

# 本地滞后对协同增强现实系统性能影响研究\*

陈 岭, 徐永健, 赵刚强, 陈根才  
(浙江大学计算机学院, 浙江 杭州 310027)

**摘 要:** 一致性控制是协同增强现实研究核心问题, 本地滞后算法可有效消除协同过程中的短期不一致, 但如何选择合适的本地滞后值是该算法的关键问题。文章构建了一个具体的协同增强现实系统, 并在不同本地滞后值下对该系统性能进行研究。实验结果表明本地滞后值(0~600 ms)对系统性能的影响是非线性的, 这些影响可分为三段: 0~100, 100~500, 500 ms以上, 当本地滞后值小于100 ms时, 本地滞后对系统性能没有影响, 当本地滞后值从100 ms增加到500 ms时, 本地滞后对系统的性能影响随滞后值变大而变大, 但被试均能完成任务, 当本地滞后值超过500 ms时, 系统性能急剧变坏, 一些被试不能顺利完成任务。

**关键词:** 增强现实; 协同增强现实; 一致性控制; 头盔显示器; 传感器

**中图分类号:** TP319 **文献标识码:** A **文章编号:** 0529-6579(2009)01-0123-05

## Effects of Local-Lag Mechanism on Task Performance in a Collaborative Augmented Reality System

CHEN Ling, XU Yongjian, ZHAO Gangqiang, CHEN Gencai  
(College of Computer Science, Zhejiang University, Hangzhou 310027, China)

**Abstract:** Consistency maintenance is a kernel problem to Collaborative Augmented Reality (CAR) research. In this area, local-lag mechanism has been proposed to eliminate short-term inconsistency, while how to select a proper lag value is a key problem to the mechanism. In this paper, a particular CAR system is presented, and an experiment was conducted to investigate the effects of lag value on the task performance of the system. Experimental results indicate that the effect of lag value on task performance is not linear, and it could be separated into three segments by two dividing points: 100 ms and 500 ms. When the lag is less than 100 ms, the effect of lag is negligible. When the lag increases from 100 ms to 500 ms, the negative effect of lag increases with the lag, but all participants can accomplish the task. When the lag exceeds 500 ms, the task performance decreases rapidly, and some participants even cannot accomplish the task.

**Key words:** augmented reality; collaborative augmented reality; head mounted display; sensor

增强现实(Augmented Reality, AR)<sup>[1]</sup>, 是虚拟现实(Virtual Reality, VR)技术的一个分支。AR技术可将计算机生成的虚拟物体或其它信息叠加到真实场景中, 从而实现虚实结合的再现结果, 在工业、医学、设计等方面得到广泛应用<sup>[2-4]</sup>。协同增强现实(Collaborative Augmented Reality,

CAR)<sup>[5]</sup>是对单用户AR技术的扩展, 使多个用户能共享一个增强现实空间。CAR技术具有AR技术的特点, 且支持多用户在共享空间上进行协作, 在游戏、教学等领域有着广阔的应用前景<sup>[6-7]</sup>。

现有CAR研究大都以共享同一真实环境为前提, 所以传统CAR系统无法支持异地协同工作,

\* 收稿日期: 2008-09-18

基金项目: 国家自然科学基金资助项目(60703040); 浙江省自然科学基金资助项目(Y107178)

作者简介: 陈岭(1977年生), 男, 副教授; E-mail: lingchen@cs.zju.edu.cn

使其应用受到极大限制。为使 CAR 系统支持分布式异地协同工作,需对以下问题进行深入研究:①如何在虚实结合的环境中对系统一致性进行控制;②在分布式环境下各用户物理环境不尽相同,如何减少这种环境不一致对协作任务性能的影响;③在分布式环境下各节点传感和显示设备存在异构性,如何在异构环境下实现系统内虚拟物体的注册和跟踪。

本文给出一个分布式 CAR 系统的设计与实现,具体介绍其采用的一致性控制、注册和跟踪等关键技术。此外,通过构建一个具体的协作任务,文章重点研究了一致性控制具体参数设置对系统协作性能的影响,为以后此类系统的设计与开发提供指导。

## 1 分布式协同增强现实系统

首先介绍系统使用的主要设备,然后给出系统结构图和流程,最后给出注册、跟踪以及一致性控制实现方法。

### 1.1 系统主要设备与结构

CAR 是 AR 技术的拓展,它们之间存在着很多共性,如均含有头盔显示器、用于跟踪用户头部方位和物体操作的跟踪设备。此外,由于分布式 CAR 系统用户不共享同一物理环境,各节点间还需通过网络设备进行互联。

现有 AR 系统主要使用头盔显示器 (Head Mounted Display, HMD) 实现具体场景的再现。HMD 根据实现原理不同被分为两种:视频透视式头盔显示器 (Video see-through HMD) 和光学透视式头盔显示器 (Optical see-through HMD)。两种系统各有千秋,本系统在实现过程中采用视频透视式头盔显示器。此外,系统采用磁传感器对用户头部方位以及用户对物体的具体操作进行跟踪,以便于往视频流中准确添加三维虚拟物体信息。

整个系统的结构如图 1 所示,主要由以下三部分组成:捆绑了摄像头和传感器的视频透视式头盔显示器;与传感器捆绑在一起的实际操作物体;用于处理数据的 PC。用户在使用该系统时需头戴捆绑了摄像头和传感器的视频透视式头盔显示器,手拿捆绑了传感器的实际操作物体,节点间通过网络传输被操作物体的状态更新信息。

系统流程主要由以下三部分组成:初始化、获取视频、主循环。初始化完成视频参数、传感器参数、网络传输参数等设置,并对系统进行初始化,启动传感器和摄像头。获取视频负责从摄像头获取

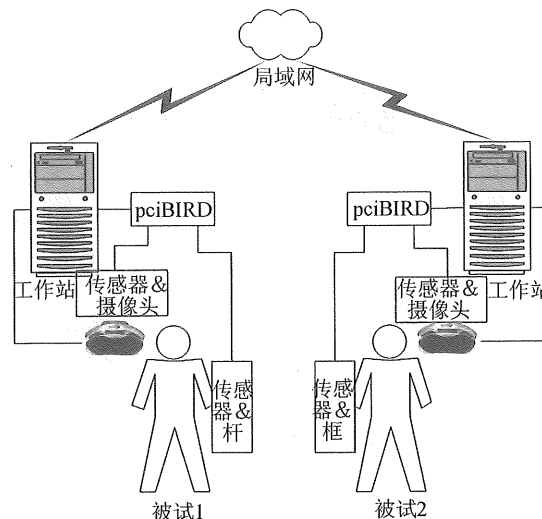


图 1 系统结构

Fig. 1 System architecture

视频图像。主循环主要完成以下四部分工作:①注册跟踪,通过传感器获得头部姿态以及对物体的操作信息;②网络传输及一致性控制,接收协作方发送的物体操作信息,发送给对方本地的物体操作信息,并维护整个系统的一致性;③碰撞检测,判断系统内的物体是否发生碰撞;④虚拟物体绘制,结合传感结果以及碰撞检测结果,将虚拟物体叠加到获取的视频图像上并通过头盔显示器呈现给用户。

### 1.2 注册与跟踪

系统使用传感器跟踪用户头部姿态以及对物体的操作,因此存在三个相关的坐标系:真实坐标系;传感器坐标系;视频坐标系。传感器得到的是传感器在发射器坐标系(真实环境坐标系)中的位置信息,它们不能被直接用于绘制虚拟物体,必须进行坐标转换和一些相关处理。图 2 给出了三个坐标系间的关系,下面给出传感器坐标系中的一点  $A(x, y, z)$  到视频坐标系的坐标转换方法。设从传感器得到的数据为  $(x_0, y_0, z_0)$  和矩阵  $M$ , 矩阵  $M$  为传感器坐标系到真实坐标系的欧拉矩阵,真实坐标系到视频坐标系的转换矩阵为  $M'$ 。传感器坐标系中的点  $A$  在真实坐标系中的坐标可由公式 (1) 得到,点  $A$  在视频坐标系中的坐标由公式 (2) 得到。

$$[x' y' z']^T = [x_0 y_0 z_0]^T + M \times [x y z]^T \quad (1)$$

$$[x' y' z']^T = M' \times [x' y' z']^T + [\delta x \delta y \delta z]^T \quad (2)$$

所谓注册就是使虚拟物体在真实环境中得到正确映射。在本系统中,坐标系转换完成后,根据两个传感器的返回结果分别得到头部和被操作物体在视频坐标系中的位置信息,进而对虚拟物体进行绘

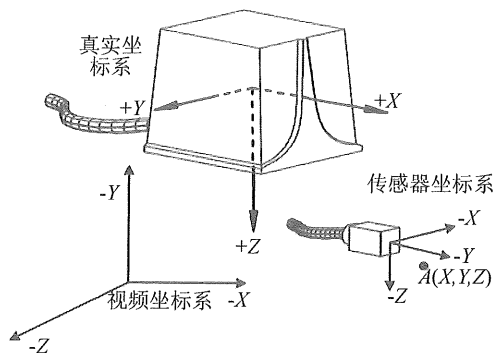


图 2 坐标系转换

Fig. 2 Coordination transformation

制，以实现虚拟物体和视频中的真实物体在用户操作过程中方位完全吻合。

### 1.3 本地滞后一致性控制

分布式 CAR 系统通过网络传输用户的物体操作信息，因此需要采用一致性控制来对整个系统的状态进行维护，使各用户看到统一的系统状态。具体实现时采用基于用户操作本地滞后的一致性控制方法<sup>[8]</sup>，该方法主要用于协同虚拟环境等连续交互媒体系统一致性控制，在 CAR 系统中是首次使用。

基于本地滞后的一致性控制技术原理非常简单，在本地用户触发一个操作后，系统不是立即执行，而是延迟一定时间后再执行，即操作的时间戳值要比操作被用户触发的时间大，这个时间延迟则被称为本地滞后，如图 3 所示。从图中可见，当本地滞后值为 0 时，系统会产生大量短时不一致，当本地滞后值增大时，系统短时不一致会减少，但用户操作的响应时间会延长。因此，必须对短时不一致和用户操作的响应时间进行综合考虑，选取合适的本地滞后值。

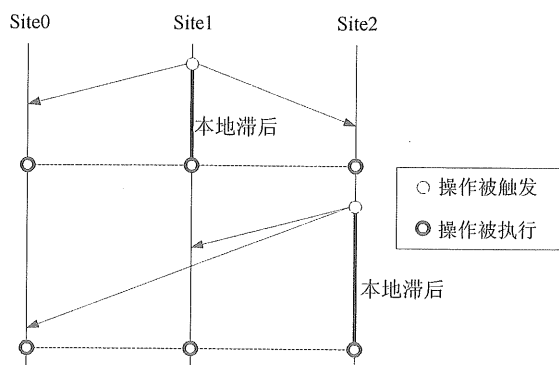


图 3 基于本地滞后的一致性控制技术

Fig. 3 Local lag based consistency maintenance

## 2 实验

从 1.3 节介绍可见，本地滞后值的选择对分布式 CAR 系统任务性能会有较大影响，因此，本文在开发好的系统原型基础上通过实验研究本地滞后值设置对系统协作性能的影响，以指导本地滞后一致性控制方法在类似分布式 CAR 系统中的应用。

### 2.1 实验任务

为对系统协作性能进行检验，实验采用的协作任务不仅要具有较高的交互性，而且要能让被试在较短时间内熟练掌握，因此采用文献 [9] 里使用的穿框任务。该任务要求一组中的两个被试分别控制框和杆进行交互，使框从杆的一个端点尽快移动到另一端点，同时杆必须在框的内部且不能和框发生碰撞。为增加交互性，均衡双方交互负担，要求在完成穿框任务时双方都必须控制相应物体运动，不能以一方静止不动一方运动的形式来完成任

务。图 4 (a) 为实验系统全貌，其中被试拿着一个捆绑了传感器的真实方框正在穿过虚拟杆，图 4 (b) 是被试通过头盔显示器看到的场景，图 4 (c) 是被试戴的头盔显示器正面示意图。

### 2.2 实验设计与实验设备

本实验采用组内设计，所有被测人员要在所有实验条件下尝试完成实验任务。共设置了本地滞后 0、100、200、300、400、500、600 ms 七组实验条件，条件出现顺序进行平衡设计。

实验用到的硬件设备如下：①两台配置一样的 (P4 1.8 G 处理器，512 M 内存，GeForce2 MX 400 显卡，100 Mbps 网卡) PC 以运行分布式 CAR 系统；②两套 ASCENSION 公司生产的 pciBIRD 用于完成用户头部和物体操作跟踪 (pciBIRD 的位置精确度为 1.44 mm RMS，角度精确度为 0.15°RMS)；③两套视频透视 HMD 都选用 Daeyang DH-4400VP，屏幕分辨率为 800 × 600；④摄像头采用了韩国现代 H103S 小型 USB 摄像头，最大分辨率为 640 × 480，最大光圈 F2.0，信噪比 > 42 dB，内部采用快速图像压缩引擎，可达 2 ~ 7 倍的压缩比率，由于采用了 35 万像素的 CMOS 传感器，其可在 352 × 288 分辨率下以 30 帧/s 的速度采集图像。

两台计算机通过 100 Mbps 局域网互联，由于网络状况良好，两台机器间的信息传输没有出现丢包。具体实验环境结构与图 1 一致，其中传感器与框的绑定见图 4 (b)，传感器绑定在框一条边的正中间，曲折杆和传感器的绑定通过将传感器固定在

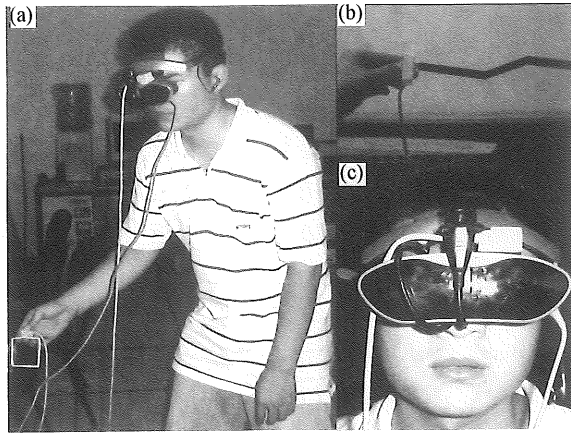


图 4 系统截图

(a) 系统全貌; (b) 用户视角; (c) 头盔显示器及摄像头

Fig. 4 Snapshots of the system

杆一条边的正中间。传感器、摄像头及 HMD 间的绑定见图 4 (c), 摄像头固定在 HMD 正中间, 然后将传感器固定在 HMD 上用来跟踪被试头部姿态。

### 2.3 实验人员与实验过程

本实验共有 20 人参加, 按两人一组被随机分为 10 组。实验被试全部是浙江大学的本科生和研究生, 年龄在 24 ~ 26 岁之间, 其中男生 18 人, 女生 2 人。

在实验开始前, 实验管理人员先向被试人员讲述实验内容和要求, 然后被试开始进行训练。由于被试对任务操作的熟练程度会直接影响到本实验的结果, 所以安排被试在正式实验前进行训练。训练首先在滞后值为 0 ms 下进行, 直至被试能顺利完成实验任务, 然后将滞后值改为 300 ms, 直至被试能顺利完成实验任务。在训练过程中, 管理人员给予被试一些相关操作技巧, 如使穿圈速度尽量保持一致, 棒尽量保持在圈的中心位置等。

训练结束后, 开始正式实验。正式实验分成 7 块, 每块对应一个本地滞后值。每组被试都必须按随机顺序完成全部 7 块内容。当两个被试都准备好后, 管理人员发出开始信号, 被试进行穿圈, 并计时。如穿圈过程中发生碰撞, 则被试退回起点重新开始。在一实验块中, 被试须成功穿圈 3 次才算完成任务, 每次成功后记录下穿圈所耗时间, 每块结束后统计总的尝试完成次数 (即碰撞发生次数), 并要求被试对穿圈过程的交互性能进行打分评价 (其中 10 分代表最好的交互性能, 0 分代表最差的交互性能)。在实验过程中, 两个被试间不允许以任何方式进行通信。此外, 实验还规定, 如一组被

试在某一实验条件下发生 20 次以上碰撞且还未完成任务时, 则该条件下的实验失败。

### 2.4 实验结果

图 5 为不同本地滞后值下的任务平均完成时间, 可见本地滞后值从 0 ms 增加到 100 ms 对任务平均完成时间没有影响, 但当本地滞后值超过 100 ms 时, 任务平均完成时间基本随本地滞后值的增长线性增长, 从 5.5 s 一直增长到近 10 s。

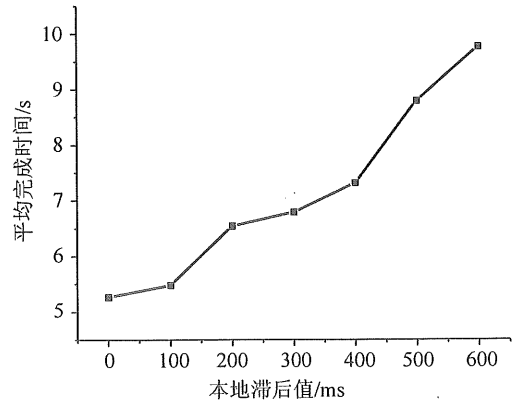


图 5 不同本地滞后值下的任务平均完成时间

Fig. 5 Mean time of completion under different lags

图 6 为不同本地滞后值下的任务平均完成次数, 可见本地滞后值从 0 ms 增加到 100 ms 不仅没有增加任务平均完成次数, 反而使之略有下降, 当本地滞后值从 100 ms 增加到 500 ms 时, 任务平均完成次数基本随本地滞后值的增长线性增长, 从约 6 次一直增长到约 11 次, 当本地滞后值增加到 600 ms 时, 任务平均完成次数有大幅增长的过程, 到达 15 次左右。

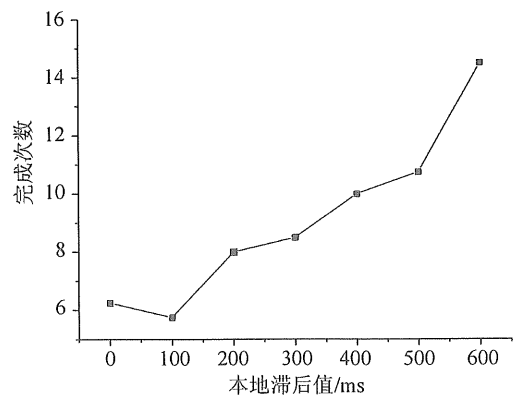


图 6 不同本地滞后值下的任务平均完成次数

Fig. 6 Mean trial count under different lags

图 7 显示的是被试在不同本地滞后值下对交互

性能的评价,可见本地滞后值从 0 ms 增加到 100 ms 对平均交互性能没有影响,但当本地滞后值超过 100 ms 时,系统平均交互性能基本随本地滞后值的增长线性下降,从近 10 一直下降到 1 左右。尤其当本地滞后值为 600 ms 时,被试对交互性能的评价仅为 1 左右,可见被试在此实验条件下已很难进行交互和协同工作。

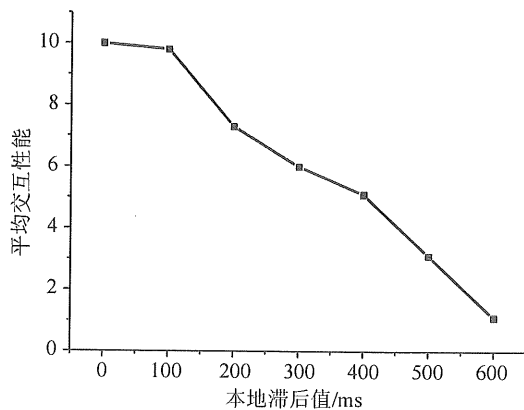


图 7 不同本地滞后值下的系统平均交互性能

Fig. 7 Mean interaction quality under different lags

此外,当本地滞后值为 600 ms 时,有 3 组被试实验失败(总碰撞发生次数大于 20),而当本地滞后值小于 600 ms 时,所有被试组均可顺利完成任务。

综合以上实验结果,可见本地滞后值对分布式 CAR 系统任务性能的影响是分阶段的,当本地滞后值小于等于 100 ms 时,任务平均完成时间、平均完成次数、平均交互性能均不受本地滞后值增加的影响,当本地滞后值从 100 ms 增加到 500 ms 时,任务平均完成时间、平均完成次数基本线性增加,平均交互性能基本线性下降,但本地滞后值的增加不影响用户完成任务,当本地滞后值大于等于 600 ms 时,各实验结果进一步恶化,且使部分被试组不能顺利完成实验任务。

根据以上结果和讨论,给出本地滞后一致性控制方法在分布式 CAR 系统中进行应用时选择滞后值的一些建议。对于网络状况良好的应用环境,系统信息传输延迟较小,100 ms 是很好的本地滞后值选择,其不会对系统交互性能造成任何负面影响。如具体应用环境网络状况不是很好,可选择设置大于 100 ms 的本地滞后值,但需注意最好不要设置大于 500 ms 的本地滞后值,因其会极大降低用户完成任务的性能,甚至使一些用户无法正常完成任务。

### 3 结 论

本文给出一个支持多用户协同操作的分布式 CAR 系统实例,并详细介绍其注册、跟踪、碰撞检测与一致性控制技术。此外,本文在原型系统上通过实验对本地滞后一致性控制方法滞后值选择对任务完成时间、完成次数和交互性能的影响进行研究。实验结果表明,当本地滞后值小于 100 ms 时,本地滞后机制对任务性能没有任何影响,当本地滞后值小于 500 ms 时,本地滞后机制对任务性能的负面影响随滞后值的增加而增加,而当本地滞后值大于 500 ms 时可能会使任务无法完成,因此在具体应用中建议不要将本地滞后值设置为大于 500 ms。文中给出的分布式 CAR 系统可被应用于游戏、教育、军事等领域,为用户提供虚实结合、一致的协同工作环境。

#### 参考文献:

- [1] AZUMA R. A survey of augmented reality [J]. Presence, 1997, 6 (4): 355 - 385.
- [2] FEINER S, MACINTYRE B, SELIGMAN D. Knowledge-based augmented reality [J]. Communications of the ACM, 1993, 36 (7): 52 - 62.
- [3] BAJURA M, FUCHS H, OHBUCHI R. Merging virtual reality with the real world: Seeing ultrasound imagery within the patient [J]. Computer Graphics, 1992, 26 (2): 203 - 210.
- [4] ROSE E, BREEN D, AHLERS K H, CRAMPTON C, et al. Annotating real world objects using augmented reality [C]. Proc of Computer Graphics International, 1995: 357 - 370.
- [5] BILLINGHURST M, BALDIS S, MILLER E, et al. Shared space: collaborative information spaces [C]. Proc of HCI International, 1997: 7 - 10.
- [6] OHSHIMA T, SATOH K, YAMAMOTO H, et al. AR2Hockey: a case study of collaborative augmented reality [C]. Proc of IEEE Virtual Reality Annual International Symposium, 1998: 268 - 275.
- [7] SCHMALSTIEQ D, HESINA G. Distributed applications for collaborative augmented reality [C]. Proc of VR International, 2002: 59 - 66.
- [8] MAUVE M. Consistency in replicated continuous interactive media [C]. Proc of ACM conference on Computer Supported Cooperative Work, 2000: 181 - 190.
- [9] CHEN L, CHEN H, CHEN G C. Echo: a method to improve the task performance of CVEs [C]. Proc of IEEE Virtual Reality, 2005: 269 - 270.