



赛伯空中的花样飞行

我紧张不安地仔细打量了我的塞斯纳 182 飞机上的仪表盘。不要人指导,就试着单独飞行,这是不是有点太荒唐了? 应不应该找一个有经验的人带我上天呢? 座舱外面就是空荡荡的芝加哥 Meigs 机场,露天里看不见一个人影,甚至没有一个机械师走出来到阳光下吸一支烟。在这次疯狂的飞行中,我惟一的伙伴就是膝上的飞行手册。我已花 10 分钟把它读过一遍。

我想我对使用操纵杆或方向盘本应是更有信心的。孩提时,我至少看到过 10 次海军的巫婆式飞机,对基本动作已经有所了解。不过,现在这架飞机明显地先进得多,



它完全由我面前的键盘控制。

我叹了一口气,翻开了手册第 46 页,手册上指出,按下键 F_2 ,油门就会慢慢打开,而键 F_4 的作用是加大油门。想到飞行有危险,我试探地按了一下 F_2 。背后发动机的嗡嗡声顿时明显地增强了,跑道开始朝我移动,霎时间,我感到有点恐慌。飞机几乎立刻就开始离开混凝土跑道。由于无知,我忘了读手册上有关方向舵和前轮的部分。现在惟一的救星就是副翼了,它们由标记为 4(左倾斜)和 6(右倾斜)的按键操纵。使我惊喜的是,我用其中一个键就使飞机慢慢地返回了跑道。

前面仅 100 码处,密执安湖深蓝色的湖水正朝着我逼近。必须加大油门。我将 F_2 连按四次。接着再加大一些。发动机响得更厉害了,但飞机并未升起。跑道在我下面越来越快地向后飞驰,我又一次感到恐慌。哪一个奇妙的按钮能让飞机飞起来呢?刹时我一生的经历在眼前闪过。我想起童年时试飞一架模型飞机的情景,这飞机仅靠机尾的一对襟翼进行控制。升降舵!我兴奋地在手册上查看“升降舵”。升降舵控制飞机的冲角;拉起它就会使机头抬起来。于是我及时按了标记为 2 的键,避免了葬身湖底的悲剧,地面和湖泊的轮廓慢慢地向后退去。我在飞行了!飞机正在自动向上升。密执安湖西岸的湖滨线在我的视野里逐渐变得清晰起来,我向北飞了一会儿,陶醉于我所取得的成功之中。

可能许多读者现在已经猜测到,飞机的座舱就是我屋里的书房,所有控制按钮都在我的个人计算机的键盘上,计算机上正运行着一个用于模拟飞行的程序。不过,熟悉的环境丝毫不减少我在冒险的感觉;飞机和风景纯粹是模拟的,这一事实只是进一步增强了我的想像力。在模拟飞行中人们可以去试做那些在真正飞机里决不会做的事情。赔偿损失的账单也只是虚构的。我将飞机转向西南,朝着芝加哥飞去。我打算突然掠过密执安大街,用我的大胆行为给来往车辆留下深刻的印象,仪表盘上的钟给出的时间是四点半,正是晚间交通高峰开始的时刻。

当我接近芝加哥时,我发现那里似乎是出了什么严重问题。沿着湖边只能看见为数不多的建筑,街道上不见车辆,城内全然无人。是遭到了核攻击吗?我注意到我错把有两个指针的高度计误当作时钟了。但是,即使发现时间离 4 点半还很远,也解释不了这个大城市空无一人的现象,最终我只得沮丧地承认:飞机模拟程序中本身没有人。它的数据库中已存满了有关北美风景的数据,自然也存贮了用于模拟四个大城市地区以及 20 多个机场的地理细节。

在后来的几次飞行中,我越来越不把无人作伴放在心上了。模型飞行实在有趣,其他许多人也明显地有此同感。该模拟程序叫做 FLIGHT SIMULATOR(飞行模拟



程序)。飞行模拟很久以来一直是专供飞行员训练用的。现在实际上已成为一种公众娱乐了。

我们来看看飞行模拟程序必须做些什么才能使用户自动放弃他们对这种模拟的怀疑。首先,它必须以每秒钟若干次的速度产生和再产生从一个特定的观察点看到的风景。由于模拟的飞机在模拟的天空中往返飞行,观察点一直在改变,因而风景的面貌必须反复计算。此外,对于每一个观察点,必须有一个合适的视角。通过模拟飞行中的飞机的运动情况就可推导出这个视角。例如,当飞机倾斜和转弯时,其视角以及观察点必须作相应的移动。基本的计算周期中用到了许多独特的、非常吸引人的诀窍与折衷手法。

地理数据库是 FLIGHT SIMULATOR 程序的基础。该数据库是一个巨大的表格,表格中列有种种景物,用以表现飞机可能要飞越的(或避免撞上的)风景画面。你可以通过追踪一组自顶向下分层的信息而获取你在飞行时看到的情景。一个标为 U. S. A. 的存贮区存放着如像芝加哥、纽约和西雅图等地区的名字。飞机的位置则由三个面向挡风玻璃的有序坐标确定: x 为纬度, y 为以米为单位的海拔高度, z 为经度。然后,FLIGHT SIMULATOR 程序将坐标与 U. S. A. 存贮区内为每一地区所规定的边界进行比较。例如,如果 x 和 z 决定的一点在芝加哥地区,程序就执行一个叫做“数据库装入”的操作,即将描述芝加哥地区的表格从磁盘装到计算机的主存中去。

182

当我刚把飞机转向芝加哥时,看不见任何建筑。但过了一会儿,西尔斯大厦突然进入了我的眼帘,再过一会儿,连细节都看得清楚了。这是 FLIGHT SIMULATOR 程序为了不超出微计算机速度和存贮容量的限制而采用的一种策略:不到确实需要时,就不显示风景中的某一景物。程序中有两个专门的检验程序段来确定画面是否出现。第一个是 ifin 2d,它只检查飞机的经度和纬度两个坐标。我已经描述了它是怎样根据这两个坐标来确定一个大的地理区域的。

第二个检查程序叫做 ifin 3d,它决定周围风景中的某一景物是否近得足以显示出来。当我接近芝加哥时,ifin 3d 不断地反复计算我离某些确定的市区地界的三维距离。当我距西尔斯大厦的距离在 12 800 米以内时,这大厦的缩小形象就突然在风景中出现,在屏幕上它表现为远处的一个由像素堆叠起来的蠕动着的情景。当我飞得更近点,到距离为 7 680 米时,就可突然看到更清晰的画面。对于每一个新的坐标集合,FLIGHT SIMULATOR 程序都要飞快地检查存贮器中存贮的显示表格上的所有景物,并根据 ifin 3d 中的显示规则检查这些景物。如果某一景物通过了检查,程序就显示出从当前的观察点所看到的它的景象;如果未能通过检查,则不显示出来。此时程



序转到表格中的下一个景物(见彩图 22)。

一般说来,模拟风景中的每一景物都是由点、线及面的表格定义的。由于用户看到的画面每秒计算五次,快得足以与视觉保持一致,所以图中的线与面都必须在尽可能短的时间内画出。因此,即使是程序中的基本绘图操作都要非常仔细地加以设计。

例如,由于微型计算机上完成乘除法花的时间是加、减法的 10~100 倍,显示一条线的基本方法必须避开乘除法,而应注重于仔细分析每一条指令所花的计算时间。这样得出的一种较聪明的画线算法基本上由四条指令(加法指令、检验指令、分支指令与绘图指令)组成。假定程序刚把点(a,b)画在(正)斜率为 1/4 的一条线上,下一步它就进入一个循环,在此循环中程序重复地将 a 加 1,并将斜率 1/4 与一个叫做跟踪误差的特殊和数相加,然后检查跟踪误差。当跟踪误差大于或等于一个正整数时,程序就退出循环,画出下一点,本例中即是点(a+4,b+1)。

FLIGHT SIMULATOR 程序画面的方法叙述起来有点麻烦,但执行起来却非常快。简言之,程序先画出表示面的边界的多边形,然后再画出里面的景物。当多边形的形状计算完毕时,绘图程序首先确定监视器上穿过多边形的扫描线。对于每一条这样的扫描线,还要确定它穿过多边形的哪些边,并按从左到右穿过的先后次序对这些边排序。然后,程序考虑扫描线从左到右的第一个像素。当扫描线遇到多边形的第一条边时,就开始显示色彩与所表现的面相一致的点。当遇到下一条边时,则停止点的显示。以后再遇到边时,又开始重复显示点的过程。

这种时画时停的用于扫描的画图技术以平面拓扑的一个基本结论为基础。任一多边形都将平面分为内部与外部两部分。因此,任何穿过多边形的一边的直线都是由内部(即要显示的面)到外部(即非面的部分),或是由外部到内部。FLIGHT SIMULATOR 程序画的大多数面的边界,都是比较简单的多边形。例如,我刚从 Meigs 机场起飞不久,密执安湖的湖岸线就从飞机下面延伸到远方的地平线。航线上接着出现的是被一片朝北的岬地部分包围着的多边形湖湾。湖湾中蓝色湖水的扫描线在岬地出现的地方中断了一下,然后又继续扫描湖的主要部分。

图形显示程序的设计者曾常常考虑所谓的虚线问题:程序怎样画一个不透明的三维物体,以使得只有离观察者最近的一侧才能够看得见呢?某些方法要求解需耗费大量时间的方程,这样大的计算量是 FLIGHT SIMULATOR 程序所承受不了的。解决方法是画出整个物体,但离观察者最近的一侧要放在最后画。这一方法在很大程度上只适用于简单的风景。使用这一方法时,物体上观察者看不见的那些部分就被看得见的部分抹去了。对纽约的世界贸易中心的两座大楼的画法,就是这种粗糙但有效的技



术的一个极好的例子。从任何角度看上去,每座大楼的最近的一侧都抹掉了最远的一侧,而较近的一座大楼则遮住了较远的大楼的一部分。

发现芝加哥没有人后,我决定调头飞回 Meigs 机场。飞机的真正主人现在可能正急切地想知道飞机的下落。但是我仍有许多东西想要学习。其一是,我只知道少数几种按钮的用法,这使我感到烦恼;其二是,我还必须试作一次真正激动人心的特技飞行。为了把我的打算变成现实,我飞到了密执安湖的上空。

我很喜欢用副翼来控制飞行。按一下控制左副翼的键,飞机便向左倾斜,我前面的地平线向右倾斜,地面风景似乎慢慢地沿斜坡下滑。根据空气动力学定律,塞斯纳飞机的机头逐渐向左转。飞机再次朝前直飞并拉平时,我按了几下标有“8”的按钮,身体几乎感觉到机头在下坠。窗外的地平线和风景上升到了视野的顶部。飞机正在俯冲。速度表表明飞机速度超过了每小时 150 英里。我试着用方向舵控制。按了一下标有“-”号的键,方向舵便稍稍向右转,飞机开始急剧转弯。我又将方向舵移回到中间,注视着下面的风景以慢得使人容易发生误解的速度朝我爬了上来。是进行特技飞行的时候了。

184 飞行员驾驶飞机翻筋斗的时候远比程序员碰到循环的时候少得多。但是我自然知道,要完成这个飞行动作,需要有较高的飞行速度。首先向下,接着向上,然后翻过来。虽然我并没意识到按键就等于拉回操纵杆,我还是本能地按了键 2。飞机轰鸣着从俯冲中拉了起来,回到水平飞行,接着又升入纯蓝的天空中。看到地平线降到了挡风玻璃的下边后,我就等待着。对能不能完成翻筋斗,我心中完全无数。此刻是一片寂静。我会因为失速而笔直地摔到地上吗?飞机上的仪表无所帮助;我弄不清楚有一半仪器是用来干什么的。

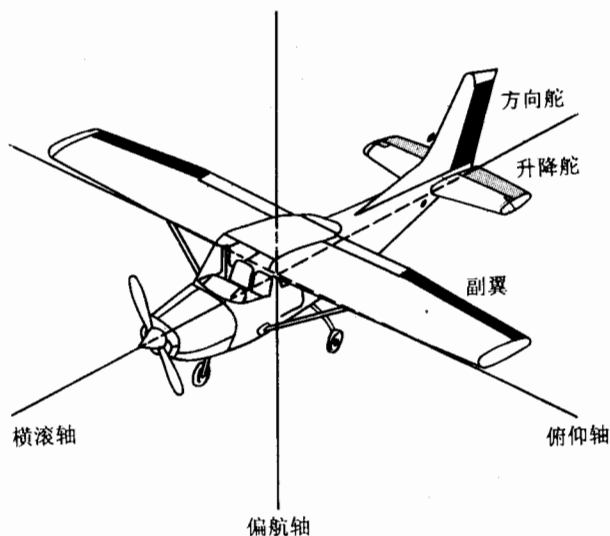
忽然,窗口上出现了颠倒的风景,绿地在上面,蓝天在下面。我翻起筋斗了吗?我已经弄得糊里糊涂,自己也回答不了这个问题。没有办法,我只得咬紧牙关坚持下去。现在,前面只看得见大地了。紧接着地平线升到了正常的位置上。我试图使机头向下,以免飞机再翻筋斗。地平线晃动着,最后稳定下来。我成功了!现在很清楚,我是一个几乎不靠任何指导就能熟练地飞行的天才!前方,Meigs 机场突然出现了。我打算着陆,然后舒舒服服洗个澡。

跑道由南向北延伸。我的飞机现在处于机场的东南。我打算先朝西北,然后转北对准跑道。但不知出了什么差错,飞机总是对不准跑道,接近机场时的高度太高。情况越来越糟,简直像是一场恶梦。我使机头朝下,以便迅速降低高度,同时,试图使飞机朝东再偏移几码。但是,跑道离开了屏幕,我看不到地平线了。当大地向上冲来时,响起了报



警的嘟嘟声。天啊！飞机坠毁了。挡风玻璃全是裂纹，上面已打印出“坠毁”二字。我死了吗？如果芝加哥数据库失去了它惟一的活着的居民，那将是莫大的遗憾。

为了模拟这些飞行过程，FLIGHT SIMULATOR 程序必须一直反复计算从塞斯纳飞机的座舱所看到的风景。飞机的控制装置可按照它们所产生的三种旋转分类（见图）。副翼操纵飞机绕横滚轴（即从机头到机尾的轴）转动；升降舵操纵飞机绕俯仰轴（即平行于机翼的轴）转动；方向舵则使飞机绕偏航轴（即对飞行员说来是垂直的轴）转动。如果飞机的现在位置由坐标 x 、 y 和 z 给出，那么计算机怎样确定瞬间后飞机的位置呢？



控制一架飞机的转动的三种控制面

塞斯纳飞机机头的方向和当前的飞行速度构成了一个矢量，由该矢量可确定出一个新的位置。程序通过飞行方程为飞机计算出新的方向，从而也就得到一个新的视角。对于每一个显示周期，方程的输入包括控制装置及环境条件（如空气压力及密度、风速、温度等）的当前值。飞行方程包含了飞机的飞行特性，并将物理定律运用到这些输入值上。推力，重力，升力和阻力也要加入方程，还须考虑飞机质量的惯性作用及几何形状。方程的输出是新的飞行方向。

飞行方程太复杂，难以快速求解，因此，采用若干数值表格来取代方程。例如，其中一个表格列出了相应于每一个冲角的飞机升力。冲角是以 $1/10$ 度为间隔给出的。采用表格还有另外一个优点：只须改变表格就可以使飞行特性改变，不必再推导新的公式。



数学迷宫

经过几次处女飞行后,读者可能就想去试试一些惊险得令人毛发直竖的冒险飞行了。当用户启动任何一个飞行模拟程序,并键入书中给出的必要的的数据时,一次冒险飞行便开始了。要键入的数据包括纬度、经度、高度、控制装置的位置、时刻等等。

很快地你就可能发觉你离曼哈顿大桥仅有数百码远了,这是一次让你直接从桥下飞过的冒险飞行。或者你也可能进行一次名叫“在圣克利门蒂附近的熄火飞行”的冒险,这时你的飞机在加利福尼亚海岸外圣克利门蒂岛的上空 5 000 英尺处,而发动机刚刚熄火。你能完全靠滑翔在岛上着陆吗?还有一种叫做“边缘地区”的冒险飞行,这时飞机停在草坪跑道上,仪表盘上的钟给出的时间是零点零三分,而外边是亮的。飞机除了从边窗和尾窗看去,可以看见窗外,是一片漆黑。飞机慢慢起飞,一旦升空,不管是窗口还是雷达都不能显示出任何风景。该怎么办呢?就我个人来说,我情愿用一个无人居住的数据库。