, LatB),按照 0 度经线的基准,东经取经度的正值 (Longitude),西经取经度负值 (-Longitude),北纬取 90-纬度值 (90- Latitude),南纬取 90+ 纬度值 (90+Latitude),则经过上述处理过后的两点被计为 (MLonA, MLatA) 和 (MLonB, MLatB)。那么根据三角推导,可以得到计算两点距离的如下公式:

```
1 C = sin(MLatA)*sin(MLatB)*cos(MLonA-MLonB) + cos(MLatA)*cos(MLatB)
2 Distance = R*Arccos(C)*Pi/180
```

这里,R 和 Distance 单位是相同,如果是采用 6371.004 千米作为半径,那么 Distance 就是千米为单位,如果要使用其他单位,比如 mile,还需要做单位换算,1 千米 =0.621371192mile; 如果仅对经度作正负的处理,而不对纬度作 90-Latitude(假设都是北半球,南半球只有澳洲具有应用意义) 的处理,那么公式将是:

```
1 C = sin(LatA)*sin(LatB) + cos(LatA)*cos(LatB)*cos(MLonA-MLonB)
2 Distance = R*Arccos(C)*Pi/180
```

以上通过简单的三角变换就可以推出。如果三角函数的输入和输出都采用弧度值,那么公式还可以写作:

```
1 C = sin(LatA*Pi/180)*sin(LatB*Pi/180) +
    cos(LatA*Pi/180)*cos(LatB*Pi/180)*cos((MLonA-MLonB)*Pi/180)
2 Distance = R*Arccos(C)*Pi/180
```

也就是:

```
1 C = sin(LatA/57.2958)*sin(LatB/57.2958) +
    cos(LatA/57.2958)*cos(LatB/57.2958)*cos((MLonA-MLonB)/57.2958)
2 Distance = R*Arccos(C) = 6371.004*Arccos(C) kilometer =
    0.621371192*6371.004*Arccos(C) mile =
3 3958.758349716768*Arccos(C) mile
```

在实际应用当中,一般是通过一个个体的邮政编码来查找该邮政编码对应的地区中心的经纬度,然后再根据这些经纬度来计算彼此的距离,从而估算出某些群体之间的大致距离范围 (比如酒店旅客的分布范围 -各个旅客的邮政编码对应的经纬度和酒店的经纬度所计算的距离范围 -等等),所以,通过邮政编码查询经纬度这样一个数据库是一个很有用的资源。附:C# 代码:

```
1 private const double EARTH_RADIUS = 6378.137; //地球半径
2 private static double rad(double d)
3 {
4     return d * Math.PI / 180.0;
5 }
6
7 public static double GetDistance(double lat1, double lng1, double lat2,
8     double lng2)
9 {
10     double radLat1 = rad(lat1);
11     double radLat2 = rad(lat2);
12     double a = radLat1 - radLat2;
13     double b = rad(lng1) - rad(lng2);
14     double s = 2 * Math.Asin(Math.Sqrt(Math.Pow(Math.Sin(a/2), 2) +
15         Math.Cos(radLat1)*Math.Cos(radLat2)*Math.Pow(Math.Sin(b/2), 2)));
16     s = s * EARTH_RADIUS;
17     s = Math.Round(s * 10000) / 10000;
18     return s;
```

