

MTDriver:一种改进的多点触摸桌面工具包*

谭国富^{1,2+}, 刘佳升^{1,2}, 戴志军¹, 张凤军¹, 王宏安¹, 戴国忠¹

(中国科学院 软件研究所 人机交互与智能信息处理实验室,北京 100190)

(中国科学院 研究生院,北京 100190)

MTDriver: An Improved Multi-Touch Table Toolkit

TAN Guo-Fu^{1,2+}, LIU Jia-Sheng^{1,2}, DAI Zhi-Jun¹, ZHANG Feng-Jun¹, WANG Hong-An¹, DAI Guo-Zhong¹

(Intelligent Engineering Laboratory, Institute of Software, The Chinese Academy of Sciences, Beijing 100190, China)

(Graduate University, The Chinese Academy of Sciences, Beijing 100190, China)

+ Corresponding author: E-mail: tanguofu@gmail.com

Tan GF, Liu JS, Dai ZJ, Zhang FJ, Wang HA, Dai GZ. MTDriver: An Improved Multi-touch Table Toolkit. *Journal of Software*, 2010,21(Suppl.):123–132. <http://www.jos.org.cn/1000-9825/10013.htm>

Abstract: Among the implementations of multi-touch technologies, computer vision is a widely used way to build the multi-touch table. The touch information are obtained through the processing the imaging of infrared lights which comes from the reflection on touching fingers. The processing includes finger area extraction, tracking, and rectification. Because of disturbance of environment and unequally spread of the infrared lights on the surface, the image processing methods of previous multi-touch toolkits have poor performance on the detection and tracking of touch fingers, and these methods have not considered the influence of finger moving and camera distortion. In this paper, we propose a multi-touch table toolkit named as MTDriver. In MTDriver, we use local extreme area to extract the possible areas of touch fingers, then we track the motion of touch fingers by the tracking method which use the adjacent frames information. Experiments show that MTDriver has higher precision, better performance, and more robust compared with the other toolkits.

Key words: multi-touch; touch table; computer vision; human computer interaction

摘要: 目前多点触摸桌面广泛采用计算机视觉技术实现。触摸信息是通过手指反射的红外线在红外摄像机下成像,然后对红外相机得到灰度图像进行手指区域提取,跟踪,校正得到。由于桌面表面红外光照射不均衡,环境噪声干扰等因素,目前存在的多点触摸工具包在检测和跟踪方面效果较差,也没有考虑手指运动和摄像机畸变的影响。本文提出基于图像局部极值点检测手指触摸区域,结合手指运动和相邻帧信息进行手指跟踪,实现摄像机畸变校正的多点触摸桌面系统工具包 MTDriver。实验结果表明,MTDriver 跟踪识别准确,效率高,鲁棒性强,具有实用性。

关键词: 多点触摸;触摸桌面;计算机视觉;人机交互

* Supported by the National Natural Science Foundation of China under Grant Nos.U0735004, 60673188 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant Nos.2009AA01Z337, 2009GJA00045 (国家高技术研究发展计划(863)); the National Basic Research Program of China under Grant No.2009CB320804 (国家重点基础研究发展计划(973))

Received 2010-05-01; Accepted 2010-11-09

多点触摸是人机交互领域的一个研究热点,触摸交互简单、方便、快捷,是自然的人机交互方式.微软在 2006 年推出的 Surface Table^[1]是最早的多点触摸桌面产品,提供了炫丽的系统界面,支持多人同时进行触摸操作.索尼在 2010 年 6 月发布 atracTable^[2],不但支持多点触摸交互,还支持与手机移动等设备交互.这些多点触摸桌面应用在餐馆、酒吧、酒店、展馆、博物馆等地,带来了全新的用户体验.这些多点触摸桌面是基于计算机视觉技术实现的.2005 年纽约大学的 Han 提出基于 FITR^[3]多点触摸桌面,基于视觉的多点触摸系统便引起人们的关注.随后人们又提出 DI、LLP、DSI 等技术^[4],相对于传统的单点触摸设备,基于视觉的多点触摸交互设备支持多点触摸交互,系统造价低,容易搭建,易于扩展.

就多点触摸桌面而言,各种应用场景光照存在差异,如阳光、灯光中也包含高强度的红外线,所以背景图像是在动态变化的,需要对背景建模.红外线照射桌面表面的方式不一样,桌面材料对红外线的吸收、反射程度也各有差别,如果两个手指靠的很近时,其图像上对应的区域会重叠在一起,对于红外线平行照射,靠近红外激光灯的手指会遮挡其后面的手指,可能导致后面手指部分没有被红外线照射,在灰度图像上形成的区域发生断裂,红外摄像机得到的灰度图像的整体、局部亮度存在较大差异.所以无法使用全局阈值分割来检测手指区域,为此我们通过区域极值点检测触摸区域,检测到当前帧的触摸区域后,需要对正在跟踪手指和触摸区域相互匹配,目前多采用最短距离匹配的算法,在桌面手指点较多,手指运动较快时,不同手指间距离很近,这样距离最近并不一定是同一个触摸手指,算法误差很大,为此我们提出首先对手指运动建模,预测手指的位置,然后结合距离和方向计算手指和检测区域的匹配相似度,采用最优二分匹配确定跟踪手指对应的区域,保证全局匹配最优.对手指区域重叠和断裂的情况分为两个阶段处理,第一阶段将每个区域作为一个单个候选的手指区域去和当前跟踪的手指进行匹配,以保证尽可能为每个跟踪的手指找到一个对应的区域.第二阶段对于未匹配区域,把相互连通的区域作为一个手指点处理,以解决由于遮挡而出现区域断裂情况.同时在搜索过程中为手指区域建立空间索引,以加快搜索速度.触摸桌面系统中的红外摄像机一般采用广角相机以扩大拍摄范围,广角摄像机存在严重的摄像机畸变,同时投影的图像并不完全重合桌面,因此需要对跟踪到的手指的位置校正.

针对上述问题,我们提出改进的检测、跟踪算法,实现校正算法,将所有算法的实现封装为工具包 MTDriver,提供应用程序接口,将得到手指触摸信息包装成触摸消息发送给应用程序.本文首先介绍了目前多点触摸工具包现状,然后介绍多点触摸桌面系统原理,然后论述我们提出的手指检测算法、手指跟踪算法、校正算法,接着给出 MTDriver 工具包的框架和编程接口,随后对 MTDriver 和一些开源工具包进行性能对比测试,然后通过 MTEarth 多点触摸应用对 MTDriver 进行应用验证,最后进行总结.

1 相关工作

多点触摸工具包的功能是从红外线灰度相机获取灰度图像,检测触摸手指区域,跟踪触摸手指运动状态,将图像中的手指触摸区域还原到桌面上实际点,然后将这些触摸信息发送给多点触摸应用程序.目前发布的一些工具包^[5-7]兼容 FTIR、DI、LLP、DSI 等技术的多点触摸桌面设备,提供简单的编程接口.多点触摸应用程序通过响应多点触摸工具包发出的触摸事件消息,实现多点触摸的用户界面.下面介绍其中一些广泛应用的工具包.

reacTIVision^[5]是 Martin Kaltenbrunner 等人在 2006 年发布的多点触摸工具包.最初它用来在 reacTable^[6]跟踪和识别标志块.在其新发布的 1.3 版本中加入了手指跟踪的功能.其为手指和标志块分别建立了模板,采用局部自适应阈值分割的方法,通过模板匹配来检测手指和标志块的位置,在跟踪上结合了多层的预测跟踪方法.由于其对标志块采用特别的编码,它可以有效的跟踪标志块,但是对触摸手指的匹配和跟踪,模板匹配的方法却不理想,其采用自适应阈值二值化图像的方法也不能适应不同桌面图像灰度分布不均的情况.

NUIGroup 是国际著名的多点触摸交互研究组织.他们在 2007 年发布了 Touchlib^[7],随后在 2009 年又发布了 CCV^[8].Touchlib 主要工作在 Windows 平台之上,用一系列滤波器对输入图像滤波,用单一的阈值对图像进行二值化,然后提取轮廓来识别手指.采用最小距离优先算法^[9]来实现跟踪,其算法复杂度与手指个数成指数函数关系,随着手指增多,算法消耗时间成倍增长.滤波器参数需要对每个桌面具体应用环境作调整,配置复杂,由于其采用单一阈值对图像二值化,不能识别手指区域过暗或手指区域很近的情况,致使跟踪产生误差.

CCV 是跨平台的多点触摸工具包,支持各个操作系统平台.它提供一个交互设置参数界面.通过用户实时调整图像处理过程的一些参数来提取手指区域,在噪声很大的情况下,很难找到一个最优阈值,同时环境也不稳定,阈值需要经常调整.在跟踪算法上其采用 KNN 跟踪算法,只在最近的 K 个候选对象中寻找匹配,有很高的效率,但是其没有处理好匹配冲突的情况,跟踪准确度不高.

在校正上,这些工具包只采用线性插值的办法,没有考虑摄像机畸变.而多点触摸桌面往往采用广角摄像机来拍摄更大桌面范围图像,广角镜头会产生严重的畸变,没有校正会使触摸点的定位存在误差.

2 多点触摸系统原理

2.1 多点触摸系统原理

我们所采用的多点触摸桌面是基于 LLP 技术的,如图 1 所示,其长宽高分别为 104cm,140cm,90cm.结构如图 1(b)所示,投影仪和红外摄像机放在触摸桌面屏幕的下方,投影仪将影像投影到半透明的桌面屏幕上,桌面屏幕便可以显示清晰的图像.四个红外线激光灯放置在桌面四角上,整个桌面用边框包围起来,从红外线激光灯发出厚度为 0.5mm 的平行桌面的红外线光铺在桌面表面上.当手指触摸到桌面时,手指边缘便会反射红外光,位于下面摄像机会拍摄到包含触摸信息的图像.图 1(c)是当双手十个手指触摸在桌面时,红外摄像机获得图像,从图中可以看出大约有 10 个白色区域,其便是手指对应的触摸区域,在四角还有 4 个白色区域是桌面 4 个角上红外线灯所形成区域,通过对原始图像进行检测手指,跟踪手指,校正,便可以得到手指的触摸信息.

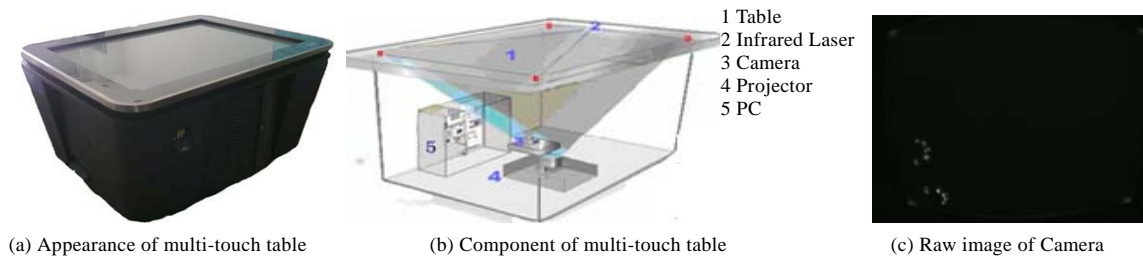


Fig.1 Multi-Touch table

图 1 多点触摸桌面

2.2 检测触摸手指

一般检测方法是采用一个阈值对图像进行二值化,然后查找轮廓,找到的轮廓变对应一个手指.但桌面上的红外线照射是不均匀的,手指靠的太近对应的区域重叠在一起,手指间会遮挡红外线光致使手势触摸发生断裂.阈值过小重叠区域就识别成一个区域,阈值过大断裂的区域识别为多个手指,而且各个区域的亮度是不一样的,所以二值化的方法检测触摸手指的准确度很低.考虑到每个手指触摸区域中包含一个或多个灰度值极大点,称之为极值点,通过检测图像中的极值点,便可以得到全部可能的手指区域,对于相连通的两个区域,保存它们的连通性,在跟踪触摸手指的阶段,再确定其对应的手指关系.

2.2.1 背景模型

尽管红外摄像机安装在多点触摸桌面内部,但是其也受到环境中红外线的干扰,如阳光,日光灯中都有很强的红外线,四角的红外线激光灯也会在图像上形成一个亮度区域.而环境中的噪声经过桌面半透明表面会衰减变化较慢,我们采用单一高斯背景模型^[10]:即对一个背景图像,特定像素亮度的分布满足高斯分布,即对背景图像点 $B(x,y)$ 的亮度满足

$$B(x,y) \sim N(u(x,y), d(x,y)) \quad (1)$$

为背景每个像素点 $B(x,y)$ 保存两个参数:平均值 $u(x,y)$ 和方差 $d(x,y)$,对于从红外摄像机获取的图像上每个像素点 $I(x,y)$.如果

$$\text{Exp}(-(I(x,y)-u(x,y))^2/(2*d(x,y)^2)) > T \quad (2)$$

则认为 $I(x,y)$ 为背景,反之为前景, T 为背景概率的阈值,通过多次统计背景像素得到.同时由于背景时刻在变化,不断更新每个像素的参数:

$$u_{t+1}(x,y) = a * u_t(x,y) + (1-a) * I(x,y) \quad (3)$$

a 为更新参数,表示背景变化的速度.统计到 d 变化很小,一般情况下不更新 d .

2.2.2 检测触摸区域

注意到图像上每个手指触摸区域内都存在一个或者多个灰度极值点,以每个极值点为种子,填充其八邻域灰度前景像素,便可以得到极值点所在的区域,每个检测到的区域都是一个可能的候选手指触摸区域.检测触摸区域的算法如下:

算法 1. 检测出触摸区域.

输入:图像的极值点.

输出:检测到的触摸区域树列表.

步骤:

S1: 初始极值点为一个区域,该点作为区域中心;

S2: 依次遍历每个区域边界像素的八邻域,若其为前景像素并且没有被加入到区域则加入到该区域中,若其已经加入到其他区域则转到 S3,否则重复 S2,若所有区域没有新像素加入转到 S4;

S3: 若这两个区域的面积小于预定义最小触摸区域面积或者这两个区域的距离小于预定义最小区域距离,则将两个区域合并成一个新的区域,加入区域链表中,否则标记这两个区域连通.返回 S2;

S4: 对相互连通的区域建立一个区域树,触摸区域作为叶子节点,所有连通区域的并集作为根节点.返回区域树列表.

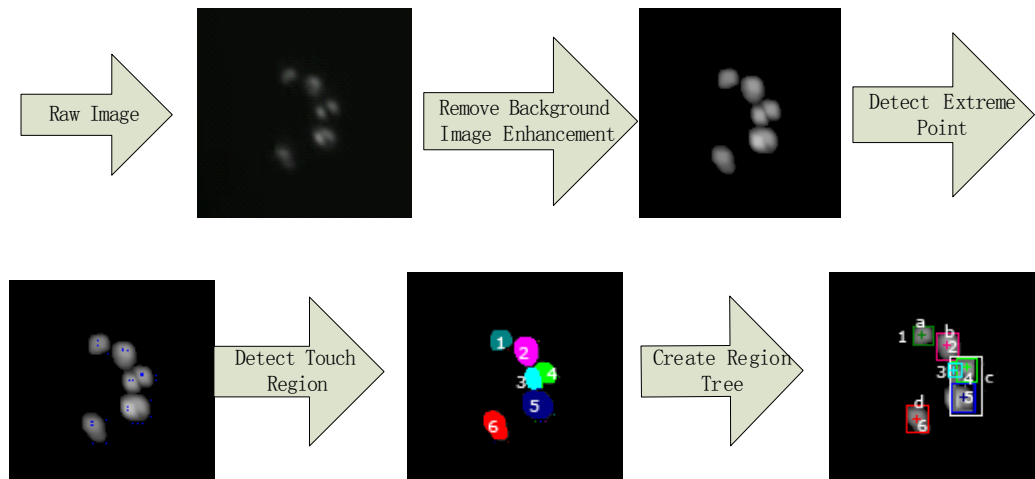


Fig.2 Detect touch region

图 2 检测触摸区域

具体流程如图 2 所示,首先对摄像机原始图像去除背景,图像增强,然后遍历图像的每个前景像素,若其像素值大于其八邻域的像素,则视为一个极值点,然后以极值点为中心,检测触摸区域.在所采用的触摸桌面上,每个手指最小包含 30 个像素,两个区域间距离最小也设为 30 个像素.为了保存区域的间的连通区域,将检测到的区域保存为区域树结构.如图 2 在检测到 1,2,3,4,5,6 区域后,建立 a,b,c,d 四棵区域树,其中 c 是由相互连通的 3,4,5 区域组成,其他则由单一区域组成,为了快速查找区域对触摸区域建立空间索引^[11].

2.3 跟踪触摸手指

检测到当前帧的手指触摸区域后,需要和正在跟踪的触摸手指匹配,确定触摸手指在当前帧的位置.对于区域断裂和重叠的问题,首先为手指运动建模,然后分两步跟踪触摸手指,第一步将区域树的所有叶子节点作为候选区域和跟踪手指计算最大权二分匹配,尽可能的为每个正在跟踪的手指确定一个匹配的区域,第二步将叶子节点都没被匹配的区域树的根节点,作为一个新的跟踪手指加入跟踪列表,同时将没有匹配的跟踪手指点移除跟踪列表.

2.3.1 手指运动模型

经过的统计触摸手指在触摸桌面的运动是平缓的,一些跟踪算法^[5-7]对每个手指建立卡尔曼滤波器^[12]预测其状态,而如果手指很多,这种方法需要消耗很大的资源和系统时间,降低系统效率.基于系统帧速率恒定,假定手指在前后两帧的运动相同,设触摸手指在第 $(\hat{x}_{i+1}, \hat{y}_{i+1}) = (x_i, y_i) + (dx_i, dy_i)$ 帧的位置为 (x_{i-1}, y_{i-1}) ,第 i 帧位置为 (x_i, y_i) 则手指在第 i 帧的速度 (dx_i, dy_i) 为

$$(dx_i, dy_i) = (x_i, y_i) - (x_{i-1}, y_{i-1}) \quad (4)$$

触摸手指在第 $i+1$ 帧的预测位置为

$$(\hat{x}_{i+1}, \hat{y}_{i+1}) = (x_i, y_i) + (dx_i, dy_i) \quad (5)$$

考虑到预测会存在一定误差,在 $(\hat{x}_{i+1}, \hat{y}_{i+1})$ 为中心,长宽分别为:

$$width = R + \alpha |dx|, height = R + \alpha |dy| \quad (6)$$

的矩形内部搜索触摸手指的候选匹配区域, R, α 是默认的调整矩形大小的参数,通过实验获得.称这个矩形为跟踪手指的搜索窗口.

2.3.2 匹配跟踪手指和候选区域

一般的工具包只基于跟踪手指和候选区域的欧氏距离来确定相似度^[5-7].我们根据提出的手指模型,结合手指运动和欧式距离来确定相似度.定义跟踪手指 f 和候选区域 r 的相似度为

$$s(f, r) = k \sqrt{d^2 + |v|^2 - 2 \times \cos \theta \times d \times v} + d \quad (7)$$

d 是 f 和 r 的欧式距离, v 是 f 的速度, θ 为 f, r 组成的向量和速度 v 的夹角.根据余弦定理, $\sqrt{d^2 + |v|^2 - 2 \times \cos \theta \times d \times v}$ 是 f 到 r 的向量和速度 v 组成的三角形的第三边的长度,表示 r 距 f 预测位置的偏差程度, f 和 r 距离越近 f 到 r 的向量与速度 v 越相近, $s(f, r)$ 越小, r 越可能为跟踪手指 f 对应的触摸区域. k 为一个比例参数,用来调节预测偏差的影响.设当前帧跟踪手指为 f_1, f_2, \dots, f_n , 检测到的触摸区域为 r_1, r_2, \dots, r_m .跟踪所要解决的问题是对于每个 f_i 确定唯一的 r_j , 并且每个 r_j 唯一对应一个 f_i , 使得

$$F = \max \sum 1/s(f_i, r_j), (1 \leq i \leq n, 1 \leq j \leq m) \quad (8)$$

这是一个二分图的最大权匹配问题,采用 Kuhn-Munkres 算法^[13]求解.

跟踪触摸手指的算法描述如下:

算法 2. 跟踪触摸手指.

输入:区域树列表和区域空间索引.

输出:新触摸手势列表,跟踪手指列表,消失手指列表.

步骤:

S1.对每个跟踪手指列表的手指对象,采用(6)式计算搜索窗口,在空间索引中查找窗口内的候选触摸区域,计算其与候选触摸区域的相似度,对不在窗口内的触摸区域设置它们相似度为无穷大.

S2.采用 Kuhn-Munkres 算法计算跟踪手指列表和候选区域的最大权二分匹配.

S3.对于每个匹配的跟踪手指和触摸区域,用触摸区域更新跟踪手指的位置和速度.对于没有匹配的跟踪手指从跟踪手指列表移除,加入到消失手指列表.对于任何叶子节点都没有被匹配的区域树,将其根节点作为一个新的触摸手指,加入到新触摸手指列表和跟踪手指列表.

对于连通的两个区域,一种情况是由两个触摸手指重叠而成的,这时跟踪列表中就会存在和这两个区域匹

配跟踪手指,它们会一一匹配.另外一种情况是由一个触摸手指断裂而成的,这是若跟踪列表存在和其中一个区域匹配的节点,则其中一个区域会和跟踪手指点匹配,另一个区域会被忽略,若跟踪列表不存在和它们匹配的跟踪手指,则将这两个区域的并集也就是它们对应区域树的根节点,作为一个新的跟踪手指.我们首先尽可能的为每个正在跟踪的手指确定一个匹配的区域,保证重叠的触摸手指能够找到匹配的区域,其次没有匹配的相互连通区域作为一个触摸手指,准确处理手指区域断裂的情况.这样结合区域连通信息和前后帧信息就能确定跟踪手指和检测区域匹配情况,解决手指重叠和遮挡的问题.

2.4 校正

由于摄像机拍摄的图像并不会完全与桌面重合,在摄像机内部也存在径向畸变和切向畸变,因此需要将手指对象在图像的位置映射到手指在桌面位置.由于图像从桌面到摄像机的传感器成像经过的是一个单应矩阵的变换过程^[14],设桌面上的点 (x_s, y_s) 在摄像机无畸变的情况下,映射到图像上 (x_d, y_d) .满足下式:

$$s \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} = M_{3 \times 3} \begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} \quad (9)$$

式中 s 为比例因子, $M_{3 \times 3}$ 为单应矩阵.由于摄像机畸变的存在, (x_d, y_d) 在图像上畸变到 (x_u, y_u) .这里只考虑径向畸变,满足摄像机畸变模型^[15]:

$$x_u = x_d + (x_d - x_c)(k_1 r^2 + k_2 r^4), y_u = y_d + (y_d - y_c)(k_1 r^2 + k_2 r^4) \quad (10)$$

式中 (x_c, y_c) 是图像的中心点, $r = \sqrt{(x_d - x_c)(x_d - x_c) + (y_d - y_c)(y_d - y_c)}$ 为 (x_d, y_d) 到中心点欧式距离. k_1, k_2 是畸变参数.因此,只要计算出多点触摸桌面设备对应的 $s, M_{3 \times 3}$ 和 k_1, k_2 ,便可以计算出图像的点对应的屏幕位置.由于有12个未知数,每个在桌面和图像中的对应点可以得到3个关于 $s, M_{3 \times 3}$ 参数的方程和2个关于 k_1, k_2 的方程,最少6个对应点便可以解出对应参数.但是由于采集点越多,参数计算越精确,我们采集20个对应点,采用最小二乘的方法计算参数.

在触摸跟踪后,对每个手指对象的位置、速度、最小包围矩形进行校正,使其准确对应到桌面位置.为了与具体分辨率无关,桌面大小无关,将坐标的范围缩放到0和1之间.

图3中网格画出的是采集的数据点,图中的两个白色矩形是两大大小为8开的书籍轮廓.从图3(a)中可以发现四角的红外激光灯已经严重偏离,书的边发生严重变形.在图3(b)中,图像已经过校正,灯已经恢复原来的矩形的位置,书则恢复了矩形形状.

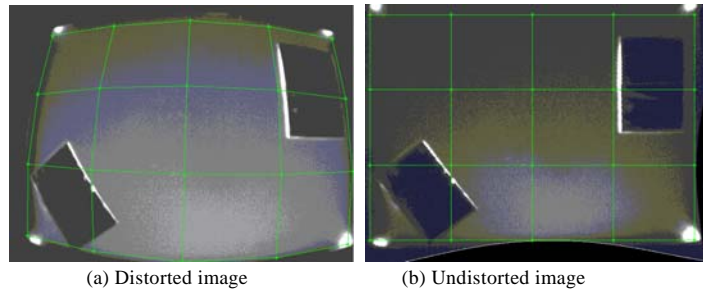


Fig.3 Calibration result

图3 校正结果

3 MTDriver 架构

我们将整个流程包装为MTDriver,以摄像机图像作为输入,先对输入图像做图像预处理,然后检测触摸区域,建立区域树和区域的空间索引,接着跟踪触摸手指和校正,最后将校正后的触摸手指信息包装成触摸消息发

送到应用程序中.系统的数据流程图如图 4 所示.

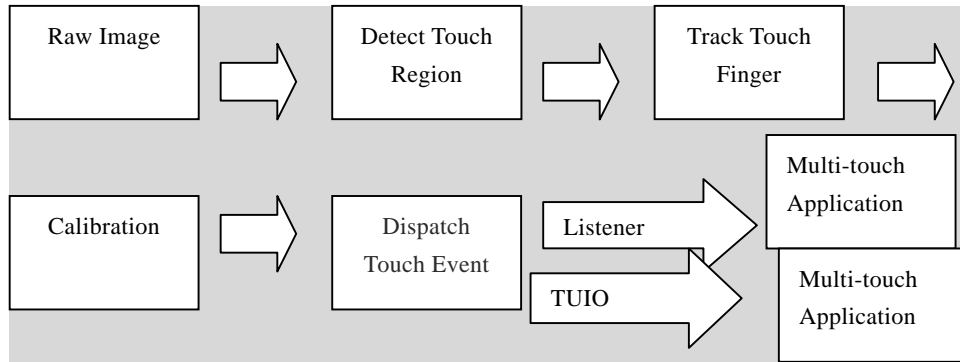


Fig.4 Data flow of MTDriver

图 4 触摸工具包数据流程

将触摸事件分为 3 种事件:TOUCH_DOWN(手指按下),TOUCH_UPDATE(手指更新),TOUCH_UP(手指抬起).然后将新触摸手指对象链表的每个对象发送 TOUCH_DOWN 事件消息,将跟踪列表的每个对象发送 TOUCH_UPDATE 事件消息,将跟踪丢失对象链表的每个对象发送 TOUCH_UP 事件消息.

为了便于和应用程序进行交互,MTDriver 采用两种方式发送消息.第一种采用监听器的方式,应用程序注册监听器到 MTDriver 中.在触摸事件发生时 MTDriver 调用注册的监听器,传递触摸消息给应用程序.这种方法简单直接,效率高,但是这种方式必须保证提供监听器,编程环境有限.第二种方式采用 TUIO^[16]协议的方式发送消息.TUIO 的协议在采用 OSC^[17]协议传输,时延小,传输可靠,支持各种平台.MTDriver 系统发送标准的 TUIO 1.1 规范的消息,应用程序采用对应的 TUIO 客户端,便可以收到触摸消息,借此,MTDriver 可以和其他任意语言程序交互,支持在各种环境开发的多点触摸应用程序.

4 实验

4.1 性能测试

对 touchlib,CCV,MTDriver 进行了对比测试.测试系统的硬件配置为 Intel Core2 Quad Q9450 2.55GHz,2GB 内存,320G 硬盘空间.操作系统为 Windows XP.在测试软件选择上,采用每个工具包提供的 OSC 服务端程序发送触摸消息,客户端采用标准实现的 TUIO DEMO^[18]统计结果.采用相同的客户端程序,目的在于只关注于工具包的性能.

首先测试每个工具包的效率和内存占用.我们测试了各个工具包在触摸手指个数为 0,10,20,30 情况下,每帧所需要的时间和程序所占用的内存.占用内存是通过资源管理器测得,每帧所占用的时间是从获取图像开始到分发触摸事件结束后所经历的时间.测试结果如图 5 所示.

从图 5 可以看出 CCV 占用的内存较多,touchlib 和 MTDriver 占用内存较小,由于 CCV 的 OSC 服务端采用图形界面实现,因此所占内存较多.但是随着手指的增加,各个工具包占用的内存都成线性缓慢增长,说明各个工具包的空间复杂度相似.在时间复杂度上 touchlib,CCV 每帧所用的时间最多,由于 touchlib 采用全局查找的跟踪算法,手指增多,其所耗时间增加很快.而 CCV 采用 KNN 的搜索算法,手指数量增多对其影响不大.而 MTDriver 基于区域查找,跟踪时利用空间索引树,只查找搜索窗口内的区域节点,效率高,同时手指增多所要遍历的像素也越多,计算的时间相对变长.总体来说当桌面手指个数在 30 以下时,三个工具包的平均时间都在 20ms 以下,可以保证每秒 30 帧的帧速率.

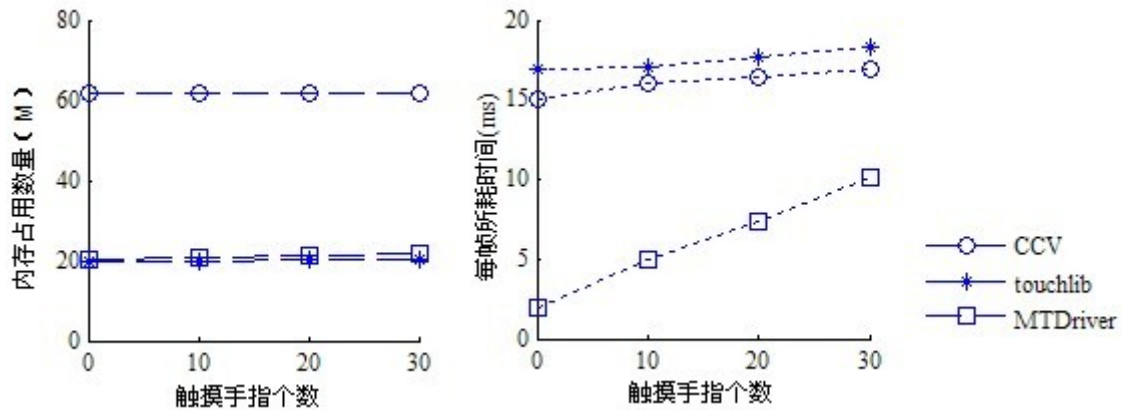


Fig.5 Test efficiency of MTDriver

图 5 系统效率测试

其次测试各个工具包检测准确度.测试的方法是统计触摸手指个数分别为 1,3,5,10 时,工具包检测准确的手指个数.为了避免单次的测试误差,取 10 次的平均结果作为测试的结果.同时我们也测试了在测试桌面上每个工具包所能检测手指间的最小距离.测试结果统计见表 1.

Table 1 Detect touch finger result

表 1 检测结果统计表

	1	3	5	10	最小检测距离
CCV	1	3	4	8	3cm
touchlib	1	3	3	7	4cm
MTDriver	1	3	5	9	2cm

在表 1 中,由于 CCV,touchlib 都是基于图像处理的方法,所以检测准确率低,最小检测距离大.CCV 通过交互设置参数,相对于 touchlib 获得了高一些的检测率.MTDriver 采用极值点检测触摸区域的方法,准确的检测到了每个手指.

再次进行跟踪效果测试.TUIO DEMO 会将跟踪到的手指轨迹画在屏幕上,分别用 1,2,3,5 个手指画 S 型曲线,截取每个工具包的手指轨迹图和 5 个手指时屏幕显示图像,统计在表 2.

从表格 2 中可以看出在手指个数少的情况下,各个工具包都跟踪的很好.当手指增多时 CCV,touchlib 得到的手指轨迹均有不同程度断裂,touchlib 在 5 手指时还出现了手指丢失的情况,只能识别出三个手指,MTDriver 可以准确的跟踪痕迹,即使手指交叉时仍然得到完整的轨迹.综合测试结果,MTDriver 识别和跟踪准确,结果优于 touchlib 和 CCV,具有良好的性能.

4.2 应用举例

我们通过多点触摸地球程序 MTEarth 的开发过程来验证 MTDriver 的可用性.MTEarth 需求是实现用多点触摸来操作 Google Earth,实现旋转、缩放、平移基本的浏览功能.采用 C#调用 Google Earth API^[19]的方式实现对 Google Earth 的调用.采用 C# TUIO 客户端实现^[18]响应 MTDriver 发出的触摸消息.MTEarth 将监听器注册到 TUIO 客户端中,对手指触摸事件作相应处理,调用 Google Earth API 显示界面.

采用简单的手势来操作地球浏览,对 TOUCH_UPDATE 事件分析,当单手指移动时,按照手指移动的方向和距离,平移地图.当双手指靠近时,按照手指靠近的距离来缩小地图,当双手指彼此分离或者手指双击时,放大地图.当双手指彼此旋转时,旋转地图.同时加入了小窗口,满足多人同时浏览的需求,每个窗口单独进行手势处理,使得不同窗口不会互相干扰.系统界面如图 6 所示.

Table 2 Track touch finger result

表 2 跟踪触摸手指统计表

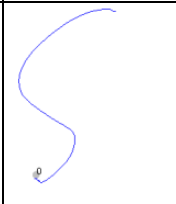


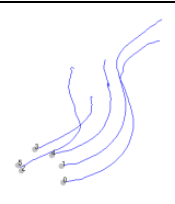

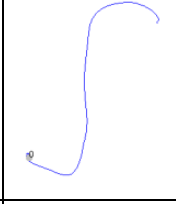
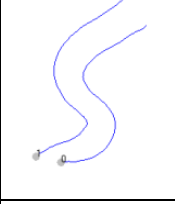



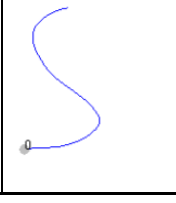
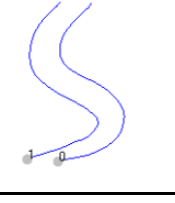

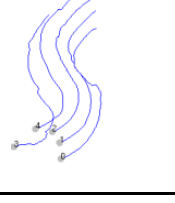

	1 个手指	2 个手指	3 个手指	5 个手指	手指检测图
CCV					
touchlib					
MTDriver					



Fig.6 MTEarth

图 6 MTEarth 界面图

通过 TUIO 接口,应用程序可以轻易的获得多点触摸消息,响应触摸事件,实现多点触摸用户界面.借助于 MTDriver,多点触摸应用程序可以用任意编程环境实现,应用开发难度降低.这将有助于开发者开发更多的多点触摸应用程序,丰富多点触摸桌面应用.

5 总 结

本文提出了一种改进的多点触摸工具包 MTDriver,提出基于极值点提取区域并结合相邻帧信息的手指识别方法,提出基于结合手指运动和欧式距离的匹配函数,采用空间索引树和搜索窗口查找的跟踪算法,采用摄像机畸变校正算法,能够准确的识别和跟踪多点触摸桌面上的触摸手指.通过对比测试,MTDriver 具有良好的性能,MTDriver 提供灵活的编程接口,简化了多点触摸程序的开发.基于 MTDriver 可以快速搭建低成本的多点触摸交互桌面,快速开发满足各个场景的多点触摸交互程序.

致谢 在此,我们向对本文的工作给予支持和建议的同学和老师,尤其是中国科学院软件研究所人机交互与智

能信息处理实验室的成员们表示感谢.

References:

- [1] Microsoft.Surface.2006.<http://www.microsoft.com/surface/en/us/default.aspx>
- [2] Sony. atracTable.2009.http://www.atracsys.com/_products/atracTable.php
- [3] Han JY. Low-Cost multi-touch sensing through frustrated total internal reflection. Symposium on user interface software and technology. 2005. 115–118.
- [4] Schöning J, Brandl P, *et al.* Multi-Touch surfaces: A technical guide. university of münster. Institute for Geoinformatics. 2008.
- [5] Kaltenbrunner M, Bencina R. reacTIVision: A computer-vision framework for table-based tangible interaction. Tangible and Embedded Interaction. 2007. 69–74.
- [6] Jordà S. The reacTable: Exploring the synergy between live music performance and tabletop tangible interfaces. Tangible and Embedded Interaction. 2007. 139–146.
- [7] NUIgroup. touchlib. 2007. <http://nuigroup.com/touchlib>
- [8] NUIgroup. CCV. 2007. <http://nuicode.com/projects/tbeta>
- [9] Cheng HD, Lui YM. Automatic bandwidth selection of fuzzy membership functions. Information Science, 1997,103:1–21.
- [10] Stauffer C, Grimson WEL. Adaptive background mixture models for realtime tracking. In: Proc. of the IEEE Int'l Conf. on Computer Vision and Pattern Recognition. IEEE Computer Society, 1999,2:246–252.
- [11] Guttman A. R-trees: A dynamic index structure for spatial searching. International Conf. on Management of Data. 1984. 47–57.
- [12] Kalman RE. A new approach to linear filtering and prediction problems. ASCE Journal of Basic Engineering, 1960,82D(1):35–45.
- [13] Munkres J. Algorithms for the assignment and transportation problems. Journal of the Society for Industrial and Applied, 1957,5(1):32–38.
- [14] Richard Hartley, Andrew Zisserman. Multiple View Geometry in computer vision. Cambridge University Press .ISBN 0-521-54051-8.2003. 32–33.
- [15] Weng JY, Cohen P, Herniou M. Camera calibration with distortion models and accuracy evaluation. IEEE Trans. on Pattern Analysis and Machine Intelligence, 1992,14(10):965–980.
- [16] Kaltenbrunner M, Bovermann T, Bencina R, Costanza E. TUIO: A protocol for table-top tangible user interfaces. In: Proc. of the 6th Int'l Gesture Workshop. 2005.
- [17] Wright M, Freed A. Open sound control: A new protocol for communicating with sound synthesizers. In: Proc. of the Int'l Computer Music Conf. 1997.
- [18] TUIO. TUIO Client Reference Implementations.2009.<http://www.tuio.org/?software>
- [19] Google. Goolge Earth. <http://earth.google.com/>



谭国富(1986—),男,河北承德人,硕士,主要研究领域为多点触摸,计算机视觉.



刘佳升(1987—),女,博士,主要研究领域为多点触摸交互用户界面.



戴志军(1981—),男,博士,助理研究员,主要研究领域为人机交互,增强现实,计算机视觉.



张凤军(1971—),男,博士,副研究员,主要研究领域为人机交互,虚拟现实,CAD.



王宏安(1963—),男,博士,研究员,博士生导师.主要研究领域为实时智能,用户界面.



戴国忠(1944—),男,研究员,博士生导师,CCF高级会员,主要研究领域为人机交互,计算机图形学.