

文章编号: 1001-5078 (2006) 04-0324-03

基于 VxWorks 的 SB/MHT 红外跟踪算法改进与实现

彭鼎祥, 汤心溢, 王世勇

(中国科学院上海技术物理研究所, 上海 200083)

摘要: 讨论了 SB/MHT 跟踪算法的改进和在 VxWorks 内嵌操作系统中的算法具体实现。对 SB/MHT 算法的门限关联、卡尔曼 (Kalman) 滤波预测和置信度轨迹评价这三个主要模块做了改进与简化, 使其在 VxWorks 平台下, 利用任务调度机制和中断管理机制, 实现多任务协作, 完成目标跟踪。SB/MHT 的 VxWorks 实现可以内嵌于 SOC 环境, 对红外跟踪系统的 SOC 实现有着重要意义。

关键词: 红外成像跟踪系统; 结构化分支多假设跟踪; 卡尔曼滤波

中图分类号: TP392.41 **文献标识码:** A

A Progressed MHT Infrared Tracking Algorithm Implemented with Great Efficiency on the VxWorks OS

PENG Ding-xiang, TANG Xin-yi, WANG Shi-yong

(Shanghai Institute of Technical Physics, Chinese Academy of Sciences, Shanghai 200083, China)

Abstract: In this paper, a progressed MHT infrared tracking algorithm is proposed in order to get great efficiency on the VxWorks Operation System. Through progressing target interfix module, Kalman filter and track evaluation module, SB/MHT implement target tracks on the VxWorks platform with wind task scheduling and interrupt service.

Key words: infrared search and tracking; SB/MHT; Kalman filter

1 引言

跟踪分析是红外成像跟踪系统的一个组成环节^[1], 建立在空间滤波、阈值分割和目标分割基础之上。空间滤波提高了图像信噪比; 阈值分割分离目标和背景; 目标分割提取潜目标向量数据; 跟踪分析应用跟踪算法, 根据目标运动特性找出其中真实目标点运动轨迹。

在IRST发展过程中, 跟踪算法很多。Donald B. Reid在1978年提出多假设跟踪算法MHT^[2], 根据潜目标的帧间匹配来构造假设轨迹, 并对轨迹进行评价。由于其计算量巨大, 休斯公司提出结构化分支多假设跟踪算法SB/MHT^[3], 结构化轨迹, 分别记录各条轨迹的状态属性, 最后进行轨迹评价和删除。

以往跟踪分析算法都是在PC机的Windows操作系统平台上实现的, Windows实时性不高; 同时PC机作为分立的系统存在与其他系统模块的接口协调问题。这里选择VxWorks内嵌操作系统作为算法实现的平台, 这是因为VxWorks可以很方便的嵌入到SOC平台, 为跟踪系统的SOC片上实现提供

了前提; 同时它具有性能优越的内核Wind和高效的多任务管理与调度机制^[4], 在Wind内核控制下, 可是实现任务的抢占式调度和轮转调度, 并配合中断管理机制完成多任务的协调运作。

本文对SB/MHT的轨迹预测、门限关联和轨迹评价这三个模块做了简化与改进, 并讨论了在VxWorks操作系统中算法的多任务高效实现的组织方式。

2 MHT算法流程

简化后的SB/MHT算法如图1。这里最主要的有三个部分: 卡尔曼滤波、门限关联和轨迹评价。卡尔曼滤波计算轨迹的状态和预测; 得到轨迹的预测位置后, 与潜目标数据进行数据门限关联; 门限关联后, 要对轨迹进行评价, 修改轨迹的各个属性, 基于置信度做出删除和确认。

作者简介: 彭鼎祥 (1977 -), 男, 博士生, 2000年毕业于厦门大学电子工程系。

收稿日期: 2005-07-01; 修订日期: 2005-10-20

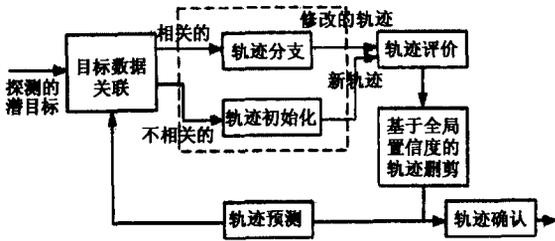


图 1 MHT算法流程
Fig 1 MHT flow chart

2.1 卡尔曼滤波预测

本算法认为目标作变速运动,其几何尺寸也在发生变化,所以需要有两组向量(状态向量和目标探测向量)来描述目标的状态,如式(1)及(2):

$$位置向量: X_1^T = [x, y, v_x, v_y, a_x, a_y], Y_1 = [x, y] \quad (1)$$

$$尺寸向量: X_2^T = [h, w, v_h, v_w, a_h, a_w], Y_2 = [h, w] \quad (2)$$

式中, [x, y] 表示空间位置; [v_x, v_y] 表示目标速度; [a_x, a_y] 表示目标加速度; [h, w] 表示空间尺度; [v_h, v_w] 表示尺度变化速度; [a_h, a_w] 表示尺度变化加速度。这两组向量拥有相同的转移矩阵和测量矩阵,分别是:

$$F = \begin{bmatrix} 1 & 0 & t & 0 & t^2/2 & 0 \\ 0 & 1 & 0 & t & 0 & t^2/2 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & t \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4)$$

在卡尔曼滤波轨迹预测中,这里做了不同以往算法的改进,不仅考虑到了目标的运动特性(位置向量),对目标的未来位置做了预测;也考虑到了目标运动中红外成像的几何特性(尺寸向量),对目标的未来几何尺寸做了预测。

2.2 轨迹与潜目标的数据门限关联

2.2.1 数据门限关联算法描述

假设已知的一组轨迹已经经过了 k 次的滤波。根据卡尔曼滤波的一般原理^[5,6]可知滤波器的残差:

$$Y(k) = Y(k) - HX(k/k-1) \quad (5)$$

残余协方差矩阵:

$$S = HP(k/k)H^T + R(k) \quad (6)$$

根据卡尔曼滤波模型,计算 k+1 时刻残余矢量 Y 和残余协方差矩阵 S,以及潜目标与真实目标之间的归一化距离 d²,也叫残余矢量的范数:

$$d^2 = YS^{-1}Y = c \quad (7)$$

在实际运算中的位置向量和尺寸向量所对应的残余矢量的范数分别为:

$$d_1^2 = \begin{bmatrix} x_{obs} - x_{pred} \\ y_{obs} - y_{pred} \end{bmatrix}^T S^{-1} \begin{bmatrix} x_{obs} - x_{pred} \\ y_{obs} - y_{pred} \end{bmatrix} = c_1 \quad (8)$$

$$d_2^2 = \begin{bmatrix} h_{obs} - h_{pred} \\ w_{obs} - w_{pred} \end{bmatrix}^T S^{-1} \begin{bmatrix} h_{obs} - h_{pred} \\ w_{obs} - w_{pred} \end{bmatrix} = c_2 \quad (9)$$

式中, obs 表示观测值; pred 表示预测值。在测量空间中,计算各轨迹残余矢量的范数,确定位置门限条件和尺寸门限条件,如果下一帧的潜目标符合这些门限条件,那么就可以认为这个新的测量点与该轨迹门限关联。

2.2.2 算法实现

在实际运算中,采用自适应数据关联门限的方法。由式(5)、(6)取得预测值与测量值之间的差距 c(n),根据 c(n)取三个门限,如图 2,来确定潜目标与相应轨迹的关联程度。



图 2 预测门限关联

Fig 2 forecast limen interfir

这三个门限的关系如式(10):

$$\begin{cases} c(n) = \sqrt{d^2} = d \\ C_1(n+1) = c(n) \times (+L_m) \\ C_2(n+1) = \times C_1(n+1) \\ C_3 = \times C_1(n+1) \end{cases} \quad (10)$$

式中, L_m 表示轨迹连续丢失的帧数; × 和 为经验参数,可以取 2、1.5、2。

用这种门限关联算法,门限将根据不同的情况适当的扩大或缩小,随时做出调整。这使得系统的跟踪有一定的自适应能力,克服了目标在某些帧中丢失的情况。同时既考虑了目标的位置门限,也考虑了目标的尺寸门限,实现了二维的门限关联,这大大保证了关联的准确性,也为后继处理减少了计算量。

2.2.3 轨迹门限关联处理

门限关联简单示意如图 3。轨迹跟踪过程中, k=1 轨迹得到正确的关联; k=2 时刻舍去了与预测尺寸不相关的点,实现正确关联; k=3 时刻没有潜目标与轨迹关联,这时算法默认该轨迹在这一时刻丢失探测目标。因此算法将做如下操作:

将轨迹 k=2 时刻的卡尔曼预测目标假设为 k=3 时刻的实际探测目标;

进行 k=3 时刻的卡尔曼运算,得到 k=4 时的预测目标,扩大 k=4 时刻关联门限 C;

如果在 k=4 时刻重新关联到目标,那么轨迹跟踪将按正常的运算继续;

如果在 k=4 时刻没有关联到目标,那么按 k

=3时刻做同样处理,置信度变低,最后可能被删除。

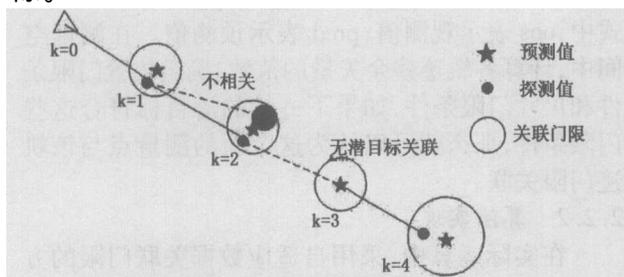


图 3 无目标关联轨迹处理

Fig 3 track management without interfix target

2.3 基于置信度的轨迹评价

置信度用来评价每一个轨迹代表一个真实目标的几率,进行轨迹的删除和确认操作。当一条新的轨迹生成时,轨迹的置信度初始化公式为:

$$L_{LR}(1) = \log \left[\frac{P_0(T)}{1 - P_0(T)} \right] = \log \left[\frac{N}{P_{FA}} \right] \quad (11)$$

$P_0(T)$ 为真目标在轨迹中的先验概率, N 是真目标点概率空间密度,它们有如下关系:

$$P_0(T) = \frac{N}{N + P_{FA}} \quad (12)$$

每经过一次成像,置信度增量计算公式为:

$$L_{LR}(k) = L_{LR}(k-1) + L_{LR}(k) \quad (13)$$

$$L_{LR}(k) = \begin{cases} \log \left(\frac{P_D e^{-\frac{d^2}{2}}}{2 P_{FA} \sqrt{|S|}} \right) \\ \frac{1 - P_D}{1 - P_{FA}} \end{cases} \quad (14)$$

P_D 表示系统的探测概率, P_{FA} 表示为潜目标数据的虚警概率。式(14)中,第1式和第2式分别表示轨迹有潜目标关联和没有潜目标关联时置信度增量的计算公式。

3 VxWorks下 MHT算法组织

VxWorks操作系统平台下 MHT算法组织如图4。MHT算法分为三个模块:目标数据关联、卡尔曼滤波预测和轨迹置信度评价。利用 VxWorks的多任务调度,以任务进程形式运行,相互之间通过队列来实现任务通信和数据共享传递。

分割)向 CPU 提交中断请求, VxWorks响应中断, ISR 释放信号量,发起新一轮的 MHT运算任务:“目标数据关联”获取硬件平台的图像预处理结果,关联运算后发送消息到操作队列,完成释放进程资源、在内存中共享结果数据和发起“卡尔曼滤波预测”运算三个操作;“卡尔曼滤波预测”运算后,以同样方式发起“轨迹置信度评价”的运算,最后提交确认的轨迹信息。

4 实验结果

实验对象是 256 x256 x8位的红外图像信号,由于 VxWorks具有良好的多任务调度机制,并建立在快速的硬件数据流基础之上,图像帧速可以达到 50帧/s,跟踪效果如图5所示,跟踪白框始终锁定跟踪目标。

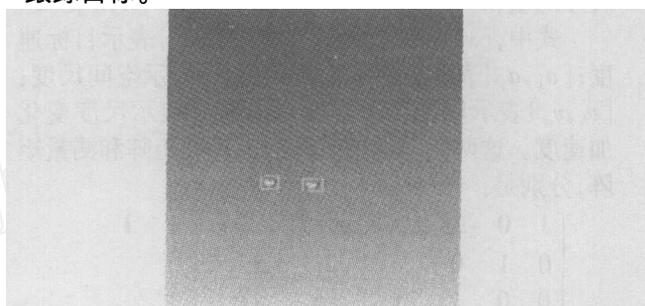


图 5 MHT实验结果

Fig 5 MHT result

5 小结

本文对 SB/MHT红外跟踪算法做了适当改进,并给出在 VxWorks操作系统下高效实现的具体方案。针对红外目标的特征,在对空间位置做出预测的基础上,增加了目标几何尺寸的预测,使得目标的跟踪预测更为精确,MHT的计算量更为减少。这种改进算法借助 VxWorks内嵌操作系统实现,使红外跟踪系统具有帧频快,实时性高,精确度高等特点。同时,SB/MHT改进算法的 VxWorks平台实现,对于红外跟踪系统的 SOC集成也具有重要的意义。

参考文献:

- [1] 崔春明,王天冠,周起勃. 用 COTS多处理机实现红外成像跟踪系统 [J]. 红外与毫米波学报, 2002, 21 (4): 261 - 265.
- [2] Donald B Reid An Algorithm for Tracking Multiple Targets [J]. IEEE Trans Automatic Control, 1979, AC24 (12).
- [3] J R Werthmann A step-by-step description of a computationally efficient version of multiple hypothesis tracking [A]. Proc SPIE, 1992, 1698: 288 - 300.
- [4] 陈智育,温彦军,陈琪. VxWorks程序开发实践 [M]. 北京:人民邮电出版社, 2004.
- [5] 史忠科. Kalman滤波新结构及其在目标跟踪中的应用 [J]. 自动化学报, 1994, 20 (5): 605 - 609.
- [6] 王瑞平,陈序三. Kalman滤波的实现和自适应处理 [J]. 核电子学与探测技术, 2001, 21 (6): 247 - 2501.

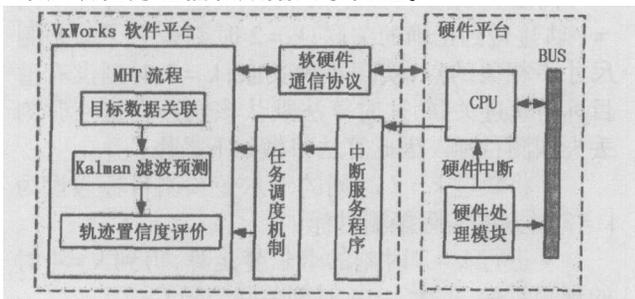


图 4 VxWorks下 MHT算法实现

Fig 4 MHT implementing on VxWorks OS

VxWorks下软件操作流程:硬件处理模块(目标