



一种基于双缓存的高速测控目标视图控制算法

刘宏建, 王明孝, 吴超辉, 蔡中祥

引用本文:

刘宏建, 王明孝, 吴超辉, 等. 一种基于双缓存的高速测控目标视图控制算法[J]. [兵器装备工程学报](#), 2020, 41(10): 192-196.

您可能感兴趣的其他文章

1. [基于粒子滤波的非线性退化设备剩余寿命自适应预测](#)

引用本文: 王玺, 周薇, 胡昌华, 等. 基于粒子滤波的非线性退化设备剩余寿命自适应预测[J]. [兵器装备工程学报](#), 2020, 41(10): 41-47+57.

2. [一种提高处理器运算能力的控制方法研究](#)

引用本文: 江帆, 张晨洁. 一种提高处理器运算能力的控制方法研究[J]. [兵器装备工程学报](#), 2020, 41(10): 239-243+254.

3. [SylixOS平台下的火控实时解算与实现](#)

引用本文: 程文博, 屈艺, 吴盘龙, 等. SylixOS平台下的火控实时解算与实现[J]. [兵器装备工程学报](#), 2020, 41(10): 29-34.

4. [基于无人机主动导航的电力线检测与定位](#)

引用本文: 吕晓丽, 姜航, 李文辉. 基于无人机主动导航的电力线检测与定位[J]. [兵器装备工程学报](#), 2020, 41(10): 210-214.

5. [固定翼无人机纵向控制回路多工况状态预测](#)

引用本文: 梁少军, 郑幸, 谢礼鹏, 等. 固定翼无人机纵向控制回路多工况状态预测[J]. [兵器装备工程学报](#), 2020, 41(10): 203-209.

一种基于双缓存的高速测控目标视图控制算法

刘宏建^{1,2}, 王明孝², 吴超辉², 蔡中祥¹

(1. 信息工程大学 地理空间信息学院, 郑州 400052; 2. 32178 部队科创中心, 北京 100020)

摘要: 高速测控目标视图控制是靶场试验指控显示系统的重要内容, 围绕靶场试验目标视图自适应控制问题, 讨论了传统视图控制算法的实现思路, 分析了传统算法存在的问题与瓶颈, 提出了地图缓存前后台两级切换的模型, 在此基础上对试验目标视图控制算法进行了优化, 采取空间换时间的策略, 消除了算法效率同比例尺与目标数量的相关性, 为试验目标自适应控制提供了一种新思路。

关键词: 双缓存; 测控; 视图自适应控制; 动态显示

本文引用格式: 刘宏建, 王明孝, 吴超辉, 等. 一种基于双缓存的高速测控目标视图控制算法[J]. 兵器装备工程学报, 2020, 41(10): 192-196.

Citation format: LIU Hongjian, WANG Mingxiao, WU Chaohui, et al. Adaptive Control Algorithm of High Speed Measurement and Control Target View Based on Double Cache[J]. Journal of Ordnance Equipment Engineering, 2020, 41(10): 192-196.

中图分类号: P231

文献标识码: A

文章编号: 2096-2304(2020)10-0192-05

Adaptive Control Algorithm of High Speed Measurement and Control Target View Based on Double Cache

LIU Hongjian^{1,2}, WANG Mingxiao², WU Chaohui², CAI Zhongxiang¹

(1. Institute of Surveying and Mapping, Information Engineering University, Zhengzhou 450052, China;

2. Science and Technology Innovation Center of 32178 Army, Beijing 100020, China)

Abstract: High speed measurement and control target view control is an important part of the command and control display system in weapon test base. This paper discussed the realization of the traditional view control algorithm, analyzed the problems and bottlenecks of the traditional algorithm, put forward the model of two-level switch between the front and back of map cache, and on this basis, carried out the test target view control algorithm. The optimization strategy of space for time was adopted to eliminate the correlation between the algorithm efficiency and the scale and the number of targets, which provides a new idea for the adaptive control of experimental targets.

Key words: double buffer; measurement and control; view adaptive control; dynamic display

常规武器装备靶场试验中, 无人机、直升机、导弹等由地面测控网跟踪、测量和控制的飞行目标称为高速测控目标^[1]。有别于常规运动目标, 高速测控目标的运动速度快, 地面测控网测量获得的武器目标实时数据帧率高, 单个试验

目标帧率可高达 20 帧/秒^[2-3]。高速测控目标视图控制是指: 在靶场指控显示系统中, 通过对目标在地图活动视图进行实时控制, 使测控试验目标在场景中获得最佳显示效果^[1]。

收稿日期: 2019-11-25; 修回日期: 2019-12-26

基金项目: 国家自然科学基金面上项目(41471337); XX 部十二五军事测绘重点项目

作者简介: 刘宏建(1980—), 男, 战略支援部队信息工程大学博士, 32178 部队科技创新中心博士后, 主要从事靶场地理空间分析与应用建模研究, E-mail: chxy_lhj@126.com。

高速测控目标视图自适应控制是靶场试验指控显示系统的核心功能,目前国内外对目标视图控制算法的研究归纳起来可总结如下:1)大部分应用基于不同的空间索引技术(四叉树、BSP树、KDB树、CELL树等),实现了单目标视图控制算法,但较少涉及多目标全局视图控制^[4-7]。2)有少部分应用实现了单目标和多目标的视图自适应控制,但在实现时多采用单级地图缓存和动态目标显示技术相结合的方法^[8],这种方法在北斗、GPS动态监控导航应用中效果良好,但难以支持强实时性的高速测控目标视图控制应用。因此,本文提出并实现了一种基于双缓存模型的高速测控目标视图自适应控制算法,较好地解决了上述方法的种种弊端,在实际应用中具有较大意义。

1 视图自适应控制算法

高速测控目标视图实时控制包括两种情况:一是焦点目标跟踪,即当试验目标接近当前地图活动视图地理范围的边界时,根据目标的运动趋势,进行地图数据的预加载,当运动目标达到活动视图边界时,将目标切换至视图中央位置;二是多目标全局视图,即同时有多个被试或参试目标时,既要保证所有目标在当前的地图视图内,又不能使其过于拥挤,需自适应调整地图显示比例尺,从而使多目标显控达到最佳效果。

从靶场试验指控显示的具体需求来看,焦点目标跟踪和多目标全局视图是两个不同的概念,操作方式也不相同,但在算法实现的过程中,可以将焦点目标跟踪和多目标全局视图算法进行融合处理,即采用统一的地图视图自适应算法进行一致处理。其原因如下:1)从本质上来讲,焦点目标跟踪是多目标全局视图的特殊形式。当多目标全局视图模式中的目标数量 $n=1$ 时,多目标全局视图即退化成为焦点目标跟踪的情况。2)不管是焦点目标跟踪还是多目标全局视图,其核心都是对地图数据的快速调度算法,即通过快速的基础地理数据调度与刷新显示来进行视图的动态切换。

高速测控目标视图自适应控制算法的主要思路是:首先从靶场试验测控网解析实时目标信息,并判断当前数据帧是否合法,然后通过目标轨迹判断目标运动趋势,确定目标活动视图是否需要调整,最后预加载地图数据,当目标达到视图阈值参数时,切换当前视图窗口并刷新地图。

本文以AE(ESRI ArcGIS Engine)为试验开发平台,在Intel Core i7 CPU,主频2.70GHz,内存16GB,Win7操作系统环境下,对上述算法进行了统计,结果如表1、表2所示。

根据表1、表2的统计数据,可发现两种现象:1)在焦点目标跟踪的情况下,初始的地图活动视图比例尺大小是影响算法性能的一个重要因素,当初始地图活动比例尺大于1:50 000时,对内存和CPU资源的消耗迅速增大,逐渐累积至地图无法再进行刷新,同时导致界面无任何响应。当初始地图活动比例尺小于1:250 000时,算法运行过程中地图刷新不受影响。这是由于比例尺的减小使当前地图活动视图的地

理范围变大,进而使地图刷新的频率大大下降,但这造成的结果就是使目标显示时无法观察细部的地形地物。2)在多目标全局视图的情况下,如果参试目标数量 $n \geq 3$,视图自适应控制算法运行3~4 min就会崩溃,而且随着参试目标的增加,算法可正常执行的时间会越来越短。

表1 焦点目标跟踪运行统计表

初始视图 地图比例 尺大小	$t=0$ 时刻		$t=5$ min		运行结果 描述
	PF	CPU	PF	CPU	
	使用 率/M	使用 率/%	使用 率/M	使用 率/%	
1:50 000	189	7	2.67	100	地图无法继续刷新,界面操作无响应
1:100 000	192	9	388	31	地图刷新正常,界面操作响应有迟滞现象
1:250 000	187	11	347	26	地图刷新正常,界面操作流畅

表2 多目标全局视图运行统计表

参试目 标数量 n	$t=0$ 时刻		$t=7$ min		运行结果 描述
	PF	CPU	PF	CPU	
	使用 率/M	使用 率/%	使用 率/M	使用 率/%	
$n=2$	213	11	2.71	100	地图无法继续刷新
$n=3$	221	14	2.73	100	地图无法继续刷新
$n=5$	223	17	/	/	运行244 s,出现内存异常,算法崩溃
$n=8$	227	18	/	/	运行197 s,出现内存异常,算法崩溃

产生这两种现象的根本原因在于地图的调度与刷新机制,AE的地图刷新机制如下:首先获取屏幕区域的地理范围,利用空间索引从地理数据库中快速获取当前屏幕范围内的地理实体要素,再将地理实体要素的坐标转换为屏幕坐标,然后用当前屏幕窗口对地理实体数据进行剪裁,剔除不需要绘制的地理实体,并将经过剪裁的地物直接绘制至当前屏幕。这种地图调度与刷新机制,在实时性要求不是很高的情况下是可行的,但对于高速测控目标视图自适应控制,目标数据帧率高、运动速度快、视图切换非常频繁,就导致了地图数据的调度和刷新成为算法的瓶颈。

综上所述,AE的地图调度与刷新机制已不适应于高速测控目标的视图控制要求,为解决上述问题,本文提出并实现了基于双缓存技术的目标视图自适应控制算法。

2 双缓存模型

显示缓存技术的实质是将大量数据的渲染从 CPU 端转移到显卡上,以充分利用显卡的性能,减少 CPU 处理负荷,使得大量的动态数据以及高频率刷新的操作可以在瞬间完成^[2]。本文将显示缓存分为地图显示缓存 M 和虚拟显示缓存 D_1 、 D_2 。地图显示缓存 M 并不是一个实际的存储区域,而是一个屏幕显示接口指针,其作用是在缓存之间进行切换。 D_1 、 D_2 是两个和活动视图屏幕显示范围相同的显示缓存空间,如图 1 所示。

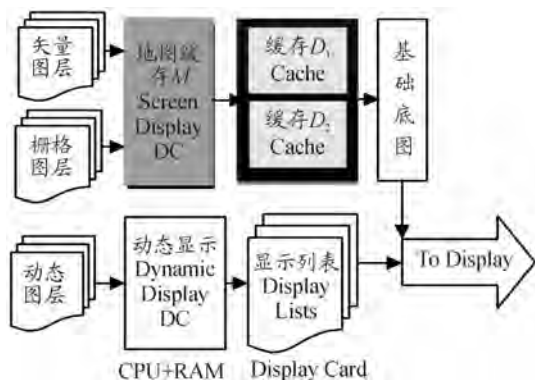


图 1 双缓存模型结构框图

$TargetsExtent$ 和 $TargetsExtent'$ 为范围小于虚拟屏幕的一个内边界。以焦点目标跟踪为例,假设地图显存 M 指向虚拟显存 D_1 ,则当试验目标移动到 $TargetsExtent$ 边界时,通过遍历试验目标图形缓存容器 Graphics Containter 中的轨迹数据,计算其相对于目标视图边界的运动趋势,如果目标仍然向视图边界继续运动,则通过 $IScreenDispaly$ 接口开始重新组织地理数据,并将其放入虚拟显存 D_2 ;同时,在缓存 D_1 上,试验目标的移动仍然在进行,当试验目标移动到地图活动视图范围 $GeoExtent_0$ 时,虚拟显存 D_2 上新的地理数据组织已经准备完毕,此时将地图显存 M 指向虚拟显存 D_2 ,就可以实现地图视图的快速切换(见图 2)。对于多目标全局视图,其过程类似,但在判断目标运动趋势时,需要考虑两个边界,即目标边界 $TargetsExtent$ 和目标视图最小边界 $TargetsExtent'$ 。

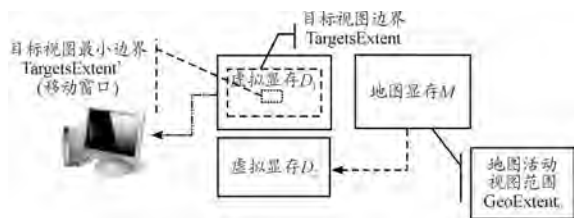


图 2 缓存切换与调度过程框图

3 算法优化

基于双缓存的视图自适应控制算法实现流程如图 3 所示。

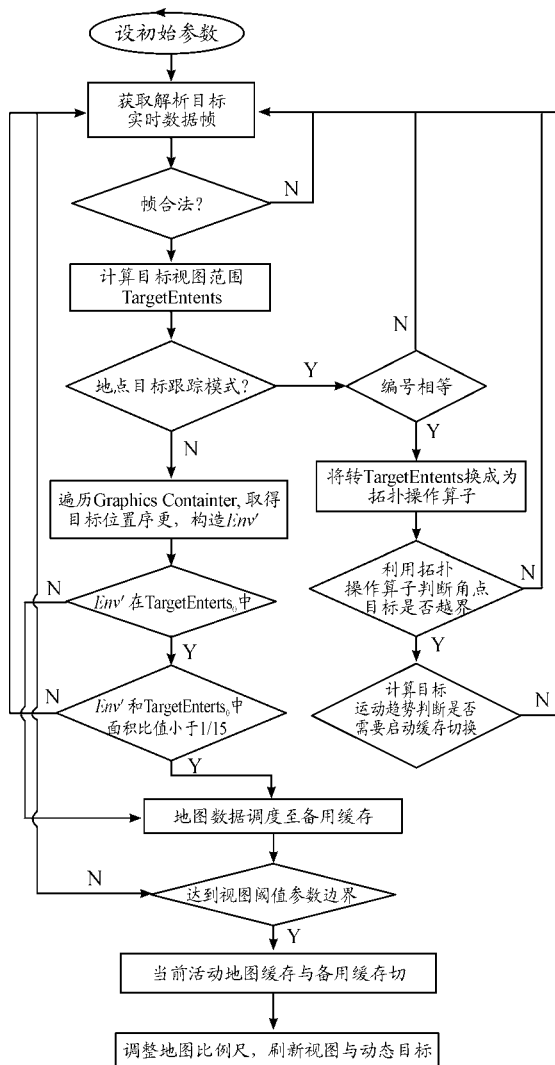


图 3 目标视图自适应控制算法流程图

步骤 1 利用 UDP 组播协议从靶场试验测控通信网获取目标实时数据帧,解析目标当前数据帧,获取试验目标的目标编号、经纬度坐标、靶场坐标、姿态角、数据时间等信息;判断实时数据帧是否合法,并进行野值剔除与数据平滑(野值剔除是通过针对不同数据段设定不同的“振荡阈值”,去除随机测量误差和野值点对算法的影响^[2];平滑处理是利用正交函数系最小二乘拟合的方法^[3]对实时数据帧进行平滑,野值剔除与数据平滑的目的是为了提高算法鲁棒性)。

步骤 2 计算目标视图范围 $TargetsExtent$,方法如下:

① 获取地图活动视图范围 $GeoExtent_0$,得到四个角点 $\{G_1, G_2, G_3, G_4\} = \{(b_1, l_1), (b_2, l_2), (b_3, l_3), (b_4, l_4)\}$,均为地理坐标系中的坐标,需将其转换为屏幕坐标系下的像素坐标,设其为 $\{S_1, S_2, S_3, S_4\} = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)\}$ 。

(x_4, y_4) }, 设目标视图范围 TargetsExtent 在屏幕上的像素宽度和高度为 w_s, h_s , 则:

$$\begin{aligned} w_s &= \frac{|x_2 - x_1| - 2 * dw}{2} \\ h_s &= \frac{|y_2 - y_1| - 2 * dh}{2} \end{aligned} \quad (1)$$

式(1)中的 d_w 和 d_h 是指目标地图符号的宽度和高度, 由于目标符号占据有一定大小的屏幕区域, GeoExtent₀ 的范围要大于 TargetsExtent 的范围。在多目标情况下, d_w 和 d_h 取所有试验目标地图符号宽度和高度的最大值。

③ 利用 w_s 和 h_s 以及地图活动视图范围的中心点求解 TargetsExtent 在屏幕坐标系中的角点坐标 $\{S'_1, S'_2, S'_3, S'_4\} = \{(x'_1, y'_1), (x'_2, y'_2), (x'_3, y'_3), (x'_4, y'_4)\}$ 。

④ 将 $\{S'_1, S'_2, S'_3, S'_4\} = \{(x'_1, y'_1), (x'_2, y'_2), (x'_3, y'_3), (x'_4, y'_4)\}$ 转换到地理坐标系中, 即可得到四个角点 $\{T_1, T_2, T_3, T_4\}$ 的地理坐标, 这样就可计算 TargetsExtent 的范围。

步骤3 通过接口查询获取当前 ActiveMapView 的地图显示缓存控制接口和屏幕缓存控制接口初始化 D_1, D_2 , 为 D_1, D_2 分配存储存储空间, 并通过地图显示缓存控制接口, 将当前活动地图窗口所有地理实体要素像素化, 存入 D_1 。

步骤4 判断当前模式是焦点目标跟踪或是多目标全局视图, 如果是焦点目标跟踪模式, 转步骤5处理; 如果是多目标全局视图模式, 则转步骤8处理; 如果未启动地图自适应控制, 则转步骤1。

步骤5 将当前目标数据帧中的目标编号和焦点目标编号进行比对, 如果当前目标是焦点目标, 则遍历试验目标图形缓存容器 (Graphics Containter), 获取并更新目标运动的位置数据和轨迹数据。

步骤6 通过接口查询, 获取当前活动视图范围的空间

拓扑操作算子 (ITopyOperator 接口对象), 利用空间拓扑操作算子判断试验目标是否在目标视图的地理范围区域内。如果目标在 TargetsExtent 之内, 转步骤1; 否则, 计算焦点目标轨迹的切线 (偏航角), 从而判断试验目标的运动趋势, 如果试验目标继续向 TargetsExtent 之外移动, 则转步骤7。

步骤7 利用 GeoExtent₀ 的宽度、高度以及当前目标位置重新构造地图视图活动范围, 并转步骤11;

步骤8 设武器装备试验目标数量为 n , 遍历试验目标图形缓存容器, 获取所有目标的当前位置信息序列 $\{P_0, P_1, P_2, \dots, P_n\}$, 利用点集构造多边形 Polygon, 并计算其外接矩形 Enw' 。获取 n 个试验目标的运动轨迹曲线, 同时计算所有试验目标偏航角序列 $\{\varphi_1, \varphi_2, \dots, \varphi_n\}$ 。

步骤9 获取 Enw' 的空间拓扑操作算子, 判断 Enw' 同目标视图范围 TargetsExtent 的关系, 循环遍历 n 个试验目标, 判断是否需要虚拟显存 D_1 和 D_2 的切换, 方法如下:

① $Enw' \cap TargetsExtent$, 这种情况下, 需判断 Enw' 与 TargetsExtent 的面积比值, 本文从多次试验中总结出了比例阈值为 $1/15$, 即外接矩形大小占据目标视图范围的比例小于 $1/15$ 时, 则转步骤10, 启动数据预加载;

② $Enw' \subset TargetsExtent$, 这种情况说明至少有一个目标飞出了当前视图阈值边界, 继续根据偏航角 φ 判断目标运动趋势, 进而确定是否需要进行地图数据组织和显示缓存切换。

步骤10 启动后台线程, 利用 IScreenDisplay 接口的地图数据调度方法开始组织新的地理数据, 并将其拷贝至 D_2 ;

步骤11 变换地图缓 M 存指向的显示缓存对象由 D_2 指向 D_1 , 自动调整地图比例尺, 算法结束。

表3及表4是优化后视图自适应控制算法的运行统计数据。

表3 优化后焦点目标跟踪运行统计数据

初始视图地图 比例尺大小	$t=0$ 时刻		$t=35$ min		$t=240$ min		运行结果描述
	PF 使用 率/M	CPU 使用 率/%	PF 使用 率/M	CPU 使用 率/%	PF 使用 率/M	CPU 使用 率/%	
1: 50 000	231	13	254	13	251	18	地图刷新流畅, 算法运行正常
1: 100 000	241	19	254	20	256	17	地图刷新流畅, 算法运行正常
1: 250 000	239	17	252	17	256	22	地图刷新流畅, 算法运行正常

表4. 优化后多目标全局视图运行统计数据

参试目标 数量 n	$t=0$ 时刻		$t=7$ min		$t=240$ min		运行结果描述
	PF 使用 率/M	CPU 使用 率/%	PF 使用 率/M	CPU 使用 率/%	PF 使用 率/M	CPU 使用 率/%	
$n=2$	237	16	286	17	282	18	地图刷新流畅, 算法运行正常
$n=3$	243	19	294	21	301	20	地图刷新流畅, 算法运行正常
$n=5$	247	22	294	23	307	20	地图刷新流畅, 算法运行正常
$n=8$	247	23	301	23	301	24	地图刷新流畅, 算法运行正常

4 结论

1) 在焦点目标跟踪的情况下,经过双缓存算法优化,初始地图活动视图比例尺同算法性能没有明显相关性,其对资源的累积消耗现象也得到了消除。

2) 在多目标全局视图的情况下,参试目标数量 n 的大小同算法性能也无明显相关性,不会随着目标的增多使算法变得不稳定(相同硬件条件下,系统最大可承载参试目标数量 $n \geq 8$)。

3) 经过双缓存技术的优化,目标视图自适应控制算法在资源的初始消耗上有一定增加,这是用内存资源换取地图刷新时间的结果,这个代价是完全可以接受的。

4) 经优化后,目标视图自适应控制算法鲁棒性大大增强,8个目标情况下,连续运行 240 min,共显示 2 304 000 个轨迹点,PF 使用率从 282 M 增至 301 M,增速平缓,整个过程中 CPU 使用率一直在 20% 上下浮动。

参考文献:

- [1] 常显奇,程永生. 常规武器装备试验学[M]. 北京:国防工业出版社,2007.
- [2] 刘宏建,刘建忠,蔡中祥. 应用缓存分流模型的高速测控目标全轨迹显示算法[J]. 测绘科学,2013,38(5):94-96.
- [3] 刘宏建,蔡中祥,刘子义. 基于正交函数系最小二乘拟合

的姿态平滑方法[J]. 兵器装备工程学报,2017,38(01):21-23.

- [4] 王德泉. VxWorks 下栅格地图显示解决方案[J]. 测绘通报,2011(1):31-34.
- [5] 焦景欣,代亮,胡含凯,等. 采用双缓存的视景仿真图像显示效果改善方法[J]. 探测与控制学报,2010,32(5):92-95.
- [6] 黄科佳,李少杰,左尧,等. 基于四叉树索引和双缓存机制的 GIS 动态可视化模型设计与实现[J]. 测绘与空间地理信息,2018,41(4):102-105.
- [7] WANG X, PULLAR D. Describing dynamic modeling for landscapes with vector map algebra in GIS[J]. Computers & Geosciences,2005,31(8):956-967.
- [8] 王柯,王腾,张文诗,等. 移动端 GIS 地图服务模式研究与实践[J]. 测绘科学技术学报,2014(4):368-371.
- [9] 胡泽明,岳春生,王志刚. 嵌入式 GIS 系统实时响应的软件方法实现[J]. 测绘科学,2007,32(1):98-99.
- [10] 黄雁,冯艳杰,孟庆祥. 嵌入式 GIS 中矢量数据的快速显示方法研究[J]. 城市勘测,2012(1):20-23.
- [11] ZHANG D, WANG L, HUI H C, LI H Y. Navigation map Data Representation and Parallel Display Algorithm in an Embedded Environment [J]. Geospatial Information Science,2008,11(1):62-65.

科学编辑 李臣(国家天文台高级工程师)
责任编辑 唐定国