

基于 Unity3D 虚拟摄像系统中摄像机姿态定位^①

袁清普¹, 石广田², 赵 璐¹

¹(兰州交通大学 电子与信息工程学院, 兰州 730070)

²(兰州交通大学 机电工程学院, 兰州 730070)

摘要: 随着虚拟现实技术的蓬勃发展, 虚拟摄像系统的研究亦炙手可热. 虚拟摄像系统的使用, 使电视节目和电影电视特效等的制作踏上一个前所未有的新台阶. 然而在虚拟摄像系统的研究与使用过程中, 虚实摄像机运动姿态的同步一直是难以解决的关键问题, 即 MEMS 传感器在测量过程中会产生噪声影响, 其精确度有待提高. 为解决这一问题, 拟基于 Unity3D 平台, 使用 C# 脚本编程语言, 开发一套实时虚拟摄像系统, 着重研究虚实摄像机姿态同步融合技术, 选用卡尔曼滤波技术对传感器数据进行滤波处理, 减少噪声影响. 研究结果显示, 此虚拟摄像系统中虚实摄像机的运动姿态的同步运动能够满足电视目录制与电影电视特效制作等的需求.

关键词: Unity3D; 虚拟摄像; MEMS 传感器; 卡尔曼滤波; 运动姿态

Camera Posture Positioning of Virtual Camera System Based on Unity3D

YUAN Qing-Pu¹, SHI Guang-Tian², ZHAO Lu¹

¹(School of Electronics and Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China)

²(School of Mechatronic Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China)

Abstract: With the vigorous development of virtual reality technology, the research of virtual camera system has been placed more and more attention. Making television programs and film special effects embark on an unprecedented new level, because of using the virtual camera system. However, in the study and use of virtual camera system, the actual situation of the synchronization camera motion attitude is a key issue to resolve, namely the MEMS sensor will generate noise in the measurement process, and it is necessary to improve the accuracy. In order to resolve the problem, a real-time virtual camera system that based on Unity3D and using C#. The system focuses on the technology of synchronous fusion that based on the posture of entitative camera and virtual camera. In order to reduce noise, Kalman filtering technique is used to filter the sensor data. The result shows that the virtual camera's synchronized movement can meet the needs of TV recording and film special effects in this virtual camera system.

Key words: Unity3D; virtual camera; MEMS; Kalman filtering; posture positioning

现如今大多数电视台的电视目录制面均采用虚拟摄影棚技术实现虚实场景的结合, 不但能够降低制作成本, 而且能够达到真实、炫酷的效果. 摄像机姿态定位研究是虚拟摄像系统的研究重点之一, 通过对 MEMS 传感器采集的数据进行有效的降噪处理, 以满足虚拟摄像系统较高精度的需求. 通过研究表明, 随机噪声是影响 MEMS 传感器精度的主要因素, 对其进行模型辨识并滤波是提高传感器性能的主要途径^[1].

文献[2]对传感器的噪声误差采用小波算法分析, 对 MEMS 传感器的信号进行小波分解, 通过对粗尺度信号的重构, 可以从噪声中有效检测出 MEMS 传感器的误差特征. 文献[3]采用神经网络时间序列对传感器漂移进行建模和预测, 建立多样本输入单输出收敛模型, 能够实现较强实时性. 然而采用小波分析、神经网络等方法进行建模可以达到提高传感器性能的目的, 但这些方法得到的模型通常具有较高的阶数, 并

① 收稿时间:2015-11-03;收到修改稿时间:2015-12-21 [doi: 10.15888/j.cnki.csa.005240]

不十分适合于低成本系统的实时在线估计^[4,5]。

本文采用渐消因子自适应 Kalman 滤波算法对 MEMS 传感器采集的摄像机姿态数据进行降噪处理, 用 Matlab 进行仿真分析, 通过改变渐消因子大小, 使得摄像机姿态数据误差保持在可允许范围之内. 并结合 Unity3D 游戏引擎软件^[6]搭建虚拟三维拍摄场景, 实现虚拟摄像系统开发。

1 总体设计思路

由于考虑到开发条件的限制, 因此没有对系统的灯光、场景匹配等方面进行配置, 这样系统得以简化, 又未丢失主要功能。

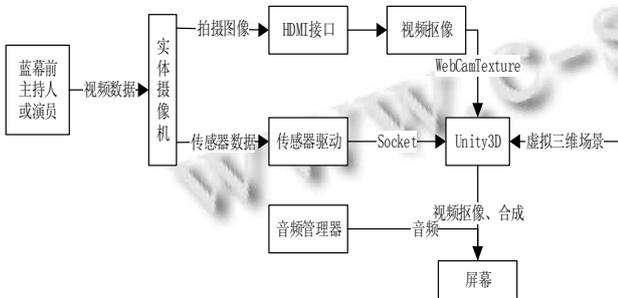


图 1 虚拟摄像系统工作流程图

系统主要针对摄像机姿态定位与蓝幕抠像两方面进行重点研究。

摄像机姿态定位: 实体摄像机由人员自由操控, 实体摄像机的运动旋转通过九轴传感器模拟信号模式实时传输到计算机, Unity3D 中的虚拟摄像机 (Main Camera) 接收到数据信息后, 经过数据去噪解算后会做与实体摄像机同步的运动旋转姿态。

视频蓝幕抠像: 由于只需要特定的真实任务或者物体出现在虚拟场景中进行效果合成, 所以拍摄背景

$$v[n] = \sum_{k=1}^n \frac{a[k] + a[k-1]}{2} \Delta t = v[0] + \frac{1}{2}(a[0] + a[n]) \cdot \Delta t + (a[1] + a[2] + \dots + a[n-1]) \cdot \Delta t \quad (5)$$

$$s[n] = \sum_{k=1}^n \frac{v[k] + v[k-1]}{2} \Delta t = \frac{1}{2}(v[0] + v[n]) \cdot \Delta t + (v[1] + v[2] + \dots + v[n-1]) \cdot \Delta t \quad (6)$$

此时, n 时刻物体的单轴移动距离如公式(7)所示。

$$v[n] = v[n-1] + \frac{a[n] + a[n-1]}{2} \cdot \Delta t \quad (7)$$

$$s[n] = s[n-1] + \frac{v[n] + v[n-1]}{2} \cdot \Delta t = s[n-1] + v[n-1] \cdot \Delta t + \frac{1}{4}(a[n-1] + a[n]) \cdot \Delta t^2 \quad (8)$$

由公式(7)、(8), 利用上一时刻的结果 $v[n-1]$ 、 $s[n-1]$ 和加速度 $a[n-1]$ 、 $a[n]$ 求得当前时刻的运

需要铺设蓝幕, 在 Unity3D 中调节抠像效果。

2 摄像机姿态定位

2.1 数据采集

摄像机的姿态定位是整个虚拟摄像系统研究的重点之一, 不但要把实体摄像机的位移旋转姿态传输到计算机, 而且必须保证运动姿态数据的精确性、实时性。

系统通过对 MEMS 九轴传感器中^[7]加速度计采集的数据进行二重积分, 进而求得位移. 假定 $a(t)$ 为加速度, 是加速度计的输出值, 随时间的变化而变化. 从 t_0 时刻开始数据采集, 从时刻 t_0 到时刻 t , 连续时间域中的位移 $s(t)$, 速度 $v(t)$ 和加速度 $a(t)$ 的计算关系如式(1)、(2)所示。

$$s(t) = \int_{t_0}^t v(t) dt + s(t_0) \quad (1)$$

$$v(t) = \int_{t_0}^t a(t) dt + v(t_0) \quad (2)$$

公式(1)、(2)中, $s(t_0)$ 表示 0 到 t_0 时刻的位移, $v(t_0)$ 为 t_0 时刻的瞬时速度。

将加速度计的输出值分解为若干个直角梯形, 初始 $s(t_0) = 0$, 那么当 $n > 1$ 时, 公式(1)可表示为公式(3)。

$$s(t) = \sum_{k=1}^n \frac{v(t_k) + v(t_{k-1})}{2} \Delta t \quad (3)$$

公式(3)在离散域中改为公式 4。

$$s[n] = \sum_{k=1}^n \frac{v[k] + v[k-1]}{2} \Delta t \quad (4)$$

动位移 $s[n]$ 和速度 $v[n]$.

由于传感器输出的是三维空间的加速度, 所以再求摄像机位移时, 加速度分别为 a_x 、 a_y 、 a_z , 利用

$$s[t] = \sqrt{(s_x[t] - s_x[t - \Delta t])^2 + (s_y[t] - s_y[t - \Delta t])^2 + (s_z[t] - s_z[t - \Delta t])^2} \quad (9)$$

2.2 误差分析

基于 Unity 3D 的虚拟摄像系统开发中, 为了确保虚实摄像机同步移动与旋转, 需要对传感器采集到的实体摄像机位移旋转姿态数据进行滤波去噪处理, 使得两者误差控制在允许范围之内.

MEMS 九轴传感器的数据在传输的过程中, 由于内外部噪声的干扰, 会使得传输的数据带有失真的状况^[8], 由于它在高频动态环境下由于整流机制会产生低频的非线性误差输出, 采用 Kalman 滤波方法进行误差补偿. 需要对测量信息进行 Kalman 滤波处理. Kalman 滤波方法具有较好的实时性, 是被广泛应用的机动目标跟踪算法

卡尔曼滤波(Kalman filtering)^[9]是一种最优化自回归数据处理算法, 利用线性系统状态方程, 通过观测系统输入输出的数据, 对系统状态进行最优估计. 由于系统的噪声和干扰对观测数据存在一定的影响, 所以最优估计也可以看作是滤波过程.

卡尔曼滤波的状态方程:

$$X = \Phi_{K,K-1} X_{k-1} + \Gamma_{k,k-1} W_{k-1} \quad (10)$$

观测方程:

$$Z_k = H_k X_k + V_k \quad (11)$$

其中, W_{k-1} 和 V_k 表示白噪声序列, Z_k 为观测值, $\Phi_{k,k-1}$ 、 $\Gamma_{k,k-1}$ 、 H_k 均为单位矩阵.

$$W_k = \Pi_{k,k-1} W_{k-1} + \xi_{k-1} \quad (12)$$

其中 W_k 为状态噪声, ξ_{k-1} 为零均值白噪声向量.

$$X_k^a = \begin{bmatrix} X_k \\ W_k \end{bmatrix} \quad (13)$$

扩增后:

$$\begin{bmatrix} X_k \\ W_k \end{bmatrix} = \begin{bmatrix} \Phi_{k,k-1} & \Gamma_{k,k-1} \\ 0 & \Pi_{k,k-1} \end{bmatrix} \begin{bmatrix} X_{k-1} \\ W_{k-1} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} \xi_{k-1} \quad (14)$$

$$Z_k = [H_k \ 0] \begin{bmatrix} X_k \\ W_k \end{bmatrix} + V_k \quad (15)$$

$\Pi_{k,k-1}$ 为传感器随机漂移模型参数.

加速度求出 v_x 、 v_y 、 v_z 之后就可以求得 s_x 、 s_y 、 s_z , 再由公式(9)求得位移.

但是一般来看, 系统的状态是一个时变系统, 这样卡尔曼滤波估计就失去了最优性, 跟踪精度下降, 收敛速度变慢, 甚至会发生丢失跟踪的问题. 通常采用自适应的滤波技术来改进上述问题^[10].

利用渐消因子自适应卡尔曼滤波, 方程描述如下:

$$P_{k,k-1} = \Phi_{K,K-1} P_{k-1} \Phi_{k,k-1}^T + \Gamma_{k,k-1} \alpha Q_{k-1} \Gamma_{k,k-1}^T \quad (16)$$

$$K_k = P_{k,k-1} H_k^T (H_k \alpha P_{k-1} H_k^T + \alpha R_k)^{-1} \quad (17)$$

$$P_k = \frac{1}{\alpha} [I - K_k H_k] P_{k,k-1} [I - K_k H_k]^T + K_k R_k H_k^T \quad (18)$$

$$\hat{X}_{k,k-1} = \Phi_{k,k-1} \hat{X}_{k-1} \quad (19)$$

$$\hat{X}_k = \hat{X}_{k,k-1} + \hat{X}_k [X_k - X_k \hat{X}_{k-1}] \quad (20)$$

Q_{k-1} 表示系统噪声方差阵, R_k 表示观测噪声方差阵, $P_{k,k-1}$ 表示一步预测误差方差阵, P_k 表示估计误差方差阵, K_k 表示滤波增益矩阵.

通过改变 α 的大小观察仿真结果选择 α 的值. 但是 α 的值具有很大的随意性, 对此对其进行改进, 应用到传感器噪声处理中.

2.3 卡尔曼滤波仿真分析

对传感器采集的姿态数据进行卡尔曼滤波之后, 噪声对数据所产生的影响明显降低, 姿态数据更具准确性, 使虚拟摄像机实时呈现出实体摄像机的运动姿态. 图 2、图 3 分别是在数据经过滤波前后在 Matlab 里面的仿真图.

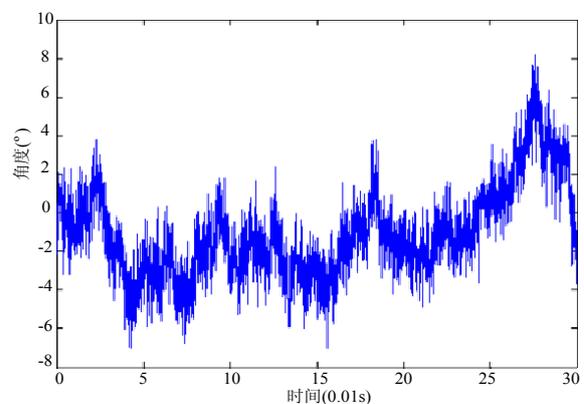


图 2 卡尔曼滤波前原始数据

由图 2 可以看出传感器采集的摄像机姿态数据存在比较大的误差, 这样在进行实际拍摄的时候会给工作人员增添许多额外的(如调试等)工作, 而且不能精准把握调试时的精确度. 图 3 是采集数据经过卡尔曼滤波之后的数据, 明显看出消除了大部分噪声误差, 使其保持在一个允许的范围之内.

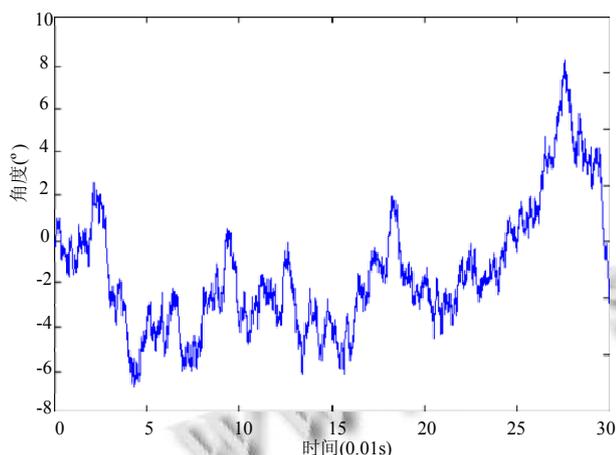


图 3 卡尔曼滤波后数据

虽然在进行滤波之后数据还会有部分噪声影响不能排除, 但此时的虚拟摄像机的运动姿态已经无限接近于真实摄像机运动姿态, 姿态数据误差在理想的范围之内, 对系统的影响可忽略不计.

3 与Unity 3D的结合

系统与 Unity3D 结合, 使得视频管理等操作更加简单流畅, 视频合成效果更符合需求, 虚实摄像机的姿态同步能够实现真实主持人或者表演者与虚拟场景进行交互. 文献[11]中采用的是 Ogre 引擎开发虚拟摄像系统, 能够实现三维场景效果, 并且对视频的质量、抠像等作了详尽的研究, 但是不能保证虚实摄像机的姿态同步, 交互性降低.

3.1 通信连接

Unity3D 中每一个场景只有一个 Main Camera 对整个场景的可视范围进行渲染, 相当于人的眼睛. 将采集到的实体摄像机的运动姿态数据绑定到 Main Camera 上, 就能准确呈现出摄像机位移、旋转、摇晃等姿态, 并且能够与实体摄像机进行同步.

在图 4 中, 表示的是将采集摄像机数据的 C#脚本 MEMSConnect.cs 添加到 Main Camera 上, 并将实体摄像机的姿态数据应用到 Main Camera 上,

MEMSConnect.cs 文件编写的是数据采集与解算的方法; Move Speed 表示摄像机移动的幅度, 一般情况设为 1, 移动幅度与实体摄像机保持一致.

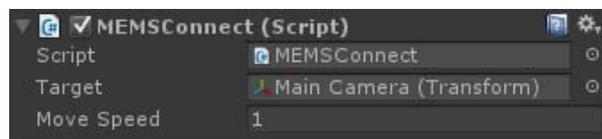


图 4 Unity 3D 中虚拟摄像机绑定采集数据脚本

Unity3D 与传感器的之间通过创建 Socket 实例, 以 TCP 协议通信, 获取传感器的姿态数据^[12].

```
Socket socket = new Socket(AddressFamily.
InterNetwork, SocketType.Stream, ProtocolType.Tcp);
EndPoint ep = new IPEndPoint(IPAddress.
Parse("10.4.18.150"), 7788);
socket.Connect(ep);
```

这样本地计算机(IP 地址为:10.4.18.150)就可以与传感器的驱动通过端口 7788 建立连接, 连接之后再创建一个线程调用 Read 方法来读取传感器的姿态数据.

```
Thread connectionThread = new Thread(new
ThreadStart(Read));
```

Read 方法循环检测是否与驱动进行连接, 如果正在进行连接, 则读取传感器数据.

3.2 数据绑定

Unity3D 平台中对对象旋转使用四元数来表示. 四元数是一种易于操作、计算量小、比较简单的姿态解算算法^[13], 一般适用于姿态解算.

传感器经过误差去噪之后的数据信息不能直接表达出摄像机的运动姿态, 需要对其进行数据转换, 成为四元数.

首先需要创建一个四元数 quaternion 作为数据容器接收传感器旋转姿态数据:

```
Quaternion quaternion = new Quaternion(0,0,0,1)
```

三维向量 moveDelta 作为数据容器接收传感器位移姿态数据:

```
Vector3 moveDelta =new Vector3(0,0,0)
```

最后, 绑定数据到 target 上. Target 在 Unity3D 中赋值为 Main Camera.

```
target.transform.position= target.transform.position
+ moveDelta;
```

target.transform.rotation= quaternion;

此时才能够保证虚拟摄像机与实体摄像机位移旋转姿态的同步,并且能够保持精确度的范围能够不影响节目效果的拍摄。

4 总结

基于 Unity3D 的虚拟摄像系统研究中采用 MEMS 九轴传感器采集实体摄像机的位移旋转姿态,并利用自适应卡尔曼滤波算法减少噪声影响,使得虚拟摄像机与实体摄像机能够保持同步的位移姿态,拍摄出虚实场景运动同步的视频效果。在数据与虚拟摄像机绑定采用四元数法表示摄像机的旋转,开发难度低,数据采集精度高,并且能够根据需要调节旋转幅度的大小。Unity3D 作为专业的虚拟现实、增强现实开发引擎软件,将会是虚拟摄像系统研究中不可或缺的开发环境。而基于 Unity3D 的虚拟摄像系统也将因为其开发成本低、操作简单等优点在未来电视节目、影视特效、动画制作等文化娱乐领域中发挥重要的作用。

参考文献

- 1 Ren YF, Ke XZ, Liu YJ. MEMS gyroscope performance estimate based on Allan. The Eighth International Conference on Electronic Measurement and Instruments (ICEMI 2008). Xi'an, China. 2008. 1326-1330.
- 2 刘宝生,闫莉萍,周东华.结合小波与增量符号相关的鲁棒图像匹配.光电工程,2005,32(9):59-62.
- 3 陈熙源,万德钧,程启明,余勇.陀螺随机漂移的神经网络预报方法研究.东南大学学报,1999,28(5):79-83.
- 4 袁广民,李晓莹,常洪龙,苑伟政.MEMS 陀螺随机误差补偿在提高姿态参照系统精度中的应用.西北工业大学学报,2008,26(6):777-781.
- 5 Chen XY. Modeling random gyro drift by time series neural Networks and by traditional method. IEEE Int Conf. Neural Networks & Signal Processing. Nanjing, China. 2003. 810-813.
- 6 Sung LK. Using Unity3D to facilitate mobile augmented reality game development. Internet of Things (WF-IoT), 2014 IEEE World Forum on Internet of Things. 2014. 21-26.
- 7 Wang YS, Li S. An implementation scheme of virtual camera system based on inertia sensing technology. Journal of Computer Engineering and Informatics, 2013, 1(1): 18-22.
- 8 郑健.基于 9 轴传感器的姿态参考系统研究与实现[硕士论文].成都:电子科技大学,2013.
- 9 敬喜.卡尔曼滤波器及其应用基础.北京:国防工业出版社,1973.
- 10 徐伟,李智,王勇军.北斗/GPS 无人飞行器动态导航中改进自适应算法.电子技术应用,2015,41(10):58-61.
- 11 王振雄.基于 Ogre 引擎的虚拟演播室系统的设计与实现[硕士论文].厦门:厦门大学,2014.
- 12 Li S, Wang YS, Xiang RP. Human body motion capture system based on inertia sensing technology. Advanced Materials Research, 2013, 722: 454-458.
- 13 孙丽,秦永元.捷联惯导系统姿态算法比较.中国惯性技术学报,2006,24(3):6-10.