

# 基于 VRML 的机动目标 CCD 跟踪的视景仿真

周云锋, 单甘霖, 吉 兵

(军械工程学院光学与电子工程系, 石家庄 050003)

**摘 要:** 实现机动目标 CCD 跟踪的视景仿真对装备训练和基于图像的跟踪算法设计具有重要意义。该文介绍在 Matlab 软件平台下利用基于 VRML 的虚拟现实工具箱进行视景仿真的基本步骤, 以机动目标 CCD 跟踪的视景仿真为目的, 针对虚拟场景的构建、3D 目标机动和虚拟 CCD 跟踪等问题, 提出相应的解决方法, 并给出仿真结果。

**关键词:** 视景仿真; 机动目标; 虚拟现实; 跟踪

## VRML-based Visual Simulation for Maneuver Target CCD Tracking

ZHOU Yun-feng, SHAN Gan-lin, JI Bing

(Dept. of Optical and Electronic Engineering, Ordnance Engineering College, Shijiazhuang 050003)

**【Abstract】** Realization of visual simulation of maneuver target CCD tracking is valuable for military equipment training and tracking-algorithm design based on target image. This paper introduces steps to perform a visual simulation using the VRML-based virtual reality tool box of Matlab platform, and for the visual simulation of maneuver target CCD, aiming at the problems of the building-up of visual scene, the maneuver of 3D target and the visual CCD tracking, some solutions are accordingly presented with a simulation result.

**【Key words】** visual simulation; maneuver target; virtual reality; tracking

### 1 概述

基于 CCD 的机动目标跟踪系统是 C3I 系统很重要的光学传感器节点, 它是雷达、红外等跟踪系统的必要补充, 其获取的机动目标动态图像数据不仅能为其他跟踪/识别提供辅助支持, 同时也是进行战场态势评估和决策的重要参考。实现机动目标 CCD 跟踪的视景仿真一方面有利于提高我军的装备训练水平, 另一方面也可以为基于图像的机动目标跟踪/识别算法的设计提供重要的模拟测量数据。

在虚拟现实视景仿真领域, 可用的编程语言或软件环境较多, 如 Vega, OpenGL Performer, OSG(Open Scene Graph), VRML(Virtual Reality Modeling Language)等。其中, VRML 是一种面向网络应用的交互式三维造型和渲染的图形描述性建模语言, 通过各种不同的节点来构建虚拟的现实世界<sup>[1-2]</sup>。与传统的 STL 文件相比, VRML 模型在几何和拓扑的健壮性以及数据的冗余量等方面具有明显的优势<sup>[3]</sup>, 而且适应未来远程制造和网络交流的趋势, 在军事、航空航天、电子商务和医学等领域具有广阔的应用空间。Matlab 虚拟现实工具箱以 VRML 为基础, 其最大特点在于实现了 VRML 和 Matlab 软件平台的无缝链接, 即在实现虚拟现实(VR)仿真的同时, 能够直接利用 Matlab 强大的数值计算能力完成视景仿真的诸多后续计算, 如图像处理、目标跟踪等, 而无须像 Vega 等软件那样需要借助第三方编程语言(如 C++)的转换来实现和 Matlab 的交互。

### 2 实现步骤

在 Matlab 软件环境下进行 VR 视景仿真, 首先需要利用 VRML 编程/建模软件生成的静态的虚拟场景(包括虚拟背景、视点、光源、地面等)和待跟踪目标的静态三维造型。一般情况下, 面元不多的 3D 造型可用 VmlPad 或 V-Realm

Builder 等可视化 VRML 编程软件实现; 而复杂场景或精细 3D 造型的建立通常用 3D MAX 等大型专业三维建模软件完成, 之后导出为\*.wrl 文件, 再在 VRML 编程软件中调用。

利用 Matlab 与 VRML 的接口驱动场景中的相关节点实现动态效果。Matlab 和 VRML 交互方式有 2 种: (1)利用 Simulink 工具箱, 生成驱动数据, 通过 VRML 的数据接收端口, 传递给 VRML 模块的相关节点; (2)直接通过编程实现, 在 M 文件中通过程序循环设置 VRML 中相关节点的状态变量, 驱动 VRML 目标的运动, 实现动态效果。本文所述的 VR 视景仿真采用了上述第(2)种方法, 其中需要驱动节点包括 3D 机动目标和虚拟 CCD 传感器。其中, 3D 目标的运动数据包括目标的位移和姿态; 而 CCD 传感器的驱动数据包括其跟踪线的指向和光学变焦的控制参数。

### 3 关键问题及其解决方法

用 Matlab 的 VR 工具箱实现本文的视景仿真, 关键要解决下面的 3 个问题。

#### 3.1 虚拟场景的构建

在 VRML 中, 默认背景是一个半球体。通过设置 Background 节点的 groundColor, groundAngle, skyColor 和 skyAngle 这 4 个域的值可以改变该球体天空和地面的颜色以及天空和地面的连接视角, 从而模拟出只有颜色变化的单纯背景。

然而, 在实际的基于 CCD 的目标跟踪中, 背景往往比较复杂, 可能包含了天空中的云、地面的山体、植物等复杂目

**基金项目:** 国家部委科研基金资助重点项目

**作者简介:** 周云锋(1979 -), 男, 博士研究生, 主研方向: 导航、制导与控制; 单甘霖, 教授; 吉 兵, 博士研究生

**收稿日期:** 2009-01-14 **E-mail:** zyf7962@163.com

标,所以,仿真中应该包含这些元素。复杂背景的模拟可以通过 Background 节点的 frontUrl, backUrl, leftUrl, rightUrl, topUrl 和 bottomUrl 6 个域来实现,它们代表了虚拟场景的前、后、左、右、上、下 6 个背景平面。在这些平面上贴上所需要背景图片就可以模拟出复杂背景,其关键就在于背景的平滑连接问题。由于上述 6 个面都是直角连接的,因此贴上去的图片通常在连接处会有明显的分界线,见图 1(a),显然这和真实情况不相符。解决该问题的方法就是利用其他图像处理软件(如 COOL360 或 Quick time VR 等)事先用多幅背景图片生成柱状的全景视图<sup>[4]</sup>,然后把该图分割以后再贴到背景平面上。这样,从跟踪系统的视角上看,背景就是平滑的,如图 1(b)所示。柱状全景视图不包括背景的上表面,要覆盖上表面就需要生成球面全景视图,这在 VRML 中相对复杂。



(a) 一般背景



(b) 平滑的背景

图 1 平滑背景的实现

除此之外,虚拟场景中的地面通常可用 Indexed Face Set 节点加上纹理映射来实现;而机动目标的静态造型用 3D MAX 完成。

为了方便用浏览器查看虚拟场景,通常还需要设置光源和多个视点。

### 3.2 驱动 3D 机动目标

在构建观视景仿真所需的虚拟场景以后,下一步要做的就是驱动静态的 3D 目标在其中实施机动。

先定义相关坐标系。在 VRML 中,大地坐标系定义为:  $y$  轴垂直向上,  $x$  轴指向观测者的右侧,而  $z$  轴指向计算机的显示屏。目标的体坐标系为:原点固定于目标中心并随目标移动,各坐标轴指向与大地坐标系一致。

同时,定义目标的姿态角。以飞机为例,假定其初始姿态为(从飞行员的角度看):飞机员的正前方  $z$  轴正方向一致,其左手指向  $x$  轴正方向,而其头顶为  $y$  轴正方向;则目标的姿态角:偏航角  $\beta_y$ ,高低角  $\beta_x$  和横滚角  $\beta_z$  分别为绕体坐标系的  $y$  轴、 $x$  轴和  $z$  轴所旋转的角度,角度的正负符合右手螺旋规则。

要驱动静态 3D 目标机动,就需要生成目标的运动点迹和姿态角数据,以分别确定目标在场景中的位移和旋转变换。

在 VRML 通过 transform 节点的 translation 域来控制目标在其父坐标系中的位移,而通过 center 和 rotation 域来控制目标的旋转。为了让目标在虚拟的场景中连续地移动,只需

要把目标三维点迹数据连续地传递给 translation 域。

对机动目标姿态角的控制相对复杂。VRML 通过 transform 节点的旋转变换实现目标姿态的变化。该旋转变换由一个 4 维的浮点型矢量  $v=[x \ y \ z \ \psi]$  决定,其中,前 3 个分量  $x, y, z$  确定了一个旋转轴,分量  $\psi$  为旋转的角度;而通常在目标跟踪系统中能够提供的姿态角数据是三维浮点型状态矢量  $[\beta_x \ \beta_y \ \beta_z]$ 。这样,提供的目标姿态数据并不能直接作用于 transform 节点来驱动 3D 目标旋转。

在 VRML 中解决该问题的方法有以下 2 种:

(1)通过数学方法把绕体坐标系 3 个坐标轴的旋转变换转换为绕单一旋转轴的旋转变换。定义绕  $x, y, z$  轴的姿态变换矩阵分别为  $T_{\beta_x}, T_{\beta_y}, T_{\beta_z}$ , 则总的旋转变换为

$$T = T_{\beta_x} \cdot T_{\beta_y} \cdot T_{\beta_z} \quad (1)$$

根据矩阵论的知识可知<sup>[5]</sup>,任意非奇异矩阵

$$T_A = [a_{ij}]_{3 \times 3} \quad (2)$$

确定了绕某一旋转轴的旋转变换,该旋转轴为  $T_A$  的特征向量,而旋转的角度

$$\psi = \cos^{-1} \left( \frac{\sum_{i=1}^3 a_{ii} - 1}{2} \right) \quad (3)$$

令  $T_A = A$ , 即可根据三维的姿态角向量求出旋转轴和旋转角度。需要注意的是,在求解  $T_A$  的特征向量时,常常会出现与旋转轴向量截然相反的结果。从矩阵论的角度来说,这是合理的,因为如果  $a$  为  $T_A$  的特性向量,则  $-a$  必然也是其特征向量。然而,绕  $a$  和绕  $-a$  旋转,结果会截然相反。

可以在程序中对此问题进行纠正。假定  $k-1$  时刻求得的特征向量  $\alpha_{k-1}$  是正确的,  $k$  时刻矩阵  $T_A$  的特征向量为  $\hat{\alpha}_k$ , 分别计算二阶矩  $d_1 = \|\hat{\alpha}_k - \alpha_{k-1}\|_2$  和  $d_2 = \|-\hat{\alpha}_k - \alpha_{k-1}\|_2$ 。如果  $d_1 > d_2$ , 则  $k$  时刻正确的特征向量  $\alpha_k = -\hat{\alpha}_k$ ; 否则  $\alpha_k = \hat{\alpha}_k$ 。该方法适用于前后 2 个时刻姿态角机动不超过  $\pi/2$  时的情况。

(2)利用 3 个 transform 节点进行“嵌套”。由于 transform 节点不具有状态记忆功能(即它的每次变换都是以节点的初始状态为基准),因此无法用单个 transform 节点绕其 3 个体坐标轴的累计旋转变换来完成上述姿态变换。要实现多次变换的累计,就需要用到节点嵌套,即让第 2 个 transform 节点做第 1 个节点的子节点,而第 3 个节点做第 2 个节点的子节点。然后,让第 1 个节点只做绕坐标轴  $x$  的旋转变换,第 2 个节点只做绕  $y$  轴的变换,而第 3 个坐标轴只做绕  $z$  轴的变换。这样,就实现了  $T_{\beta_x}, T_{\beta_y}, T_{\beta_z}$  变换的累积。该方法避免了上述第 (1)种方法的复杂计算,但是需要更多的 VRML 代码。

### 3.3 CCD 跟踪仿真

在 VRML 中,可以用视点(viewpoint 节点)来模拟 CCD 传感器,定义该视点为 CCD 视点。CCD 跟踪的仿真涉及到以下 3 个问题:

(1)目标跟踪。CCD 视点的初始位置在原点  $O$ , 指向  $z$  轴正方向。如图 2 所示,为了使 CCD 视点跟踪目标(即保持目标的成像处于成像平面的中央),就须使其指向与跟踪线  $OD$  一致。跟踪线  $OD$  由 CCD 视点和目标的当前位置决定。根据目标当前位置  $D(x_t, y_t, z_t)$ , 可得到目标球坐标  $(\theta, \varphi, R)$ , 其分量分别为方位角、高度度和球半径。要使 CCD 跟踪目标,就须使 CCD 视点从其初始位置  $OB$  经过 2 次旋转变换到  $OD$ 。由 3.1 节的分析可知,为了实现该变换,需要构建一个 2 层的 transform 节点嵌套,然后把 CCD 视点设定为里层 transform

节点的子节点。这样,对于任意目标点  $D$ ,驱动 CCD 视点在第 1 个 transform 节点里绕  $x$  轴旋转角度  $\varphi$  到  $OC$ ,然后在第 2 个 transform 节点绕  $y$  轴旋转角度  $\theta$  到  $OD$ ,就实现了对目标的跟踪。

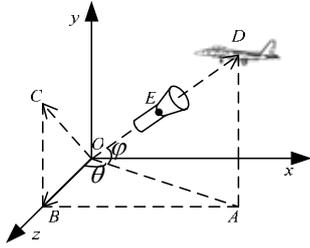


图 2 CCD 视点跟踪

(2)光学变焦。CCD 跟踪系统的光学变焦能力由其光学变焦镜头决定,目的是为了增大目标的成像尺寸,从而增大目标的发现距离,其作用相当于把远处的目标拉近。根据该原理,在 VRML 中,CCD 视点光学变焦的仿真可以利用它在父节点(即 transform 节点)的位移变换实现,即只需要沿着跟踪线向目标方向/后移动视点就可以增大/减小虚拟 CCD 的光学变焦镜头的成像放大倍数。如图 2 所示,此时 CCD 的光学变焦倍数为

$$\lambda = OD/ED \quad (4)$$

于是,CCD 视点的位移即为

$$[x_E, y_E, z_E] = [x_t, y_t, z_t] \cdot (\lambda - 1) / \lambda \quad (5)$$

(3)视角。视角反映了观测范围的大小。VRML 中 viewpoint 节点的 fieldOfView 域决定了视角的大小,默认值为  $45^\circ$ 。通常 CCD 跟踪系统的视角要远小于这个角度。所以,CCD 视点的视角要根据实际情况作相应的调整。

#### 4 仿真结果

定义 CCD 的光学变焦倍数为 10 倍,视角为  $7^\circ$ ,位于大地坐标系的原点。

目标点迹如图 3 所示。目标起始点位置为  $x_0 = -3 \times 10^3$  m,  $z_0 = 1 \times 10^4$  m,  $z_0 = 2 \times 10^3$  m, 初始速度  $v_{x0} = 200$  m/s,  $v_{z0} = -180$  m/s,  $v_{y0} = 0$ , 目标的运动模式和持续实现依次为 CV(20 s), R-CT(50 s), CV(50 s), L-CT(40 s)和 CV(40 s),其中, CV 为匀速直线运动; R-CT 为右拐弯; L-CT 为左拐弯; 拐弯的角速率都为  $0.0524$  rad/s; 数据点的间隔为 1 s。

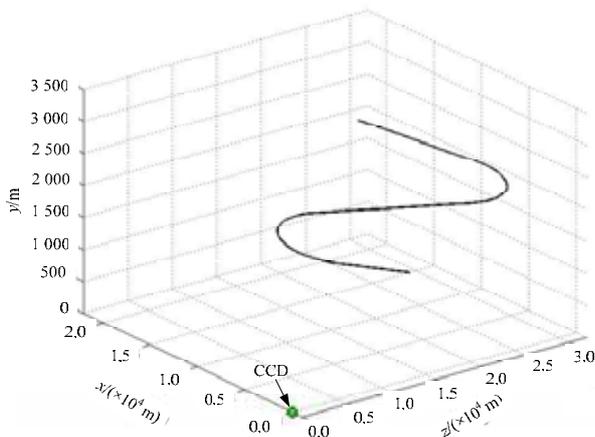


图 3 目标运动点迹

目标的姿态角数据如图 4 所示,图 4(a)~图 4(c)分别为对应于点迹的偏航角、高低角和横滚角。

仿真结果如图 5 所示,其中,图 5(a)为第 1 段直线运动中的某点;图 5(b)~图 5(g)为右拐弯机动中的采样点;图 5(h)为第 2 段直线运动中的某一点;图 5(i)为进入左拐弯机动中的某一点。

可以看出,从视觉感受上,仿真结果和真实情况具有较好的一致性。

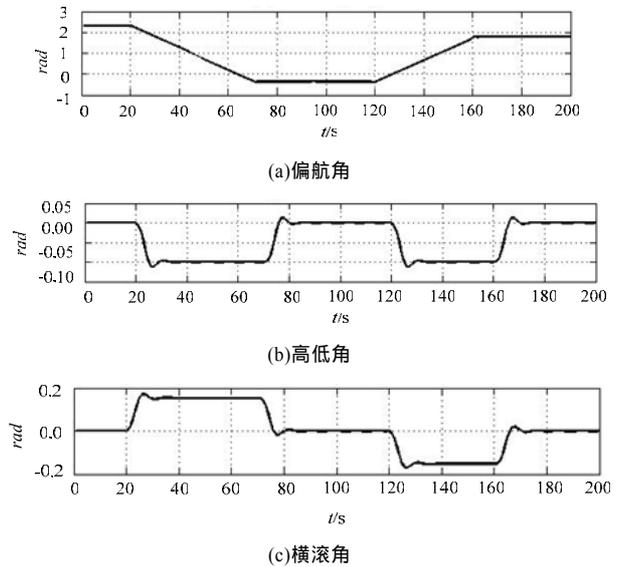


图 4 目标姿态角



图 5 仿真结果

#### 5 结束语

本文在研究虚拟场景构建等问题的基础上,用基于 VRML 的 Matlab 虚拟现实工具箱实现了机动目标 CCD 跟踪的视景仿真,仿真结果较好地再现了真实情况下的视觉效果。

同时,通过本文的仿真,笔者认为,为了达到更完善的、基于机动目标 CCD 跟踪的视景仿真效果,以下 2 个问题值得进一步的研究:

(下转第 250 页)