

# 基于深度信息的实时手势识别和虚拟书写系统

黄晓林, 董洪伟

HUANG Xiaolin, DONG Hongwei

江南大学 物联网工程学院 江苏 无锡 214122

College of Internet of Things, Jiangnan University, Wuxi, Jiangsu, 214122, China

**Depth-based real-time hand gesture recognition and virtual writing systems. Computer Engineering and Applications**

**Abstract:** Inspired by the successful application of the contactless somatosensory interaction technology in the field of human-computer interaction, this paper presents a real-time Kinect depth camera based contactless virtual hand writing technique. In our technique, hand regions are first detected and segmented utilizing both color and depth information. Then hand fingers are rapidly recognized using a modified scan conversion method of a circle in order to recognize different hand gestures. Finally, using tracking by detection of a fingertip to obtain the fingertip trajectory and segment out individual trajectories for English and Chinese characters. Further a random forests algorithm is used to identify the correct characters from these individual trajectories. This depth information based hand gesture recognition and virtual hand writing technique can overcome the effects of light and color overlap and make the real-time detection and identification of gestures and hand writing more robust and reliable. The recognition accuracy can arrive at 93.25% and the speed of the recognition is about 25fps.

**Key words:** depth image, hand gesture recognition, hand writing recognition, random forests

**摘要:** 鉴于无接触体感交互技术在人机交互领域的成功应用, 本文提出了一种基于 Kinect 深度相机的实时隔空虚拟书写方法. 首先结合颜色和深度数据检测和分割出手掌区域; 进一步, 通过修改的圆扫描转换算法获得手指的个数, 以识别不同的手势指令; 最后根据指尖检测从指尖的运动轨迹分割出独立的字符或汉字运动轨迹, 并采用随机森林算法识别该字符或汉字. 这种基于深度信息的手势检测和虚拟书写方法可以克服光照和肤色重叠的影响, 可靠实时地检测和识别手势和隔空书写的文字, 其识别率达到 93.25%, 识别速度达到 25 帧/s。

**关键词:** 深度图像, 手势识别, 手写识别, 随机森林

doi:10.3778/j.issn.1002-8331.1308-0096 文献标志码: A 中图分类号: TP391.4

## 1相关工作

作为一种不同于键盘和鼠标的更加自然直观的人机交互技术, 基于计算机视觉的手势人机交互技术是实现新一代自然人机交互不可或缺的关键技术之一, 来自计算机视觉、图像处

理、人工智能等领域的许多国内外学者从手(势)的检测、分割、跟踪与识别等多个相关问题进行了许多深入的研究。

文[1-2]采用人脸检测中的Haar cascades方

**基金项目:** 国家大学生创新创业训练计划(No.121029579)。

**作者简介:** 黄晓林(1992-),女,本科,研究领域为计算机图形学与计算机视觉;董洪伟(1967-),男,博士,副教授,研究领域为计算机图形学、计算机视觉、高性能计算、数控技术。E-mail: hxlsmlingbluesky@126.com

法研究人手的检测问题,然而与人脸不同,人手自由度高达26,且光照、阴影、背景等的复杂变化使得人手的RGB图像的变化是无限的,使得该方法难以获得令人满意的结果。而实际上,尽管计算机视觉库OpenCV早已包含了Haar cascades方法的人脸、人眼、人嘴等检测功能,但并没有人手检测功能。[3]提出基于纹理和轮廓的多线索实时人手跟踪方法,首先基于轮廓和曲率检测人手区域,然后结合速率权特征和颜色对手进行跟踪。文[4]提出一种基于“运动掩码”的改进Camshift算法,综合利用合颜色、运动概率分布和概率简化跟踪多个人手。[5]提出了基于自适应肤色分割和Haarlets小波的实时人手检测方法,其皮肤颜色模型根据检测的人脸肤色实时地更新,可以有效地处理与肤色相近的衣料颜色、红色或棕色物体,然而仍然不能处理人手与皮肤重叠或包含背景人脸等场景。可见单纯依赖肤色检测人手是一个非常困难的问题。针对三维造型、动画和虚拟现实等特定应用问题,为了准确可靠地检测、跟踪人手及其关节运动,文[6]通过人手上套上一个特殊制作的彩色手套,简化了手的检测和姿态估

计,他们采集了18000个手指配置数据库,并采用简单的最近邻方法跟踪和重建人手配置的变化。

为克服单纯依赖颜色信息检测和跟踪人手的限制,文[7]用立体相机获取人手的3D数据,提出基于模型的隐马尔可夫模型Hidden Markov Models (HMM)的手势跟踪和识别方法。[8]采用颜色和深度信息检测和跟踪人手,首先根据高斯肤色模型和直方图肤色模型混合的肤色模型检测人脸和人手,并根据深度信息确定人手区域,进一步,结合颜色和深度数据,应用基于Haarlet小波的手势识别方法识别手势。

近年来,受到Kinect为代表的体感交互技术的成功应用的鼓舞,由于Kinect具有成本低、实时性高等优点,许多学者开始研究基于Kinect的手势交互技术。文[9-10]首先基于皮肤颜色和k-means聚类分割人手,然后计算人手凸包和确定手的轮廓,进一步检测手指及其方向,最后根据检测的手掌、手指及其方向、手指数目确定手势类型。文[11]通过对从深度图像中检测出的人手建立表观特征,并用决策树识别手势。

本文提出结合颜色和深度图像的指尖跟踪

和手势识别的隔空手写识别技术，如图1所示，学习模型即可在线地识别用户的手势轨迹。主要工作可分为如下三部分：

1) 结合颜色和深度图像的人手检测和分割，首先利用肤色模型检测潜在的人手，然后结合深度信息得到离kinect最近的距离，进而对两个条件相与得到手掌区域。

2) 利用手指数目及指尖检测的手势识别，首先利用图像形态学操作确定掌心位置，然后采用圆扫描转换算法快速检测手指数目及指尖位置。

3) 根据指尖运动轨迹的隔空手写识别，采用随机森林方法对指尖轨迹进行学习训练决策树森林，根据训练的决策树森林对手指描绘的隔空文字轨迹进行识别。

2. 系统流程

本文设计实现了基于kinect深度信息的手势识别和虚拟书写系统,如图1 所示. 首先系统识别出手势指令，然后为轨迹集中的每个动作采集样本，系统通过旋转变化的方法自动生成训练集，可很大程度上减轻用户采集训练样本的劳动强度；经过预处理后，这些样本被用于离线的训练机器学习模型；经过训练的机器

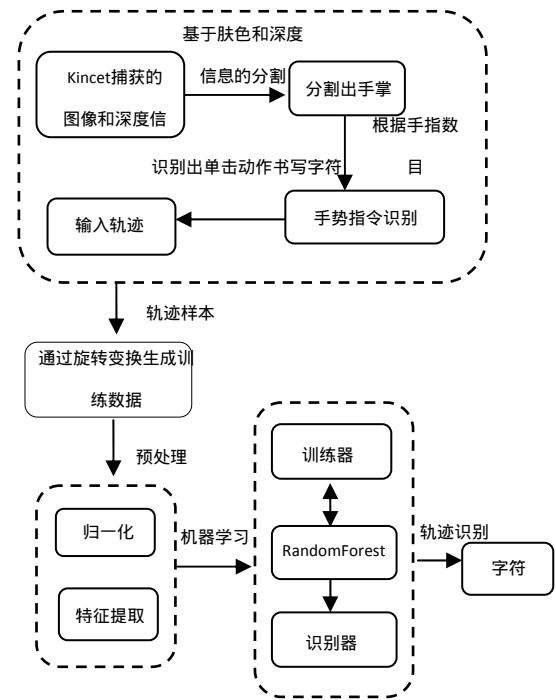


图1 系统流程图

3. 实现原理和步骤

3.1 结合肤色模型和深度数据的人手检测与分割

尽管基于肤色的人皮肤区域如人脸检测已得到广泛研究，但光照、背景等因素的影响，使得我们很难设计出一个健壮的皮肤检测算法，结合深度信息可以有效地解决肤色模型在肤色检测中的局限性。如图2所示，首先利用肤色模型检测潜在的人手，然后再根据深度信息得到离kinect最近的手掌区域，最后应用图像形态化运算(操作)得到光滑的手掌区域。

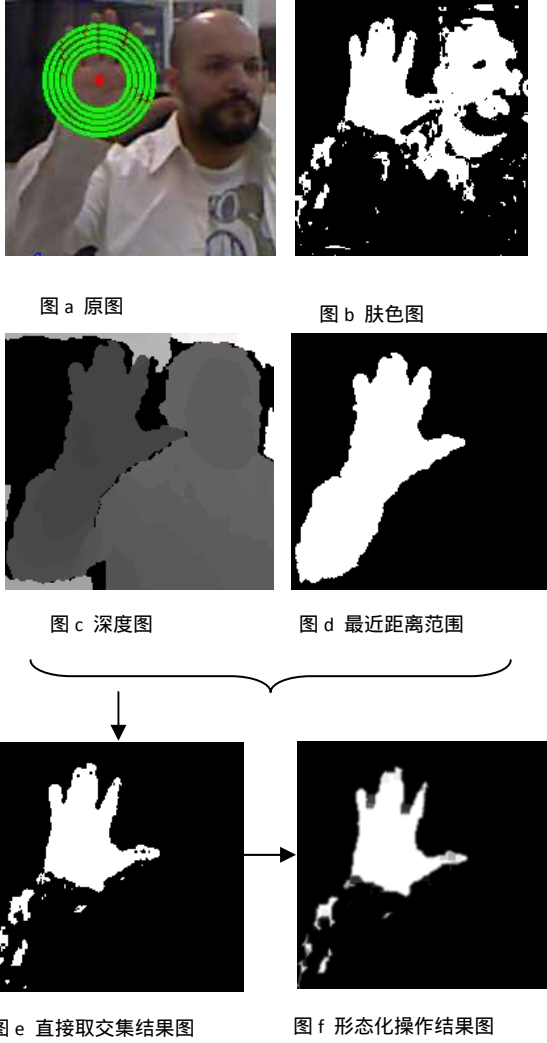


图2. 结合肤色模型和深度数据的人手检测与分割

### 3.1.1 肤色区域检测

本文通过对肤色高斯建模方法来检测可能的皮肤区域, 首先通过采集不同的人手采样, 根据这些采样的皮肤像素颜色信息, 构造一个肤色高斯模型, 为了尽可能避免光照的影响, 在学习该高斯模型前, 对每个皮肤像素的rgb值进行光补偿预处理. 设一个像素点p的rgb值为 $r_p, g_p, b_p$ , 修改后的 $r_p, g_p, b_p$ 为:

$$r_p = \frac{r_p}{\max(r, g, b)}; \quad g_p = \frac{g_p}{\max(r, g, b)}; \quad b_p =$$

$$\frac{b_p}{\max(r, g, b)} \quad (1)$$

其中 $\max(r, g, b)$ 是所有皮肤像素的红色、绿色、蓝色值的最大值。

在建立肤色模型之前我们首先选取大量彩色图片进行处理只留人体皮肤区域, 接下来将彩色图片颜色空间转换到YCbCr空间, 然后对该区域的CbCr值进行统计处理。处理的方法采用如下的高斯模型:

$$m = E(x), \quad x = (Cb, Cr)^T \quad (2)$$

协方差矩阵C

$$C = E\{(x - m)(x - m)^T\} \quad (3)$$

通过肤色高斯模型, 我们可以得到待检测

彩色图像中任意一个像素点属于皮肤的概率。

对于某像素点s, 从RGB空间转换到YCbCr色彩空间得到色度值(Cb, Cr)则该像素的肤色密度可由下式得到:

$$p(Cb, Cr) = \exp[-0.5(x - m)^T C^{-1}(x - m)] \quad (4)$$

其中 $x = (Cb, Cr)^T$ ,  $C^{-1}$ 为C的逆矩阵。

根据(Cb, Cr)值通过式(4)找到该值所对应的肤色似然度, 并且得到整幅图像的最大肤色似然度。每一点像素的肤色似然度除以最大肤色似然度所得到的值, 作为该像素点的灰度值, 从而得到肤色似然度图像, 其中每一个像素点的灰度值表征了这个像素点属于皮肤的概率。

最后我们选择了皮肤概率大于80%的作为肤色。因而确定一幅彩色图像中潜在的肤色区域。

### 3.1.2 基于深度图像的人手检测

基于肤色模型只能确定可能的皮肤区域，为了确定手掌区域，本文进一步根据深度信息检测人手位置，考虑到实际隔空书写时，人手是距离相机最近的，我们认为离kinect最近的肤色区域为手势区域。为此，我们将整张深度图像(640\*480)分成100个矩形，在每个矩形(64\*48)随机选取8行，首先求得每行像素的深度平均值进一步求得这8行像素点的深度平均值作为该矩形的深度平均值，最后比较这100个矩形，得到平均深度最小值即最近距离，在该最近距离一定深度范围内的，被认为是手掌区域。设该最近距离为A，则手掌区域的深度位于区间 $[A, A+a]$ ，其中a是一个深度阈值，根据人手大小，可取为20mm。

对图像的矩形分割数目的取值很重要，我们发现如果分割图像的矩形个数偏小时，会造成手和背景划在一个矩形中而导致平均深度变大，最近距离不能得到保证；而矩形个数偏大时，一些噪声数据就会产生很大的影响，经过

多次的实验，我们在图片大小为640\*480的情况下将图片分成了100份，此时的手掌检测准确性和速度上较为理想。

### 3.1.3 手掌的检测和分割

肤色区域和人手区域的交集为手掌区域，由于噪声等因素的影响，采用这种交集分割出的手掌区域可能存在空洞，为此，我们采用简单的图像形态化操作，即先膨胀后腐蚀的方法，得到一个较为光滑的无空洞的手掌区域。从图2.e到图2.f可以看到手指处的空洞处较好的被处理掉了，手臂的残余部分也被消除很多，虽然还剩下部分残余，但可以观察到这些部分都比较细小，我们将在3.2中讲述如何处理这些部分。

图2.f为结合两者得到的手部区域，可以看到脸部被深度信息成功的分离，只剩下所需的手部

## 3.2 利用手指数目及指尖检测的手势识别

本文根据图像的轮廓信息识别出手指及指尖，以构建和识别手势指令，确定手指的运动轨迹，我们采用圆与手掌区域的相交信息来快速检测手指，为进一步提高算法的实时性，利

用改造的圆中点扫描转换算法, 在圆的扫描转换过程中就即时地确定圆与手掌区域的相交区段, 使程序效率得到极大的提高。

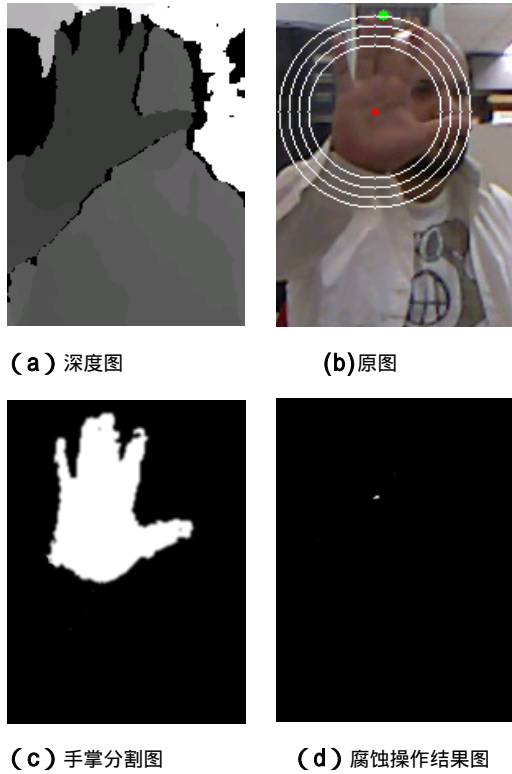


图3 利用手指数目及指尖检测的手势识别

### 掌心位置的确定

首先通过数学形态学中的腐蚀操作获取手势区域中心位置点。由于手掌作为手势表现的主要组成部分, 在手势区域它占有最大的面积而且像素点较为集中。通过连续的腐蚀操作, 可以消除手势区域的边界点, 使手势区域逐步缩小, 最终得到手势区域的中心位置点 $O$ , 实际效果如图3(d)所示。

### 基于圆扫描转换的手指检测

计算出中心点与手势区域边缘的最大距离值 $r$ , 对距离进行10等分, 每段的长度 $h=r/10$ 。

经过经验分析, 前六等份为手掌区域, 而我们只关心手指区域所以我们只取最后四个圆的轨迹, 对最后四个圆轨迹进行分析, 如图3(b) 所示。

按顺时针方向记录每条圆轨迹线上像素值变化点 $P_{ij}$ (0到1, 即从黑色区域到白色区域)和 $Q_{ij}$ (1到0, 即从白色区域到黑色区域)的位置坐标值,  $i$  表示轨迹圆的编号,  $j$  表示同一轨迹圆上 $P$ 或 $Q$ 点编号, 同时删除单独存在的 $P_{ij}$ 点和 $Q_{ij}$ 点

如图2.f), 由于分割的手掌区域存在噪声或包含手臂残余区域, 圆与这些区域的相交区段显然不属于手指, 为此, 我们采取中值滤波方法过滤掉这些不合理相交区段, 具体方案是:

首先通过 $P_{ij}$  和 $Q_{ij}$  的位置坐标, 然后计算每对 $P_{ij}$  和 $Q_{ij}$  之间的距离 $d_{ij}$ , 当轨迹圆与指尖部分区域相截时获得较小或者较大的 $d_{ij}$ , 不能表现手指的实际宽度值, 我们对某个圆与手掌区域的所有相交区段 $d_{ij}$ 进行排序, 然后确定其中值长度, 设为 $e$ , 然后我们计算所有长度 $d_{ij}$ 和中值的

残差平均, 设为 $e$ , 定义一个区间阈值

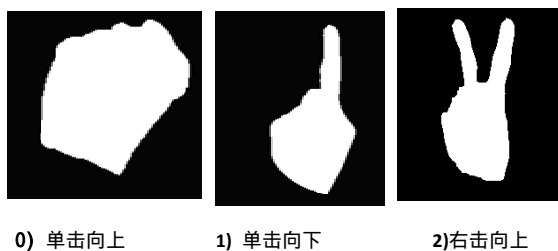
$=\min(1.2e, 0.2)$ , 只有当相交区段 $d_{ij}$ 在 $[e - \epsilon, e + \epsilon]$

+ ]时,才认为其是手指区段, 否则是其他区段。设计算每个轨迹圆上获得的手指数目为 $n_j$ , 所有轨迹圆得到的数目的最大值被认为是实际的手指数目, 即 $n = \max \{n_j | j = 1, 2, 3, 4\}$ 。

我们可以用手指数目的不同表示区分不同的手的形状, 我们既可以用一个手形表示某个手势, 也可以通过连续的两个手形表达更多复杂的手势或动作, 以模拟鼠标、键盘等动作或命令, 例如模拟鼠标的点击、键盘的击键(如F1,ESC)等动作。假如以两个连续手势表示某个动作, 最多可以表达模拟 $5 \times 4 = 20$ 个动作。

例如可以用于表示模拟鼠标的动作, 如图4所示, 其中 $N = 0, 1, 2, 3, 4$ 分别表示鼠标的单击向下、单击向上、右击向下、右击向上、双击, 0和1连续的手势表示单击动作, 2和3的连续的动作表示右击。也可以模拟键盘动作, 例如 $N = 5$ 表示ESC, 4和5连续动作表示

Backspace等



0) 单击向上

1) 单击向下

2) 右击向上



3) 右击向下

4) 双击

5) ESC

图4 基于手指数目的不同手指指令

具体在我们的隔空书写系统中, 我们定义了基本的指令集合: 图4.1)  $N = 1$ , 则认为是开始指令, 系统开始进入书写状态, 模拟鼠标单击向下; 图4.0)  $N = 0$ , 模拟鼠标单击向上, 作为一个字书写完毕的指令; 图4.5)  $N = 5$ , 模拟键盘的ESC键, 在程序运行的情况下, 识别出该手势后可正常退出。

### 获得指尖的位置

本文采用二分法的思想获得指尖的位置。首先由3.2节中获得手指与圆的交点, 由这些交点可以拟合出两条直线, 如图5(a), 得到最外层与圆相交的点命名为P,Q; 在距离 $d$ 范围内采用二分法, 每次在分割线上取 $2/5$ 处和 $3/5$ 处的点a,b. 分割线的初始位置为 $d/2$ , 若a,b点的深度值和PQ接近, 则分割线的位置为 $d' = d' + d'/2$ , 否则 $d' = d' - d'/2$ , 直到 $d' < 16/d$ 则停止如图5(b)。

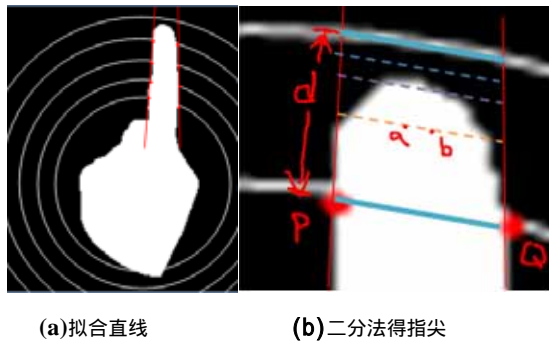


图 5 获得指尖的位置

### 3.3 根据指尖运动轨迹的隔空手写识别

指尖运动轨迹为系统检测到单击动作时, 鼠标沿着手指尖的移动而移动形成的轨迹。指尖运动轨迹具有类别多, 数据和特征空间处理量大等特性, 而随机森林在学习方面的表现非常优秀可以处理大量的数据和特征空间, 对于大量样本数据, 它可以产生高准确度的分类器, 所以我们选择了随机森林的方法。

隔空手写识别算法的主要模块包括离线的指尖运动轨迹训练和在线的指尖运动轨迹识别。经过预处理的指尖运动轨迹样本被用于训练机器学习模型, 一旦模型训练完毕, 即可用来实时地识别用户的指尖运动轨迹所表达的文字 (字母或汉字)。

#### 3.3.1 训练样本的采集

传统的基于机器学习的方法要求用户采集大量的训练样本, 但该过程枯燥乏味, 会大大

降低用户对基于隔空手写识别的交互系统的使用兴趣, 本文试图寻找并验证自动生成训练样本的方法, 模拟不同人的手写文字轨迹。在图1的系统流程图中提取出一个用户指尖运动轨迹样本后, 将该样本分别左旋和右旋, 从该采样样本可以自动生成一系列新的轨迹样本, 根据用户的书写习惯, 我们假设新轨迹相对于采样轨迹的旋转角度一般不超过45度, 且旋转角小的概率较大, 旋转角大的概率较小, 总体来说可看成一个高斯分布, 因此, 我们根据高斯分布采样旋转角度, 生成这些新的轨迹样本, 其中高斯分布的数学期望 $\mu$ 为 $0^\circ$ , 标准差 $\sigma$ 为 $15^\circ$ , 这样使得95%的轨迹样本偏离角度在 $[-30^\circ, 30^\circ]$ , 99%的轨迹样本偏离角度在 $[-45^\circ, 45^\circ]$ 。对于采集的每个轨迹样本, 系统都自动根据这种采样方式, 自动生成对应该样本的一系列样本, 这样, 我们只要采集少量的样本, 就可以自动获取大量的样本。

#### 3.3.2 数据的归一化处理

由于不同的人在屏幕输入的方位、幅度和速度都不同, 所以我们要对输入的数据进行归一化。首先得到轨迹的外接矩形, 取得外接矩



形的长度值 $w$ ，将其放大或者缩小 $500/w$ 倍，这样可以很好地控制轨迹的幅度，然后选择输入框的中心点作为样本的原点。考虑到速度的问题，可能会造成输入轨迹的疏密，为此，用直线段链接这些采样点轨迹，据此计算出轨迹的周长，然后采用均分分割方式进行重采样处理，保证轨迹点的个数为30个。

### 3.3.3 特征向量的提取

设每个轨迹经过归一化处理后的采样点坐标为 $(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots, (x_{30}, y_{30})$ ，按横、纵坐标组成2个30维的向量 $(x_1, x_2, x_3, \dots, x_{30})$ 和 $(y_1, y_2, y_3, \dots, y_{30})$ 。本文采用离散傅里叶变换(DFT)对这两个向量进行降维处理。

对于 $N$ 个复数： $x_0, x_2, x_3, \dots, x_{N-1}$ ，可按如下公式计算其离散傅里叶变换

$$f_k = \sum_{n=0}^{N-1} e^{(-i\frac{2\pi}{N}nk)} x_n, k = 0, \dots, N-1 \quad (5)$$

其中 $i$ 是虚数单位， $e$ 是自然对数的底数。由此得到 $fx_1, fx_2, fx_3, \dots, fx_{30}$ ，我们选取前5个作为特征数据的一部分，相同的 $fy_1, fy_2, fy_3, \dots, fy_{30}$ 也只取前5个，由此一个轨迹的特征向量的构成一个长度为10个实数的特征向量 $Fi[fx_1, fx_2, fx_3, \dots, fx_5, fy_1, fy_2, fy_3, \dots, fy_5]$ 。

### 3.3.4 分类器的训练

随机森林是由Amit和Geman[12]和之后的Breiman[13]引入的，可以被认为是一种Bagging分类的扩展。假设随机森林 $T$ 由 $N_t$ 个决策树 $\{T_i\}$ ， $i \in [0, N_t]$ 组成，每个决策树是根节点 $n_{root}$ ，分支节点 $n_{split}$ 和叶子节点 $n_{leaf}$ 的有序集合，根节点可以看成是一个特殊的分支节点，而每个分支节点可以看成是一个弱分类器。在训练生成随机森林前，由上文的3.3.3节生成的特征向量作为输入训练样本，对于任意的决策树 $T_i$ ，所有的训练样本加入该决策树的根节点。从根节点开始递归地将分支节点按照一定的分裂机制分裂成2个子节点（左子节点和右子节点），同时判定每个节点是否还可以继续分裂下去，如果符合终止条件，则该节点就变成一个叶子节点。对一棵树的某个节点，首先利用到达该节点的训练样本集合来计算该节点类别不纯度，如果类别不纯度大于0，则表示该节点还可以继续分裂，属于分支节点，如果该节点的类别不纯度为0，则该节点属于叶子节点。

使用信息熵来衡量节点的类别不纯度：

$$\text{Entropy}(I) = - \sum_{i=1}^m p(i, I) \log_2(p(i, I)) \quad (6)$$

其中， $I$ 是到达该节点的训练样本集合； $m$ 为类别的个数， $p(i, I)$ 是到达该分支节点的样本

中属于第*i*类的概率密度(比例), 可以将其定义为:

$$p(i, I) = \frac{|I_i|}{|I|} \quad (7)$$

其中,  $I_i$  是该节点中属于第*i*类的训练样本的个数;  $|I|$ 是该分支节点总样本数。

对待分类的训练集*S*用属性*A*分类以后可得到其系统总熵 $E(A)$ , 现在属性*A*是带有*V*个不同值的属性 $\{a_1, a_2, \dots, a_v\}$ , *A*可以把*S*分成*V*个子集 $\{S_1, S_2, \dots, S_v\}$ 其中 $S_i = \{x | x \in S \text{ \& } x.A = a_i\}$ 代表了从集合*S*的出发的所有树枝。设 $s_{ij}$ 表示在 $S_j$ 中类为 $C_i$ 的记录个数( $C_i$ 为第*i*个的类别,  $i \in [1, m]$ , 本文中为字符类别)。那么, 这时按*A*的每个属性值(更一般的是取*A*的一个子集)进行分割后的信息量, 也就是系统总熵为:

$$E(A) = \sum_{j=1}^v \left( \frac{s_{1j} + s_{2j} + \dots + s_{mj}}{s} \right) * \text{Entropy}(s_j) \quad (8)$$

信息增益为

$$\text{gain}(A) = \text{Entropy}(A) - E(A) \quad (9)$$

最后该节点的所有候选分裂属性中  $\text{gain}(A)$  最大的属性, 即  $A = \max_{A_i \in \Omega} \text{gain}(A_i)$  作为该节点的分裂属性。当所有的样本都落在一个节点即属于同一类或者达到了最大的深度, 则分裂结束该节点被标记为最大的分类。

### 3.3.5 手势的识别

随机森林是通过树来进行预测的, 而这些树又是依靠一个特征向量的值来进行预测的。这些具有相同分布的特征向量被无关地采样作为森林中每棵树的输入[13]。为了预测一个新的轨迹的分类, 让每棵树都对输入的特征向量 $F_i$ 进行投票, 计算所有的投票数, 找出票数最高的一个就是特征向量 $F_i$ 的分类标签。随机森林能计算出特征向量 $F_i$ 可预测出一个轨迹*c*的可信度

$$p[c] = \frac{1}{N} \sum_{j=1}^N \delta_{c(T_j(x))} \quad (10)$$

*N*是森林中树的个数,  $T_j(x)$ 是*x*落入第*j*棵树 $T_j$ 的叶子,  $\delta_c(a)$ 是克罗内克 $\delta$ 函数, 当 $c=a$ 时 $\delta_c(a)=1$ , 否则 $\delta_c(a)=0$ 。这样可以提供各种目标结果的可能性。在我们的系统中, 我们选择可信度最大的作为最终的目标分类。

## 4. 实验与讨论

### 4.1 训练样本的人工采样和自动采样的比较

为了验证我们训练数据的正确性我们做了如下的对比实验。我们请了10个用户, 实验一我们对测试轨迹集

$T\{A, B, C, D, E, F, G, H, I, J, K, L, M, N, W, X, \text{人, 也, 上},$

一,大}中的每个元素让用户表演十次作为训练样本集。实验二,我们让用户对每元素只表演一次,其余九次按本文3.3.1节原理由系统生成,产生训练样本集。最后任选30个用户,每个用户按照自己的书写习惯对T集合中的每个轨迹书写2遍,分别用实验一和实验二训练的训练模型对这些手写轨迹进行识别。



































































































































实验一中30个用户的1260次的测试结果中1182个轨迹可以识别,所以实验一人工采集样本的识别率为93.81%。实验二中1175个轨迹可以识别,所以使用系统经过旋转变换采样的样本识别率为93.25%。可以看出2个实验的手势识别率在统计学意义上相差不大,从而印证了本文假设的合理性。

4.2隔空手写轨迹的识别

为了验证本文方法的鲁棒性,我们要求30个用户对T集合中的每个轨迹故意做出速度和幅度差异比较大的动作5次并且记录系统识别率,采用的轨迹集仍为4.1节中的测试轨迹集T,需要注意的是这些手势必须一笔画完,中间不能有停顿。本系统采用Kinect通过USB 口与电脑进行数据通讯,电脑的配置是2.0GHz 的CPU 和

2G 的RAM,采样图片大小为640×480时,识别速度可以达到25 帧每秒,达到实时手势识别的要求。表1为30个用户中10名用户的部分字符轨迹图及识别率。

表1 部分用户轨迹识别率表

字符	用户1	用户2	用户3	用户4	用户5	用户6	用户7	用户8	用户9	用户10	识别率
A											94.00%
B											93.33%
C											90.67%
D											95.33%
E											90.00%
F											93.33%
G											94.67%
L											91.33%
X											92.00%
Y											94.67%
人											95.33%
也											93.33%
一											96.00%

从表1中可以看出C、E、L的识别率比较低,因为C和L在书写的时候比较相近,如用户7写的C和用户5写的L在顺序和形状都比较相近,所以系统的识别率下降;E的书写方法比较多,每个用户书写的顺序不同,而本文是按照归一化后轨迹点的坐标作为特征向量的所以需要足够多的不同书写方式的训练样本才能提高识别率。针对其他书写方式单一,简洁的轨迹来说系统的识别率就会较高。

由于本文在3.3.2节中对轨迹的缩放、旋转、速度都做了归一化处理所以本实验的识别率仍能达到92.59%。与运动传感驱动的3D直观手势交互系统[14]识别率相近,但本文具有更好的用户友好性、鲁棒性。

## 5. 结语

随着深度相机硬件技术的日益成熟,基于深度相机的3D体感交互等技术在近年来得到迅速发展,如何应用3D感知技术解决实际问题给研究人员提供了新的机遇和课题,不同于现有的触摸式手写识别技术,本文研究了基于深度相机的隔空书写技术,通过深度相机的颜色和深度信息快速分割人手并检测指尖运动,在此基础上,应用随机森林学习方法对隔空书写轨迹进行训练和识别,算法设计时充分考虑了实时性这一需求,无论是人手快速检测,还是基于圆扫描转换的指尖检测和轨迹跟踪,特别是随机森林算法的实时性特性,都保证了算法具有很好的实时性,实验结果表明,该算法可以用于隔空手写文字的实时识别。

目前本文算法只对少量文字进行了训练和识别以验证算法的可行性,我们将进一步采集

更多的文字书写轨迹用于训练随机森林模型,以识别更多的文字和提高随机森林模型的识别准确性。另外,本文算法没有对指尖轨迹进行光顺优化,我们也将进一步探讨如何对指尖轨迹进行时空优化。

## 参考文献:

- [1] Kölsch M, Turk M. Robust Hand Detection[C]//FGR, 2004, 614-619.
- [2] Ong E-J, Bowden R. A boosted classifier tree for hand shape detection[C]//Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on, 2004, IEEE, 889-894.
- [3] Araki R, Gohshi S, Ikenaga T. Real-time both hands tracking using CAMshift with motion mask and probability reduction by motion prediction[C]//Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific, 2012, IEEE, 1-4.
- [4] Pan Z, Li Y, Zhang M, et al. A real-time multi-cue hand tracking algorithm based on computer vision[C]//Virtual Reality Conference (VR), 2010 IEEE, 2010, IEEE, 219-222.
- [5] Van den Bergh M, Koller-Meier E, Bosché F, et al. Haarlet-based hand gesture recognition for 3D interaction[C]//Applications of Computer Vision (WACV), 2009 Workshop on, 2009, IEEE, 1-8.
- [6] Wang R Y, Popović J. Real-time hand-tracking with a color glove[C]//ACM Transactions on Graphics (TOG), 2009, ACM, 28:63.
- [7] Elmezain M, Al-Hamadi A, Appenrodt J, et al. A hidden markov model-based continuous gesture recognition system for hand motion trajectory[C]//Pattern Recognition, 2008. ICPR 2008. 19th International Conference on, 2008, IEEE, 1-4.
- [8] Van den Bergh M, Van Gool L. Combining RGB and ToF cameras for real-time 3D hand gesture interaction[C]//Applications of Computer Vision (WACV), 2011 IEEE Workshop on, 2011, IEEE, 66-72.

- [9] Li Y. Hand gesture recognition using Kinect[C]//*Software Engineering and Service Science (ICSESS), 2012 IEEE 3rd International Conference on, 2012, IEEE, 196-199.*
- [10] Yan W, Chuanyan H, Guanghui Y, *et al.* A robust method of detecting hand gestures using depth sensors[C]//*Haptic Audio Visual Environments and Games (HAVE), 2012 IEEE International Workshop on, 2012, 72-77.*
- [11] 曹雏清, 李瑞峰, 赵立军. 基于深度图像技术的手势识别方法[J]. *Computer Engineering*, 2012, 38(8): 16-18+21.
- [12] Amit Y, Geman D. Shape quantization and recognition with randomized trees.[J]. *Neural computation*, 1997, 9(7): 1545-1588.
- [13] Breiman L. Random forests.[J]. *Machine learning*, 2001, 45(1): 5-32.
- [14] 梁秀波, 张顺, 李启雷, *et al.* 运动传感驱动的3D直观手势交互[J]. *计算机辅助设计与图形学学报*, 2010, 22(3): 521-526.

## 基于深度信息的实时手势识别和虚拟书写系统

作者: [黄晓林, 董洪伟](#)  
作者单位: [江南大学 物联网工程学院 江苏 无锡 214122](#)  
刊名: [计算机工程与应用](#)  
英文刊名: [Computer Engineering and Applications](#)  
年, 卷(期): 2014

本文链接: [http://d.g.wanfangdata.com.cn/Periodical\\_pre\\_39a85f30-1890-43f0-a808-52069fe6e049.aspx](http://d.g.wanfangdata.com.cn/Periodical_pre_39a85f30-1890-43f0-a808-52069fe6e049.aspx)