

## 基于激光笔的远程人机交互技术

刘芳 林学闯 史元春

(清华大学计算机系媒体所, 北京 100084)

**摘要** 当今投影仪已经广泛地应用于教学和会议之中,可是,传统的基于键盘和鼠标的人机交互仍然将演讲者限制在计算机设备旁边,从而给演讲者带来了不便.为解决此问题,提出了一种新的基于激光笔的远程人机交互系统.该系统只需要添加一个摄像机和视频采集卡就可以实现远程的人机交互,即在进行交互时,首先用一个摄像机实时拍摄投影屏,然后在得到的图中,检测激光点区域,并且跟踪识别它的轨迹,再将识别的结果作为用户输入,提供给计算机系统.与以往的类似系统相比,该系统的新颖之处有:在使用前即可训练出关于环境和激光点的特征,以用于增强系统的适应性;由于可通过融合多种信息(比如颜色、运动以及形状信息)来检测激光点,因而增强了系统的鲁棒性.实验结果表明,该系统具有较强的可用性(鲁棒性)和适应性.

**关键词** 计算机图象处理(520·6040) 远程人机交互 激光笔交互 虚拟鼠标 激光点跟踪

**中图分类号**: TP391 **文献标识码**: A **文章编号**: 1006-8961(2003)11-1356-05

### Remote Human-computer Interaction Using Laser Pointer

LIU Fang, LIN Xue-yin, SHI Yuan-chun

(Department of Computer Science and Technology, Tsinghua University, Beijing 100084)

**Abstract** Video projection is in widespread use for multimedia presentations in classrooms and in conferences. Unfortunately, the usual human-computer interaction (HCI) with keyboard and mouse confines the speaker to the computer devices. In this paper, we present a system for remote HCI using a laser pointer. In addition to the projection equipments, the system only requires a video camera and a video capture card. We use the video camera to capture the projection plane, detect the laser point (that is, the region induced by the laser pointer) in the captured frames. We then track and recognize the trajectory of the laser point and translate it into messages sent to the computer system. Compared with the previous work in this area, we develop some novel ideas in the system: first, a training process is introduced to improve the system adaptability; second, multiple cues such as color appearance, motion, and shape, are integrated in laser point detection to improve the system robustness. The experiments demonstrate the usability (robustness) and adaptability of our system.

**Keywords** Remote Human-computer interaction, Laser pointer interaction, Virtual mouse, Laser point tracking

## 0 引言

投影仪在教学和会议中已广泛地应用于多媒体信息的演示.它虽可以将演讲中丰富多彩的多媒体信息方便地传播给大量的听众,但是,由于目前传统的人机交互模式是基于键盘和鼠标的,因而将演讲者仍然限制在计算机设备的附近,不利于演讲者的自由走动和与听众的交流.为了解决这个问题,本文提出了一个基于激光笔的人机交互技术,以便让演讲者可以通过激光笔在远处同计算机系统进行交

互,就像使用鼠标一样方便.

基于激光笔的人机交互技术的想法是:首先利用摄像机拍摄投影屏,并在摄像机捕捉到的图中检测由激光投射到投影屏上形成的一个红色区域(以下称这个红色区域为激光点);然后通过跟踪来分析和识别激光点的轨迹,并将激光点的轨迹转化为不同的消息发送给计算机系统.在激光点的检测中,用到了不同的线索,如激光点的运动特征和模式特征<sup>[1, 2]</sup>,以及颜色特征<sup>[3]</sup>等.由于激光点的区域很小,模式的特征不是很明显,所以文献<sup>[2]</sup>易于受到噪声的影响;由于激光点的颜色特征在不同的使用环境中可以有很大的变化,因

**基金项目**: 国家自然科学基金项目(60103004, 69975009), 国家高科技发展 863 计划项目(2001AA114171)

**收稿日期**: 2003-01-28; **改回日期**: 2003-06-16

此,用固定的特征检测激光点,对于某些特定的场景效果很好,但是检测算法的适应性不是很好,而且在所有这些激光点的检测方法中,都没有考虑系统的适应性.而在激光点的行为描述上,文献[4]定义了一套复杂的使用激光笔的人机交互方式;文献[5]则通过实验给出了激光点交互中的一些参数,如判断激光点停留在一个位置附近所需的时间以及这个附近区域的大小.实践证明,这些参数对于系统的实现和应用具有较强的指导意义.激光笔人机交互系统与以前的工作不同的是:(1)由于在使用前设计了一个训练的过程,用以学习环(背)境和激光点的特征(如颜色值、形状)等,从而增加了系统的适应性;(2)由于融合了多种线索,如利用颜色、运动和形状来检测激光点,从而提高了检测激光点算法的鲁棒性.

## 1 系统综述

激光笔交互系统充分利用了原有的设备,如投影屏和激光笔.为了知道激光点的位置,需要一个摄像机拍摄投影屏,图 1 显示了激光笔交互系统的配置.在进行交互时,首先融合多种线索,在摄像机拍摄到的图中检测激光点,并进一步跟踪、分析和识别它的轨迹;然后将它在图平面上的坐标转化为投影屏上的坐标,再将其识别的结果作为用户的输入,提供给计算机系统,就像来自于真实的鼠标一样.

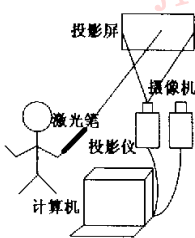


图 1 激光笔交互系统的配置图

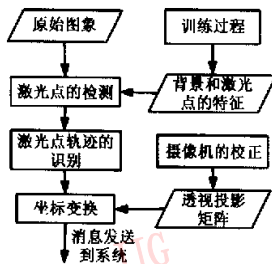


图 2 激光笔交互系统的流程图

激光笔交互系统主要由摄像机校正、背景和激光点的训练、激光点的检测以及激光点轨迹的识别等 4 个模块组成.图 2 是激光笔交互系统的流程图.

## 2 摄像机的校正

为了将摄像机图平面上的坐标转换成投影屏上的对应的坐标,必须对摄像机进行校正.假设摄像机

没有径向畸变(radial distortion),那么从图平面到投影屏的坐标变换就是一个透视投影变换,其可以用一个透视变换矩阵(Homography)来描述,它是一个  $3 \times 3$  的矩阵,若不考虑尺度因子,则它有 8 个自由度,可以由至少 4 对对应点求出.如果摄像机有径向畸变,则需要求出摄像机的内参数,以消除径向畸变,消除径向畸变的方法请参考文献[6].

为了求出透视变换矩阵,可以首先让投影影在投影屏上显示出一个白色的矩形框,然后在图中检测出对应区域(一个凸 4 边形),即可找出它的 4 个顶点,若将它们与矩形框的 4 个顶点对应起来,即可用这 4 个对应点求出透视变换矩阵.在图中分割出与矩形框对应的区域是一个 2 分类的问题,即找出一个阈值  $T$ ,如果由它将图象的像素分为两类,则可使它们对应的 Fisher 准则函数<sup>[7]</sup>最大,即

$$T = \arg \max_u \frac{(m_1 - m_2)^2}{S_1^2 + S_2^2}$$

设图象中灰度值小于  $th$  的像素为第 1 类,灰度值大于或等于  $th$  的像素为第 2 类,  $m_1, m_2$  分别表示第 1 和第 2 类的像素的灰度平均值;  $S_1$  和  $S_2$  分别表示这两个类的类内离散度;分割时,首先用阈值  $T$  将图象分割为二值化图,同时找出最大的连通区域,再提取这个区域的边缘点;然后利用这些边缘点即可拟合出 4 条直线;最后由这 4 条直线的交点即可确定该区域的 4 个顶点.

考虑到这里的透视变换是从一个小的图象平面“放大到”一个大的投影屏上,以及激光点位置估计存在误差的情况,因此摄像机校正算法的误差在实际应用中不是那么重要,即以上校正算法的精度已经足够,而更重要的是校正算法的鲁棒性,对于一个经常需要校正的系统而言,这就大大方便了使用.

## 3 背景和激光点的训练

在激光笔交互系统的实际应用中,由于不同的使用者所使用的环境不同,因此为了适应不同的使用环境,故提出了一个新的训练方法,这样系统经过简短的训练,就可以适应新的环境.通过训练,系统即可学习到背景的光照情况和激光点的特征.在激光笔交互系统中,应先考虑这样一种特殊的情形,即当投影机投影的是一片白色的区域时,如何检测激光点,因为此时激光点最容易“淹没于”较亮的背景图象之中,所以可设计如下训练过程:先在此种情况

下训练出背景图象,再分析出合适的阈值;然后用背景图象和阈值来检测激光点.除了最初建立起背景图象  $B$  之外,还训练出一些阈值,分别用于分割当前帧与背景图象灰度的差分图和当前帧,以及进行区域面积的验证.

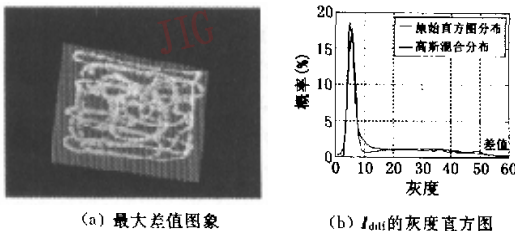
具体实现时,注意到激光点在投影屏上是一个非常亮的红色区域,由于摄像机存在截断效应 (clipping),因此在图象上,红色区域里面的点,其  $R$ 、 $G$ 、 $B$  各个分量的值都不超过 255,但是由于红色分量  $R$  是该像素最主要的分量,所以对于一个 RGB 彩色图,只需采用图象中的红色分量平面即可.本文中所述的灰度,在无特殊说明的情况下,都是指图象中像素的红色分量  $R$ .

(1) 打上激光点后,即可得到一些训练图象  $I_1, I_2, \dots, I_n$ .若是直接考虑单帧与背景图象的差值,由于激光点的区域很小,它将会“淹没于”噪声之中,所以应将多帧训练图象中的激光点叠加起来,使得最终的训练图象上含有多个激光点,这里最大差值图象  $I_{diff}$  定义为:

$$I_{diff}(x, y) = \max(I_1(x, y) - B(x, y), I_2(x, y) - B(x, y), \dots, I_n(x, y) - B(x, y))$$

其中,  $I_n(x, y)$  为每帧含有激光点的训练图的灰度值,  $B(x, y)$  为背景图象像素灰度值.

$I_{diff}$  在没有激光点经过的地方表示了该处相对于背景的最大噪声水平,而在激光点经过的地方则是激光点与背景图象的差值.图 3 是得到的一个差值图象  $I_{diff}$  以及它的灰度直方图.由该直方图明显可见,  $I_{diff}$  灰度分布主要由两个成分组成:一个是靠近 0 的峰,其代表噪声;一个是远离 0 的平缓的区域,其代表了激光点的灰度与背景图象灰度差值的分布.据研究,可以用混合高斯分布去逼近其真实的分布,图 3 显示了混合高斯分布逼近直方图的结果.由第 2 个成分的均值和方差就可以得出当前激光点与背景图象灰度差值的阈值  $T_{diff}$ .这样,当前帧中只



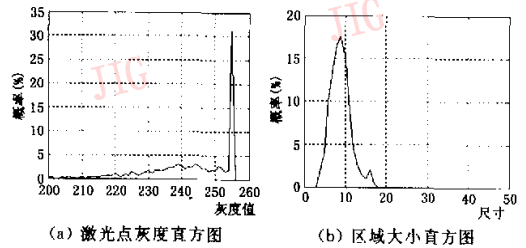
(a) 最大差值图象 (b)  $I_{diff}$  的灰度直方图  
图 3 最大差值图  $I_{diff}$  和它的灰度直方图

有灰度值与背景图象灰度值的差值大于  $T_{diff}$  的那些像素才可能属于激光点.

(2) 因为从直方图上看,代表了激光点和噪声的两个成分在有些地方是重叠的,所以由阈值  $T_{diff}$  分割后的图象中还含有一些噪声.考虑到激光点区域具有较大的灰度值,而噪声的灰度又跟背景图象灰度相关,且背景图象的灰度在不同地方的差别可能很大,所以可以根据像素的绝对灰度去除掉一部分噪声,即先统计出激光点的绝对灰度值的直方图,再根据一定的可信度水平 (confidence level) 来求出阈值  $T_{intensity}$ ,使当前帧中只有灰度大于  $T_{intensity}$  的像素才属于激光点.

(3) 另外,由于激光点是一个具有一定大小的连通区域,且通过训练可以统计出激光点区域大小的分布,因此也就可以确定激光点的面积的范围  $T_{Size\_Low}$  和  $T_{Size\_High}$ .图 4 显示了激光点的绝对灰度值和区域大小的直方图.

下面考虑一般的情况.实验表明,在一般的情况下,由于激光点在图象中的灰度值变化不大,所以还是可以利用以上训练得到的背景图象和阈值来检测激光点.



(a) 激光点灰度直方图 (b) 区域大小直方图  
图 4 激光点的灰度直方图和区域大小(以像素为单位)的直方图

### 4 激光点的检测

激光点的检测融合了多种线索,其中包括它的颜色特征、运动信息和形状特征.这种多线索的融合不仅可使得激光点的检测更加鲁棒,并可减少误检(在没有激光点的地方错误地检测到了激光点)以及漏检(在有激光点的地方没有检测到激光点)的概率.激光点的检测主要由如下 3 个子模块组成:

#### 4.1 颜色分割

如果图象中某个像素的绝对灰度值大于  $T_{intensity}$ ,并且与背景图象灰度的差值也大于  $T_{diff}$ ,则将它作为候选的属于激光点区域的像素.定义颜色

分割得到的图象为  $I_{color}$

$$I_{color}(x,y) = \begin{cases} 1 & \text{if } I(x,y) - B(x,y) > T_{diff} \\ & \text{and } I(x,y) > T_{intensity}; \\ 0 & \text{otherwise} \end{cases}$$

这里  $I(x,y)$  是图象中  $(x,y)$  处像素的灰度值. 阈值  $T_{intensity}$  和  $T_{diff}$  分别是在训练过程中得到的灰度值阈值和当前激光点与背景图象灰度值差值的阈值.

### 4.2 运动检测

当用户手持激光笔的时候,激光笔会有一些微小的抖动(jitter),由于其在图中表现为激光点也是在不断抖动的,因而可以利用运动的信息来帮助进行激光点的检测,具体地说,就是利用帧间差来进行检测. 这里定义运动检测得到的图象为  $I_{motion}$ :

$$I_{motion}(x,y) = \begin{cases} 1 & I_t(x,y) - I_{t-1}(x,y) > T_{diff} \\ 0 & \text{otherwise} \end{cases}$$

这里  $I_t(x,y), I_{t-1}(x,y)$  分别为当前帧和前一帧中的像素  $(x,y)$  的灰度值.

### 4.3 形状验证

由以上的  $I_{color}$  和  $I_{motion}$ , 首先可以得到一个二值化图  $I_{support}: I_{support} = I_{color} \wedge I_{motion}$ ; 其中,  $\wedge$  表示两个图象的与运算; 然后利用形态学的算子去除  $I_{support}$  的噪声后, 即可找出所有的连通域, 再根据训练得到的激光点的面积范围  $[T_{Size\_Low}, T_{Size\_High}]$  去掉那些太大(可能是背景灰度的变化)或者太小(可能是噪声)的连通域. 若是只剩下一个合乎要求的连通域, 那么就可将它作为激光点的估计.

## 5 激光点的轨迹识别

由于使用者手的抖动,激光点的轨迹一般是不规则,而且是粗糙的,为此在识别它的轨迹之前,应先用 Kalman 滤波器平滑它的轨迹. 目前,针对智能教室的需求,已定义了如下 4 种动作(消息):

- (1) SpotOn: 激光点开始出现在投影屏上
- (2) SpotOff: 激光点从投影屏消失.
- (3) SpotMove: 用于将与图中激光点对应的投影屏上坐标发送给系统,使得光标随之而动.
- (4) SpotSelect: 用于用户选择某个目标,其对应于鼠标的双击动作. 为了完成一个选定的动作,用户需要先将激光笔停留在某个位置附近一会,然后粗略地画一个圈.

激光点轨迹的识别利用了有限状态机. 目前定

义了如下 4 个激光点状态: None (没有激光点), Move (激光点的移动), Static (激光点停在某个位置附近), Track (跟踪识别激光点的轨迹). 图 5 显示了有限状态机的状态以及它的输出(即动作或消息).

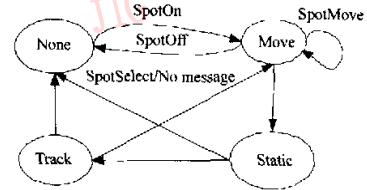


图 5 有限状态机: 状态和消息

由图 5 可见,系统的初始状态是 None, 当连续若干帧检测到激光点后,系统就进入状态 Move, 并且发送消息 SpotOn, 此时光标将随着激光笔的移动而移动; 在 Move 状态, 当检测到激光点的位置连续若干帧处于一个窗口里面时, 则系统进入 Static 状态; 在 Static 状态, 如果激光点出了那个窗口, 那么系统就进入了 Track 状态; 在 Track 状态, 系统将激光点的轨迹记录下来, 当激光点在画完一个圈后又回到以前的某个位置的附近, 则系统回到 Move 状态, 并且向计算机系统发送消息 SpotSelect 以及选中目标的位置, 如果连续若干帧激光点都没有回到以前某个位置的附近, 那么系统就不再向计算机系统发送消息而回到 Move 状态; 如果连续若干帧都没有检测到激光点, 则系统的状态才回到 None, 这使得即使有少量的漏检, 系统也还是稳定的.

## 6 实验和讨论

决定这个系统可用性的关键问题在于激光点检测的鲁棒性, 以及识别激光点轨迹对于用户的要求. 对于后者, 用户不需要经过专门的训练, 就可以熟练地使用该系统; 对于前者, 可以用以下 3 个参数来评价系统的可用性: 正确检测率(系统正确地检测到的激光点帧数占总测试帧数的百分比), 误识率(检测到错误的激光点帧数在总测试帧数中所占比例), 以及漏识率(没有检测到存在激光点的帧数在总测试帧数中所占的比例). 激光点轨迹的识别算法可以容忍一定的漏识, 因为漏识只是意味着发送给系统的激光点数目减少了, 所以漏识是不太重要的; 但是由于误识会造成激光点轨迹识别的错误, 所以应该尽量减少误识率, 这虽然会使正确检测率减少, 漏识率增

加,但是对于实用性而言,并不太重要.

文献[1,2]声称他们的系统正确检测率达到了50%,且在实验中没有发现误识.而本系统,对5个长度为1000帧的视频序列进行测试,结果表明该系统正确检测率为93%,误识率为1%,漏识率为6%.通过提高相应的阈值  $T_{intensity}$ ,  $T_{diff}$  和  $T_{Size\_Low}$ ,  $T_{Size\_High}$ , 正确识别率可达81%,误识率为0,漏识率为19%.另外还可利用卡尔曼滤波器来平滑激光光点的轨迹,以减少噪声对整个轨迹的影响.

如前所述,激光笔交互系统通过增加训练过程,

从而提高了系统的适应性;通过融合多线索检测激光点,又提高了系统的鲁棒性.实验证明,系统的实时性和鲁棒性都很好,具有较好的实用价值.

激光笔交互系统已作为智能教室中智能人机交互的一部分在运行.该系统运行于 Dell Pentium I, 400 MHz 的 CPU 和 Windows 2000 上.系统每秒能处理 25 帧,系统运行时,CPU 的负载在 30% 左右.图 6 显示了在智能教室中使用激光笔交互的两个真实场景<sup>[8]</sup>.今后的工作是定义并识别新的动作.

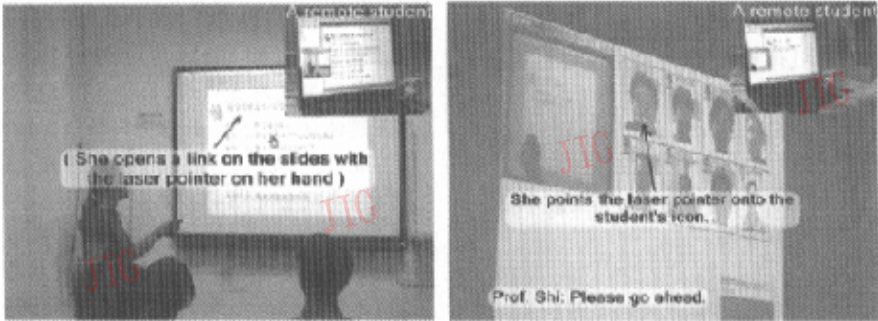


图 6 激光笔远程人机交互系统在智能教室中的应用场景

## 参 考 文 献

- 1 KIRSTEIN C, MULLER H. Interaction with a projection screen using a camera-tracked laser pointer [A]. In: Multimedia Modeling Proceedings [C]. Lausanne, Switzerland; IEEE Computer Society, 1998;191~192.
- 2 KIRSTEIN C. A System for human-computer interaction with a projection screen using a camera-tracked laser pointer [R]. Technical Report 686, Informatik VII, University of Dortmund, German 1998.
- 3 ECKERT R R, MOORE J A. The classroom of the 21st century: The interactive learning wall [J]. Special Interest Group-Computer Human Interaction Bulletin, 2000, 23(2):33~40.
- 4 OLSEN D R, NIELSEN T. Laser pointer interaction [A]. In: Proceedings of Association for Computing Machinery Computer Human Interaction Conference: Human Factors in Computing Systems[C]. Seattle, WA, USA, 2001; 17~22.
- 5 PECK C H. Useful parameters for the design of laser pointer interaction techniques [R/OL]. <http://www.cs.cmu.edu/~pebble>.
- 6 ZHANG Z. A flexible new technique for camera calibration [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(11):1330~1334.
- 7 边肇祺,张学工. 模式识别(第二版)[M]. 北京:清华大学出版社, 2000, 1.

- 8 刘芳,林学闯. 激光笔远程人机交互系统[EB/OL]. <http://media.cs.tsinghua.edu.cn/~liufang>.



刘芳 1977年生,2000年获华中理工大学学士学位,2003年获清华大学计算机系硕士学位.研究方向为人工交互、计算机视觉、多媒体信息处理.



林学闯 1940年生,清华大学计算机系教授,博士生导师.研究方向为计算机视觉、图象处理、模式识别.



史元春 1968年生,教授,1999年获清华大学计算机系博士学位,现担任清华大学计算机系人机交互所副所长.研究方向为分布式多媒体技术、普适计算.