IrPen

——用于平板电脑交互的6自由度笔

|  |  |
| --- | --- |
| **Jaehyun Han and Seongkook Heo** | *Korea Advanced Institute of Science and Technology* |
|  |  |
| **Hyong-Euk Lee** | *Samsung Advanced Institute of Technology* |
|  |  |
| **Geehyuk Lee** | *Korea Advanced Institute of Science and Technology* |

# 引言

传统的笔交互界面使用2维平面的笔尖位置进行交互。为了增加笔交互的输入维度，研究人员也使用笔的移动方向、笔到屏幕平面的距离作为输入。例如，Tovi Grossman和他的同事[[1]](#endnote-1)、Sriram Subramanian和他的同事[[2]](#endnote-2)提出的笔交互系统可使用笔的屏幕平面上的3个维度的位置。Xiaojun Bi和他的同事使用笔的移动方向来调用上下文菜单[[3]](#endnote-3)。

此外，研究人员使用这些输入维度来使3D操控更直观。例如，一种光线投射方法使用笔的位置和方向来在3维虚拟环境中选择对象。[[4]](#endnote-4) Baxter和他的同事在他们的绘图应用程序中使用笔的6个自由度（DOF）来模拟各种类型画笔的物理特性。[[5]](#endnote-5)

**除了结合了IrCube跟踪器的基本操作的原则，IrPen还考虑到了平板电脑的特殊要求。它为平板电脑使用了足够小的传感器和能降低平板平面反射影响的笔结构，该设计还解决了与闭塞环境的光照问题**

然而，当前的笔（程序）系统跟踪范围有限，所以不能完全支持这些交互技术。所以，研究人员使用了可替代的传感工具，例如Vicon运动跟踪系统或者9自由度的传感器模块。[[6]](#endnote-6) 但是，这些工具在平板电脑上是不可实现的。（更多关于笔接口的信息，请参见边栏。）

要为平板电脑打造出实用的六自由度笔系统，我们回顾6 自由度跟踪方法，包括磁，光，和声学追踪器。我们发现， IrCube系统是最有潜力的，因为它的硬件结构包括:一些照片传感器和LED指示灯，我们可以很容易的把它们分别嵌入到平板和笔中。我们为一个专门为平板使用的笔系统定义了一些附加的要求。在这些要求的基础上，我们实现了原型，并通过测量它的性能证明其可行性。

# **IrCube**

IrCube追踪一个指针的位置和方向，这个指针包含一个发光二极管（LED）阵列，在每个方向上都有一个发光二极管。传感器测量LED（光）强度；然后ircube采用优化算法求解逆问题。当传感器固定而LED移动时，传感器上会测量到LED的位置和移动方向。这种前向模型为:

|  |
| --- |
| , |

F0是第m个LED（光照）强度，d是LED灯到第n个传感器的距离，DLED和Ddiode是LED和二极管的方向参数，θn和ϕm是一个传感器到一个LED的相关角度。图2解释了这个模型。

LED灯按时间按顺序打开和关闭，一次一个，IrCube在每个传感器测量其光强度，在一个完整的照明周期，我们得到了m\*n个传感器数据。如果我们了解每个传感器的位置、方向、方向性和灵敏度，和每个LED的位置、趋向、方向性和强度，那么就可以使用前向模型来预测这些值。

|  |  |
| --- | --- |
|  |  |
| 图1. IrPen交互空间。IrPen扩展了IrCube跟踪系统使其考虑到平板电脑的特殊需求。 | 图2. IrCube跟踪系统的前向模型[[7]](#endnote-7)。当LED与传感器的距离和角度给定时，它给出了传感器的输出。d是第m个LED和第n个传感器的距离，θn和ϕm是传感器到LED的相关角度。 |

为了确定指针的位置和方向，我们使用一个优化算法最大限度地减少错误函数E:

|  |
| --- |
| , |

Omn是传感器测量结果，F(xm-xn , qm)是前向模型的预测结果，xn和xm是第n个传感器和第m个LED的位置，qm是一个四维向量代表指针的趋向。

IrCube的最大传感器尺寸142\*82mm，使得很难将他集成到平板上。在一次平板上使用IrCube传感器的实例中，此传感器和平板一样大。7 ­所报告的位置误差分别是静态条件下23.5 ， 20.2 ，和27.5毫米（分别在三个轴方向），动态条件下28.2，34.1，和18.1毫米。这样的精度用在电视上是可以接受的，但在平板上，位置误差必须要减小。Seongkook Heo和他的同事承认，由于反射作用，IrCube有一个基本的限制。7在平板电脑上使用IrCube，当指针接近平板表面，平板表面的反射会造成指针位置消失。

# **IrPen**

IrPen系统包括一支笔和一个传感器框架，该框架安装在带有11.6 英寸显示器的三星Series 7平板上（见图3a）。我们使用C#语言实现了跟踪器软件。

## **笔**

这支笔包含12个LED，一个笔尖，一个压力传感器和一个按钮（见图3b）。首先，LED灯同时打开、关闭发送同步信号。它们按顺序打开和关闭，一次一个。此过程发生两次，一次以全光强度发生，另一次以大约20%的光强度发生。这种重复克服了传感器动态范围有限的问题。整个过程大约花费15毫秒。

当笔位于交互空间时，用户按下按钮来选择一个对象或者使拖动。当笔触及平板表面时，笔尖将会按动压力传感器。笔将压力值和按钮状态同步编码，并将信号发送给传感器框架。

## **传感器框架**

该传感器框架包含六个传感器模块和一个微控制器。传感器位于平板电脑的四角以及顶部中心和底部边缘（见图3a）。这些传感器覆盖在屏幕上，并解决了遮挡问题。微控制器使用12位 模拟 - 数字 转换器来读取传感器的输出电压，并通过USB连接将这些值发送到计算机上。USB总线可以支持发送传感器帧的供能，因为每个传感器模块电流小于30毫安。

|  |
| --- |
|  |
| 图3a IrPen的整个系统，IrPen将采用的6个传感器安装在带有11.6 英寸显示器的三星Series 7平板上 |
|  |
| 图3b. 笔的原型 |
|  |
| 图4. 信号发生步骤，些步骤的计算时间约为13毫秒，使用3-GHz的 Intel Core i5-2320 CPU 。 |

## **信号发生**

传感器框架的数据采集时间是大约15毫秒，数据包从框架发送到跟踪软件大约需要13毫秒。数据包包含146个整形数（12 个LEDs × 6 传感器 × 2 个光强度等级 + 按钮信息和按压状态）。当系统接收到数据包时，跟踪器软件需要三个信号处理步骤（见图4）。总的来说，数据帧速率大小为35fps，这些步骤的计算时间约为13毫秒，使用3-GHz的 Intel Core i5-2320 CPU 。

第一步将传感器数据归一化。每个传感器芯片具有不同的传感范围和灵敏度，所以，这是必需的。我们针对每个芯片使用线性的映射函数将传感器的输出映射到物理世界的光照强度。我们实验性的确定了这些功能。传感器的输出有两个——一个是每一个LED的光强度。我们设计了信号处理模块来选择一个不饱和输出。若两个输出均是不饱和的，该模块选择较大的一个，因为它具有较高的信噪比。若两个输出均是饱和的，模块则丢弃它们。

在第二步，优化算法使用第一步获得的归一化值和初始猜测的位置和趋向来预测笔的位置和趋向。这种猜测假定笔尖的速度是恒定的。我们使用信赖域反射算法来优化；[[8]](#endnote-8) 随着它的运行，前向模型预测传感器的输出。如果优化的剩余部分超过了某个阈值，该算法将再次运行不同的初始猜测（如图4中的虚线表示）。

在第三步，运行一个误差校正器和一个稳定过滤器。因为最初我们在预测中发现了系统误差，我们引入校正器来增加IrPen的准确度。我们共三个多项式函数（每个轴一个）来对这种误差建模。

我们筛选误差校正器的结果，来降低手颤造成的运动影响。我们首先尝试了低通滤波器来保证稳定性，但这样做会导致延迟效应。为了解决类似的问题，之前的研究使用自适应的低通滤波器，该滤波器使用指针的平均速度来控制滤波器的极值大小。[[9]](#endnote-9)该滤波器能有效解决手颤摇晃造成的问题，并且不会造成延迟效应。我们使用了这个滤波器，它在我们的系统上同样有效。

# **从IrCube到IrPen**

从IrCube到IrPen我们遇到了主要的几个技术问题。

## **手部遮挡**

因为拿着IrPen的手可能会遮挡住一个传感器，IrPen系统应该能够容许1个或2个传感器被遮挡。这是可以被满足的，因为传感器输出数量（6个传感器×12个LED = 72个输出）远远超过了优化算法所需要的参数数量。

在我们考虑遮挡问题之前，我们决定使用6个传感器。为了覆盖整个屏幕，我们认为至少需要6个传感器。

1. T. Grossman et al., “Hover Widgets: Using the Tracking State to Extend the Capabilities of PenOperated Devices,” Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI 06), 2006, pp. 861–870

   . [↑](#endnote-ref-1)
2. S. Subramanian, D. Aliakseyeu, and A. Lucero, “Multi-layer Interaction for Digital Tables,” Proc. 19th Ann. ACM Symp. User Interface Software and Technology (UIST 06), 2006, pp. 269–272. [↑](#endnote-ref-2)
3. X. Bi et al., “An Exploration of Pen Rolling for PenBased Interaction,” Proc. 21st Ann. ACM Symp. User Interface Software and Technology (UIST 08), 2008, pp. 191–200 [↑](#endnote-ref-3)
4. J.J. LaViola and D.F. Keefe, “3D Spatial Interaction: Applications for Art, Design, and Science,” ACM Siggraph 2011 Courses, 2011, article 1. [↑](#endnote-ref-4)
5. B. Baxter et al., “DAB: Interactive Haptic Painting with 3D Virtual Brushes,” Proc. Siggraph, 2001, pp. 461–468. [↑](#endnote-ref-5)
6. . D. Lee et al., “PhantomPen: Virtualization of Pen Head for Digital Drawing Free from Pen Occlusion & Visual Parallax,” Proc. 25th Ann. ACM Symp. User Interface Software and Technology (UIST 12), 2012, pp. 331–340. [↑](#endnote-ref-6)
7. S. Heo et al., “The IrCube Tracker: An Optical 6-DOF Tracker Based on LED Directivity,” Proc. 24th Ann. ACM Symp. User Interface Software and Technology (UIST 11), 2011, pp. 577–586. [↑](#endnote-ref-7)
8. J. More and D. Sorensen, “Computing a Trust Region Step,” SIAM J. Scientific and Statistical Computing,

   vol. 4, no. 3, 1983, pp. 553–572. [↑](#endnote-ref-8)
9. G. Lee et al., “A TV Pointing Device Using LED Direct ivit y,” Proc. 2011 IEEE Int’l Conf. Consumer Electronics(ICCE 11), 2011, pp. 619–620. [↑](#endnote-ref-9)