

DOI: 10.19650/j.cnki.cjsi.J1905207

基于深度信息的指尖检测-跟踪-监督算法*

孟浩, 尹维考, 李洪进, 郭永新

(哈尔滨工程大学自动化学院 哈尔滨 150001)

摘要:针对快速准确识别手指并实时跟踪手指的问题,提出了基于 Kinect 深度信息的指尖检测-跟踪-监督算法。检测部分主要为利用深度数据实现指尖的识别,并结合指尖检测结果与卡尔曼滤波确定出指尖的稳定状态。跟踪部分主要为将检测部分得到指尖的稳定状态作为初始条件,并使用具有监督环节的快速判别尺寸空间跟踪算法实现对指尖的实时跟踪,在对每一帧的指尖运动实时跟踪的同时,监督环节对每一帧的指尖实时跟踪结果进行监督,通过与指尖的特征进行匹配,矫正跟踪结果发生偏差的点,来提高跟踪结果的准确性,从而保证整个系统具有较强的鲁棒性。本文对指尖检测识别与跟踪进行了多组实验,均能够准确的检测识别出指尖,并能稳定的实时跟踪指尖的位置。实验结果表明该方法具有较强的鲁棒性与准确性。

关键词: 深度信息;轮廓提取;凸包检测;卡尔曼滤波;监督环节

中图分类号: TP391.4 TP391.7 **文献标识码:** A **国家标准学科分类代码:** 510.8060

Fingertip detecting-tracking-supervising algorithm based on depth information

Meng Hao, Yin Weikao, Li Hongjin, Guo Yongxin

(College of Automation, Harbin Engineering University, Harbin 150001, China)

Abstract: To recognize and track fingertip quickly and accurately, one kind of fingertip detecting-tracking-supervising algorithm based on Kinect depth information is proposed. The detection process mainly uses the depth data to realize the fingertip recognition. The fingertip detection result is combined with Kalman filter to determine the steady state of the fingertip. The tracking process mainly takes the steady state of the fingertip in detection process as the initial condition. Fast discriminative scale space tracking algorithm is adopted as the supervising process to realize the real-time tracking of the fingertip. During tracking the movement of the fingertips of each frame in real time, the supervising process supervises the real-time tracking results of the fingertips in each frame. By matching the features of the fingertips to correct the deviation of the tracking results, the accuracy of the tracking results can be improved. Hence, the robustness of the whole system can be realized. In this paper, multiple sets of experiments have been implemented on fingertip detection, recognition and tracking. Both of them can accurately detect fingertips and track the movement of fingertips in real time. Experimental results show that the proposed method has strong robustness and accuracy.

Keywords: depth information; contour extraction; convex hull detection; Kalman filter; supervising part

0 引言

近年来,人机交互技术得到了迅猛的发展。常见的人机交互技术包括:语音、表情、手势等。而手势则被认为是自然交互技术之一^[1]。基于手势的交互技术的关键就是手势识别,而在手势识别的特征提取中,手掌区域与指尖是最常见的特征值,手指的检测识别与实时跟

踪在手势识别中不可或缺。在以前的研究中,手掌分割有许多方法。Keaton 等提出基于固定背景的背景减法来分割手掌区域^[2-3],该方法对背景的要求较高。Shahzad 等提出的基于肤色实现手掌分割^[4],该方法容易受衣物等的影响。指尖检测主要有 3 种方法:基于轮廓^[5-6],基于模板^[7-8],基于形状^[9-10]。叶植超提出双模式切换定位算法实时获取指尖位置信息^[11]。指尖的实时跟踪属于目标跟踪的范畴。早期的目标跟踪的方法有 mean-

收稿日期:2019-05-30 Received Date:2019-05-30

* 基金项目:智能船舶仿真验证评估技术研究项目资助

shift, 卡尔曼滤波等, 此类方法的跟踪速度较慢, 准确性有待提高。随着 David 将相关滤波 (correlation filters, CF) 的概念引入目标跟踪领域中^[12], CF 系列算法在目标跟踪领域表现突出。Henriques 等通过引入循环矩阵和核技巧, 提出了核相关滤波 (kernelized correlation filters, KCF)^[13], 进一步提高了目标跟踪的准确性。随后针对 KCF 没有解决的目标跟踪过程中的尺度变化问题, Danelljan 等提出了具有尺度自适应性的目标跟踪算法 (discriminative scale space tracking, DSST)^[14]。Danelljan 等又在 DSST 的基础上提出了快速判别尺寸空间跟踪 (fast DSST, FDSST)^[15]。

为了能准确快速的提取出手掌区域轮廓以及检测识别出指尖并能稳定跟踪指尖, 本文提出了一种基于深度信息的指尖检测-跟踪-监督算法 (detecting-tracking-supervising, DTS) DTS 算法, 解决了指尖实时准确跟踪的问题。该方法首先利用深度信息准确的检测到指尖, 并根据检测结果与卡尔曼滤波结果的相似度匹配方法克服了跟踪算法的目标区域标定不精确的问题。并通过增加监督环节, 实现了跟踪结果发生偏离时的矫正, 提高了跟踪算法的鲁棒性。

1 DTS 算法流程

本文提出的基于深度信息的指尖 DTS 的流程如图 1 所示。

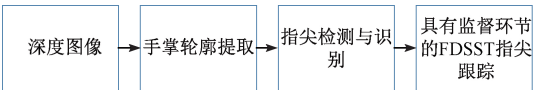


图 1 指尖 DTS 流程
Fig.1 Flow chart of the fingertip DTS

在 DTS 算法中, 首先需要从 Kinect 直接获取的深度图像进行滤波处理, 然后对得到的滤波后的图像进行阈值分割和特征提取来获得手掌的轮廓, 再对提取到的手掌轮廓图进行指尖的检测与识别, 并获得指尖的稳定初始状态, 最后将指尖的稳定初始状态作为具有监督环节的 FDSST 跟踪算法的初始条件来启动跟踪, 从而实现指尖的实时稳定跟踪。

2 深度图像滤波

2.1 原始深度信息图

Kinect 直接提供的深度图如图 2 所示。
受制于 Kinect 设备本身的硬件限制, Kinect 直接提供的深度图像质量不高, 不符合进一步应用的需求。



图 2 未经滤波的深度信息图
Fig.2 Unfiltered depth image

因此对 Kinect 直接提供的深度图像进行滤波处理^[16-18]。

2.2 深度图像滤波

对深度信息进行众数滤波, 具体步骤为:

- 1) 遍历深度图像, 将深度值为 0 的点做为目标点, 以目标点为中心点, 找到 3×3 的正方形邻域;
- 2) 统计正方形邻域中各深度值出现的次数;
- 3) 如果某个深度值出现的次数超过了设置的阈值 (这里阈值设置为 3), 则将中心点设置为该深度值, 否则, 中心点的深度值仍为 0。

如图 3 所示, 红色框里有 5 个点的深度值为 2 400, 因此红色框里的中心点的深度值可以设置为 2 400。

1 200	1 200	1 200	1 200	2 400	1 200	1 200	1 200	1 200	2 400
1 200	1 200	1 200	2 400	2 400	1 200	1 200	1 200	2 400	2 400
1 200	1 200	0	2 400	2 400	1 200	1 200	2 400	2 400	2 400
1 200	2 400	2 400	2 400	2 400	1 200	2 400	2 400	2 400	2 400
2 400	2 400	2 400	2 400	2 400	2 400	2 400	2 400	2 400	2 400

图 3 众数滤波示意
Fig.3 Diagram of majority filtering

如图 4 所示, 可以看出图 2 中图像内部的“空穴点”得到了明显改善, 而且图像边缘的锯齿感明显减小, 同时图像的边缘等重要特征仍然保留着。



图 4 经滤波后的深度信息
Fig.4 Filtered depth image

3 手掌轮廓提取

手掌轮廓提取的前提为手掌距离 Kinect 最近。

3.1 手掌区域分割

Kinect 提供的深度图中,每个像素都是 16 位,其中使用 13 位记录深度数据,用 3 位记录用户 ID。首先把用户 ID 为 0 的点(即背景)去除。然后遍历滤波后的深度图像,查找到深度值最小的点,即该点为距离 Kinect 最近的点,考虑到手掌是一个区域,因此还应该在最小的深度值的基础上加上 50(实验经验值),将其作为手掌分割的阈值^[19-20]。具体的阈值分割如下所示,其中 I 为深度图像中像素的 RGB 值。

$$I = \begin{cases} (0,0,0), & \min - 50 \leq \text{depth} \leq \min + 50 \\ (255,255,255), & \text{其他} \end{cases} \quad (1)$$

手掌区域分割图如图 5 所示。



图 5 手掌区域分割

Fig.5 Segmentation image of palm area

3.2 手掌轮廓提取

在本文中,提出了基于手掌轮廓边缘点的特征进行二值图像的快速边缘检测方法。定义 Q 为手掌轮廓边缘点的集合,即 $Q = \{T(n) \mid T(n) > \text{Threshold}\}$ 。其中 $T(n)$ 为定义的边缘轮廓点的特征,表示的是第 n 个待检测的点上下左右 4 个邻域里白点的个数, Threshold 为定义的阈值。检测的过程为:

- 1) 遍历手掌区域二值图像,选取黑色的点为候选点,将它的上下左右 4 个邻域作为检测区域;
- 2) 统计检测区域里白点的个数 $T(n)$;
- 3) 若 $T(n) \geq \text{Threshold}$ (这里取为 1),则该点为边缘上的点,否则该点为手掌内部的点。

如图 6 所示,可以得到 $T(1) = 2, T(2) = 0$, 因此编号为 1 的点就是手掌轮廓上的点,编号为 2 的点为手掌内部的点。同理找到所有位于手掌轮廓上的点。

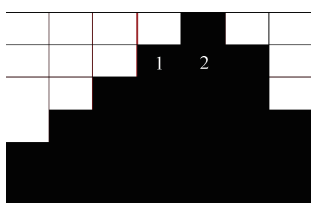


图 6 轮廓提取示意

Fig.6 Diagram of contour extraction

如图 7 所示,红色的曲线表示的是手掌的边缘轮廓,其余部分均为白色。



图 7 手掌轮廓

Fig.7 Palm contour

4 指尖检测与识别

4.1 凸包检测

可以通过检测手掌轮廓的凸包^[21-23],来获得包含指尖的集合。本文在 Graham 扫描法的基础上融入了深度图像信息进行凸包的检测。

对于得到的深度图像和手掌轮廓 RGB 图像,这两个图像上的点的对应关系如图 8 所示。

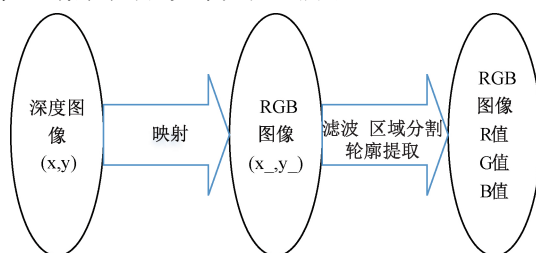


图 8 图像关系示意

Fig.8 Diagram of image relationship

对手掌轮廓上的二维点集 P 进行凸包检测:

- 1) 遍历点集 P , 找到纵坐标最小的点,如果有多个这样的点则选取最左边的点作为基点 M ;
- 2) 计算点集 P 中其余点与基点所构成的向量 $\langle M, H \rangle$ ($H \in P$ 且 $H \neq M$) 与 X 轴的夹角,对夹角按从大到小(从小到大)排序,得到排序后的点集容器 data ;
- 3) 用容器 vector 来存储符合在凸包上的边界点,遍历 data , 计算 3 个相邻的点所构成的三角形的有向面积,通过有向面积的正负来确定凸包上的点。

对于给定的三点 $A(x_a, y_a), B(x_b, y_b), C(x_c, y_c)$, 这三点构成的三角形 ABC 的有向面积 ΔS_{ABC} 如下:

$$\Delta S_{ABC} = \frac{1}{2} \begin{vmatrix} 1 & x_a & x_a \\ 1 & x_b & y_b \\ 1 & x_c & y_c \end{vmatrix} \quad (2)$$

如果 $\Delta S_{ABC} > 0$,则可以得出 A、B、C 三点的分布顺序 $A \rightarrow B \rightarrow C$ 为逆时针,反之则为顺时针分布。

设 2 个向量分别为 $V_1(x_1, y_1), V_2(x_2, y_2)$, 定义 2 个

向量的模分别为 m_1, m_2 , 向量 V_1, V_2 与向量 $(1, 0)$ 的内积分别为 n_1, n_2 , 则有:

$$m_1 = \sqrt{x_1^2 + y_1^2} \quad m_2 = \sqrt{x_2^2 + y_2^2} \quad (3)$$

$$n_1 = x_1/m_1 \quad n_2 = x_2/m_2 \quad (4)$$

若 $n_1 > n_2$, 则 V_1 与 X 轴的夹角大于 V_2 与 X 轴的夹角。凸包检测结果如图 9 所示。



图 9 手掌凸包

Fig.9 Palm convex hull image

需要对初步得到的凸包进一步筛选, 解决凸包距离过近的问题。设凸包 $A(x_1, y_1), B(x_2, y_2)$, 点 A 与点 B 之间的欧式距离为 dis , 阈值为 T (实验取为 20), 则有:

$$dis = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (5)$$

遍历初步得到的凸包集合 $vector$, 计算两两相邻凸包之间的距离, 并与 T 进行比较, 若两凸包之间距离小于阈值, 则只保留两凸包中横坐标大的凸包。



图 10 筛选后的手掌凸包

Fig.10 Diagram of palm-shaped convex hull

4.2 指尖检测

可以通过 Kinect 传回的骨骼数据流来获得手掌掌心位置信息, 如图 11 所示。



图 11 掌心示意

Fig.11 Palm diagram

获取到手掌的掌心后, 找到手掌的最大内切圆。如图 12 所示。

$$\sigma\% = \frac{5}{num} \times 100\% \quad (6)$$

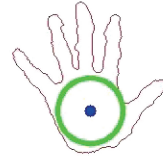


图 12 手掌最大内切圆

Fig.12 Maximum inscribed circle of palm

设阈值 $T = N \times R$, 通过比较凸包集 $vector$ 中各凸包到掌心的欧式距离与阈值 T 的大小去除手掌上的凸包。设凸包集 $vector$ 中凸包的总个数为 num , 指尖个数在 $vector$ 中所占的百分比为 $\sigma\%$ 。当 $N = 1.5 \sim 2.5$ 时, $\sigma\%$ 为 100%。指尖示意图如图 13 所示。



图 13 指尖示意

Fig.13 Fingertip image

在检测到 5 个指尖后, 根据指尖的相对位置与相对距离进一步分别识别出 5 个指尖。

4.3 指尖稳定状态的确定

在初步识别检测出指尖后, 得到的识别结果可能并不稳定, 需要进一步确定出一个稳定的状态。本文首先使用基于深度信息的卡尔曼滤波, 再将滤波的结果 $P_k(x_k, y_k)$ 与检测的结果 $P_c(x_c, y_c)$ 进行相似度 α 比较, 如式 (7) 所示, 如果相似度超过了阈值 (本文中设置为 1), 则认为该帧的检测结果不稳定, 否则认为该帧的检测结果稳定, 可以进行跟踪部分。

$$\alpha = \sqrt{(x_k - x_c)^2 + (y_k - y_c)^2} \quad (7)$$

对于每一帧的深度数据, 我们要测量指尖在每一帧时的坐标位置 $(x(t), y(t))$ 和速度 $(v_x(t), v_y(t))$ 。因此定义卡尔曼滤波器里的状态向量为 x_t , 且

$$x_t = (x(t), y(t), v_x(t), v_y(t)) \quad (8)$$

定义卡尔曼滤波器里的测量向量为 z_t , 且

$$z_t = (x(t), y(t)) \quad (9)$$

由上面的定义, 可得到关于指尖的卡尔曼方程为:

$$x_t = Ax_{t-1} + w_{t-1} \quad (10)$$

$$z_t = Hx_t + v_t \quad (11)$$

式中: A 为状态转移矩阵; H 为测量矩阵; w_t 为系统噪声; v_t 为测量噪声。且

$$p(w) \sim N(0, Q_t) \quad p(v) \sim N(0, R_t) \quad (12)$$

式中: Q_t 为 w_t 的协方差矩阵, 这里 Q_t 为对角线元素为 10^{-5} 的 4×4 对角矩阵; R_t 为 v_t 的协方差矩阵, 这里 R_t 为

对角线元素为 10^{-1} 的 2×2 对角矩阵。

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13)$$

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (14)$$

考虑到 Kinect 传回深度图像的帧数为 30 FPS, 即指尖前后位置的时间差 Δt 为 $1/30$ s, 可认为指尖 Δt 内的运动为匀速运动。可以得到 $\mathbf{A}$ 矩阵和 $\mathbf{H}$ 矩阵。

$\mathbf{P}_t$ 为误差协方差矩阵, 表示对当前预测状态的信任度, 它越小说明越相信当前预测状态。因为卡尔曼滤波是一种迭代优化滤波器, $\mathbf{P}_t$ 的值会不断的改变, 当系统进入稳态之后 $\mathbf{P}_t$ 值会收敛成一个最小的估计方差矩阵。在实际应用中, 一般把 $\mathbf{P}_0$ ($\mathbf{P}_t$ 矩阵的初始化) 设置为:

$$\mathbf{P}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (15)$$

在前面的设置条件下, 利用指尖的卡尔曼滤波器, 通过 t 帧之前的位置预测它在 t 帧时的位置。

卡尔曼滤波图如图 14 所示, 黑色的圆圈显示的是经 Kinect 提供的深度信息而检测到的中指指尖位置。蓝色的圆圈表示的是卡尔曼滤波器预测的中指指尖位置, 红色的圆圈表示的是经卡尔曼滤波器更新阶段之后矫正后的中指指尖位置。可以看到, 在这一帧中, Kinect 检测到的点与卡尔曼滤波结果的相似度为 1, 因此可以认为该帧指尖的状态为稳定状态。

Current Point:(267,222)
Predict Point:(267,221)
Current Point:(267,221)



图 14 卡尔曼滤波示意

Fig.14 Diagram of Kalman filter

5 具有监督环节的 FDSST 跟踪算法

在指尖的跟踪过程中, 考虑到指尖可能会在运动的过程中发生旋转或者倾斜, 这将会导致指尖的追踪目标的尺寸发生改变, 为了能够在追踪目标尺寸发生改变的情况下保持对目标的精准跟踪, 本文在 FDSST 算法的基

础上提出了具有监督环节的跟踪算法。

5.1 FDSST 跟踪算法

FDSST 算法主要思路为: 通过一系列大小为 $M \times N$ 的训练样本 $\mathbf{f}_i$ 来得到相关滤波器 $\mathbf{h}$, 并在下一帧中根据互相关来定位目标所在的位置。 $\mathbf{g}_i$ 为 $\mathbf{f}_i$ 所对应的响应输出, $\mathbf{g}_i$ 由高斯函数产生, 且其峰值位置是在 $\mathbf{f}_i$ 的中心位置。

只考虑一个训练样本的情况下, 最优的相关滤波器 $\mathbf{h}$ 由相关响应与期望相关输出的最小损失函数得:

$$\varepsilon = \left\| \sum_{l=1}^d \mathbf{h}^l * \mathbf{f}^l - \mathbf{g} \right\|^2 + \lambda \sum_{l=1}^d \left\| \mathbf{h}^l \right\|^2 \quad (16)$$

式中: λ 为正则项系数, $l(l=1, 2, \dots, d)$ 为样本 $\mathbf{f}$ 的特征维数。求解出的最优相关滤波器 $\mathbf{H}^l$ 为(在本节中大写字母均为对应小写字母的傅里叶变换, $*$ 表示复共轭)

$$\mathbf{H}^l = \frac{\mathbf{G}^* \mathbf{F}^l}{\sum_{k=1}^d (\mathbf{F}^k)^* \mathbf{F}^k + \lambda} = \frac{\mathbf{A}^l}{\mathbf{B}_l} \quad (17)$$

在实际应用中, 通常将 $\mathbf{H}^l$ 拆分为分子 $\mathbf{A}^l$ 和分母 $\mathbf{B}^l$ 进行更新, 更新公式为:

$$\mathbf{A}_t^l = \eta \mathbf{G}_t^* \mathbf{F}_t^l + (1 - \eta) \mathbf{A}_{t-1}^l \quad (18)$$

$$\mathbf{B}_t = \eta \sum_{k=1}^d (\mathbf{F}_t^k)^* \mathbf{F}_t^k + (1 - \eta) \mathbf{B}_{t-1} \quad (19)$$

式中: 下标 t 表示第 t 帧图像; η 表示学习率。

FDSST 为了提高运算速度, 利用 PCA 进行了特征降维。目标模板 $\mathbf{u}_t$ 更新为:

$$\mathbf{u}_t = (1 - \eta) \mathbf{u}_{t-1} + \eta \mathbf{f}_t \quad (20)$$

构造投影矩阵 $\mathbf{P}_t$, 该矩阵定义了特征投影到的低维子空间, 其中 $\mathbf{P}_t$ 的维度为 $\tilde{d} \times d$, $\tilde{d}$ 为压缩后的特征维度, $\mathbf{P}_t$ 通过最小化重构目标模板 $\mathbf{u}_t$ 误差获得:

$$\varepsilon = \sum_n \left\| \mathbf{u}_t(n) - \mathbf{P}_t^T \mathbf{P}_t \mathbf{u}_t(n) \right\|^2 \quad (21)$$

滤波器使用压缩后的训练样本 $\tilde{\mathbf{F}}_t$ 与压缩后的目标模板 $\tilde{\mathbf{U}}_t$ 进行更新(其中 $\mathbf{F}$ 表示傅里叶变换):

$$\tilde{\mathbf{F}}_t = \mathbf{F} \{ \mathbf{P}_t \mathbf{f}_t \} \quad \tilde{\mathbf{U}}_t = \mathbf{F} \{ \mathbf{P}_t \mathbf{u}_t \} \quad (22)$$

$$\tilde{\mathbf{A}}_t^l = \mathbf{G}^* \tilde{\mathbf{U}}_t^l, l = 1, \dots, \tilde{d} \quad (23)$$

$$\tilde{\mathbf{B}}_t = (1 - \eta) \tilde{\mathbf{B}}_{t-1} + \eta \sum_{k=1}^{\tilde{d}} (\tilde{\mathbf{F}}_t^k)^* \tilde{\mathbf{F}}_t^k \quad (24)$$

对于下一帧中的待测样本 $\mathbf{z}_t$, 通过 PCA 降维得到基于压缩后的样本 $\tilde{\mathbf{z}}_t = \mathbf{F} \{ \mathbf{P}_{t-1} \mathbf{z}_t \}$, 由式(25)得到样本 $\mathbf{z}_t$ 的相关分数 y_t 。

$$y_t = \mathbf{F}^{-1} \left\{ \frac{\sum_{l=1}^{\tilde{d}} (\tilde{\mathbf{A}}_{t-1}^l)^* \tilde{\mathbf{z}}_t^l}{\tilde{\mathbf{B}}_{t-1} + \lambda} \right\} \quad (25)$$

待测样本 $\mathbf{z}_t$ 中, y 的最大值所对应的点即为这一帧目标区域的中心位置。因为样本的维数 d 为任意值, 所以 FDSST 中相互独立的位置滤波器与尺度滤波器都使用上述方法, 来分别获得新的一帧中目标区域的位置与目标区域的尺寸。

5.2 具有监督环节的 FDSST 跟踪算法

1) 目标区域标定

在 FDSST 中,输入图像 f 通常为使用者手动标定的一个目标区域,并将这个手动标定的目标区域作为初始的目标区域。但是手动标定的追踪目标区域的误差较大,不适用于追踪指尖等细小物体。本文提出了自动准确标定目标区域的方法。

以得到的指尖稳定状态 $P_0(x_0, y_0)$ 为中心点构建一个 20×24 的矩形区域,用这个矩形区域来代替 FDSST 中需要手动标定的目标区域 f 。这样,在之后的指尖跟踪过程中,尽管指尖的位置一直在发生变化,矩形区域的位置也一直在发生变化,但是指尖能一直位于矩形区域的中心点,这为后面从每一帧的跟踪结果中提取出指尖的位置提供了方便。

2) 监督环节

根据 1) 中提出的目标区域标定的改进方法,每一帧中跟踪到的矩形区域的中心位置就是每一帧中指尖的位置。但是,在实际的指尖运动过程中,由于指尖的运动相对自由,可能在运动中由于指尖的弯曲或者倾斜等导致指尖所在的矩形区域发生较大的形变,或者是指尖在某一帧时间内运动速度过快,这些都会导致仅仅根据追踪的结果而得到的指尖的位置发生偏差,如图 15 所示(矩形框表示的是跟踪到的目标区域,圆点表示的是跟踪到的指尖位置)。



图 15 目标区域发生较大形变时指尖跟踪

Fig.15 Fingertip tracking diagram in the deformed target area

在图 15 中,可以看到,在第 10 帧中,还能准确的跟踪到指尖的位置,但是在第 13 帧中,由于指尖在极短时间内发生了大幅度的弯曲,导致目标区域发生了大的形变,因此此时跟踪到的指尖位置与实际指尖位置存在偏差。为了提高追踪过程中的鲁棒性,本文在 FDSST 的基础上提出了具有监督环节的追踪算法。

考虑到在指尖检测部分,我们是通过指尖的特征在深度图像中检测到了指尖,而由于背景部分或者手掌等部分并不具有指尖的特征,因此可以将指尖的特征作为监督环节的判断器,通过将追踪到的指尖输入到判断器中,在判断器中对追踪到的指尖进行指尖特征匹配,从而判断出追踪到的指尖是否正确。如果不正确,则需要对

这一帧的追踪结果进行矫正。

设追踪到的指尖的位置为 P_i , 指尖的特征集合为 R , 且 $\{P_i \in R \mid P_i \in Q, P_i \in vector, dis > T\}$, 其中 Q 为深度图像中得到手掌轮廓集合, $vector$ 为深度图像中得到的去重后的凸包集合, dis 为 P_i 与掌心的欧式距离, T 为掌心距离阈值。监督环节的具体算法步骤如下:

1) 启动指尖检测部分,从 Kinect 提供的深度图中获得稳定的指尖状态,以稳定状态为中心位置,在图像中标定出大小为 20×24 的图像 patch,将这个 patch 作为具有监督环节的 FDSST 跟踪算法的初始目标区域,并根据式(17)、式(23)~(24)得到目标的位置模型和尺度模型;

2) 在后续帧的图像中,由式(25)得到跟踪到的目标位置 P_i ,通过深度图像—彩色图像的坐标转换,得到 P_i 在深度图像中的对应位置 P_i ;

3) 将 P_i 输入到判断器中,将 R 中的指尖特征与 P_i 进行特征匹配;

4) 如果 P_i 与 R 中的全部特征完全匹配,则说明这一帧的指尖追踪 P_i 准确,继续步骤 2) 进行下一帧的指尖追踪;

5) 如果 P_i 不能与 R 中的全部特征完全匹配,则说明这一帧的指尖追踪 P_i 存在偏差,需要返回到步骤 1) 中,重新得到指尖的稳定状态并重新初始化具有监督环节的 FDSST 跟踪算法。

如图 16 所示,在 13 帧中,指尖的跟踪发生了偏差,被监督器检测到了,通过重启基于深度图像的指尖检测部分,在 17 帧中,跟踪算法又准确的跟踪上了指尖。



图 16 带有监督环节的跟踪算法的指尖跟踪

Fig.16 Fingertip tracking algorithm with supervision

6 实验结果

本次研究的实验是在 VS2017, OpenCV3. 3. 4, Kinect SDK V1. 8 的环境下进行的。

6.1 指尖检测实验

记准确性为 δ , 表示检测到的指尖个数占总的凸包个数的比例。准确性实验结果如表 1 所示。

表 1 关于指尖检测准确性的实验结果

Table 1 Experimental results of fingertip detection accuracy

指尖 组数	拇指	食指	中指	无名指	小指	总凸包 个数	准确性
第一组	1	1	1	1	1	5	1.00
第二组	1	1	1	1	1	6	0.83
第三组	1	1	1	1	0	5	0.80
第四组	1	1	1	1	1	5	1.00
第五组	1	1	1	1	1	5	1.00
第六组	1	1	1	1	0	5	0.80
第七组	1	1	1	1	1	5	1.00

从表 1 中可以看出,5 个指尖检测的准确性可靠,满足需求。

之后进行 5 个指尖的识别准确性实验,该实验旨在检验算法能够识别出待检测指尖具体为哪一个指尖,以及指尖识别的准确性。实验结果如表 2 所示。

表 2 关于指尖识别准确性实验结果

Table 2 Experimental results of fingertip recognition

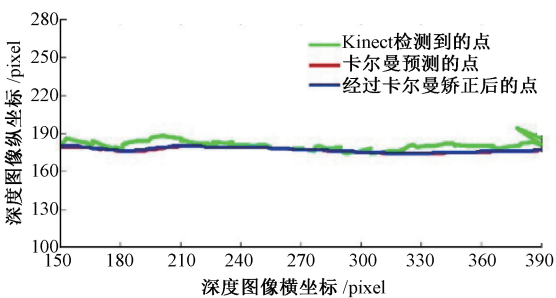
检测 实际	拇指	食指	中指	无名指	小指	准确性
拇指	√					1.00
食指		√				0.90
中指			√			0.90
无名指				√		1.00
小指					√	0.80

从表 2 中可以得知,指尖识别准确性符合需求。

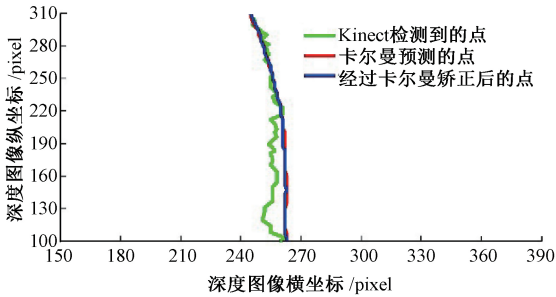
6.2 基于深度信息的卡尔曼滤波实验

进行了 3 组基于深度信息的卡尔曼滤波实验,该实验旨在证明基于深度信息的卡尔曼滤波能够在指尖抖动时仍然能够得到指尖的平稳运动状态。

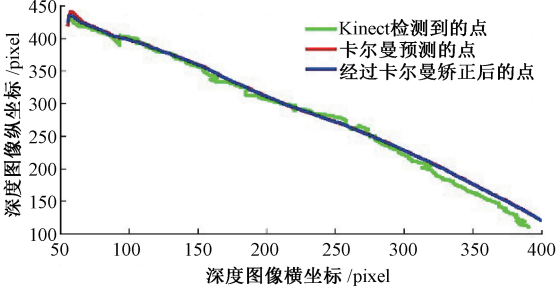
图 17 中,横轴表示深度图像的横坐标,纵轴表示深度图像的纵坐标,绿线表示指尖的实时轨迹,即直接通过 Kinect 提供的深度数据而检测到的指尖实时坐标。红线表示的是基于深度数据的卡尔曼滤波预测的指尖的轨迹。蓝线表示的是基于深度数据的卡尔曼滤波矫正后的指尖的轨迹。在图 17(a)、(b)、(c)中可以看到绿色所表示指尖的实际的运动轨迹不稳定,一直处于波动的状态,这是因为在实时采集指尖的深度信息时伴随着较大的噪声干扰。而对于蓝色所表示的经过卡尔曼滤波矫正后的指尖的轨迹,可以发现轨迹的波动很小,而且更接近于事先设计的每种运动轨迹,表明在复杂的环境干扰下,基于深度数据的卡尔曼滤波仍然具有较强的鲁棒性。因此在指尖发生细微抖动时,基于卡尔曼滤波,仍然能得到指尖的稳定状态。



(a) 指尖横向直线运动效果
(a) Lateral linear motion tracking of the fingertips



(b) 指尖纵向直线运动效果
(b) Longitudinal linear motion tracking of the fingertips



(c) 指尖斜线运动效果
(c) Oblique movement of the fingertips

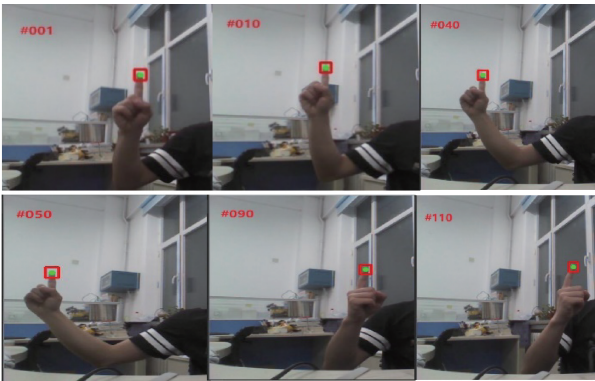
图 17 卡尔曼滤波器在指尖不同运动下的滤波效果
Fig.17 Kalman filter in different movements of fingertips

6.3 指尖实时追踪实验

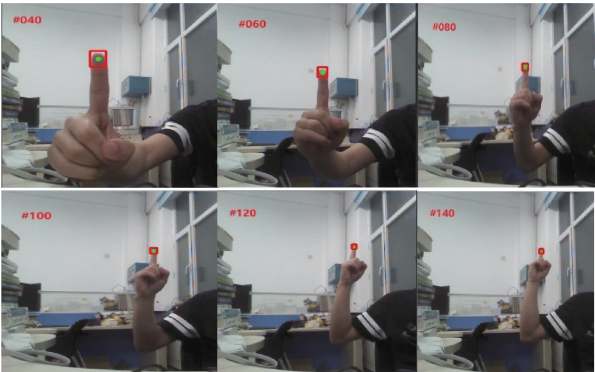
在确定了指尖的稳定状态后,进行了多组基于 DTS 算法的指尖跟踪实验。

图 18 中,红色矩形方框表示跟踪到的目标区域,绿色圆圈表示跟踪到的指尖位置。在 Kinect 彩色摄像头传回图像的速率为 30 帧/秒的前提下,指尖的跟踪算法的速度为 23~25 帧/秒。通过图 18 所示的 4 种运动看到,基于 DTS 算法的指尖运动跟踪稳定,不论是平常的直线运动(图 18(a)),还是目标尺度不断变化的运动(图 18(b)、(c)、(d)),都能够准确稳定的追踪指尖。尽管在运动期间,在有些帧中,追踪到的指尖的位置与实际位置发生了偏离,但是由于 DTS 算法中的监督环节的存在,能够及时的检测到错误的跟踪结果,并给与及时的修正,因此本系统具有较强的鲁棒性。

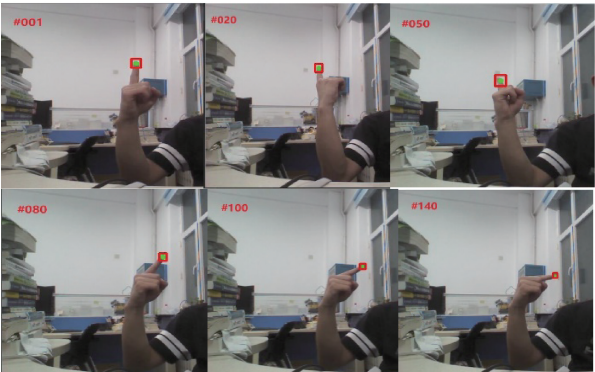
本文提出的基于深度信息的指尖 DTS 能够应用在空中汉字书写领域,如图 19 所示,利用 DTS 算法实现了汉字“尹”的空中书写。



(a) 指尖直线运动跟踪图
(a) Fingertip tracking in linear motion



(b) 指尖由近到远运动跟踪图
(b) Fingertip tracking in near-far motion



(c) 指尖倾斜运动跟踪图
(c) Fingertip tracking in tilt motion



(d) 指尖弯曲运动跟踪图
(d) Fingertip tracking in bending motion

图 18 DTS 算法在指尖不同运动下跟踪效果

Fig.18 DTS algorithm in different movements of fingertips



图 19 汉字“尹”的空中书写

Fig.19 The air writing of the Chinese character “Yin”

7 结 论

本文提出了一种基于深度信息的指尖追踪算法 (DTS 算法),算法融合了检测,跟踪与监督 3 大部分,通过将指尖检测结果与卡尔曼滤波结果相结合从而能得到指尖稳定状态,解决了跟踪算法初始状态不稳定问题。通过将监督环节融入到跟踪算法中,解决了当跟踪目标发生偏移时的矫正问题,提高了跟踪算法的鲁棒性。并通过汉字“尹”的空中书写,证明了本算法在人机交互领域具有应用性。

本文提出的基于深度信息的 DTS 算法虽然实现了指尖的实时准确跟踪,但是在跟踪速度等方面仍然存在改进空间,这是下一步研究的方向和重点。

参考文献

- [1] WU ZH H, PAN G. Smart: models and methods for pervasive computing[M]. Springer,2013:83-100.
- [2] KEATON T, DOMINGUEZ S M, SAYED A H. SNAP&TELL: A multi-modal wearable computer interface for browsing the environment [C]. Sixth International Symposium on Wearable Computers, 2002: 75-82.
- [3] MICHAEL J J, JAMES M R. Statistical color models with application to skin detection[J]. International Journal of Computer Vision, 2002,46(1):81-96.
- [4] SHAHZAD M. Real-time hand tracking and finger tracking for interaction[R]. CSC2503F Project Report, 2003:1-21.
- [5] SHI J H, ZHANG M M, PAN ZH G. A real-time

- bimanual 3D interaction method based on bare-hand tracking [C]. 19th ACM International Conference on Multimedia, 2011:1073-1076.
- [6] PAN ZH G, LI Y, ZAHNG M M, et al. A real-time multi-cue hand tracking algorithm based on computer vision[C]. IEEE Virtual Reality, 2010:219-222.
- [7] SATO Y, KOBAYASHI Y, KOIKE H. Fast tracking of hands and fingertips in infrared images for augmented desk interface[C]. Fourth IEEE International Conference on Automatic Face and Gesture Recognition, 2000: 462-467.
- [8] OKA K, SATO Y, KOIKE H. Real-time fingertip tracking and gesture recognition [J]. IEEE Computer Graphics and Applications, 2002,22(6):64-71.
- [9] SONG P, WINKLER S, GILANI S O, et al. Vision-based projected tabletop interface for finger interactions[C]. IEEE international conference on Human-computer interaction, 2007:49-58.
- [10] 范文婕,王命延,杨文姬.基于深度图像的指尖和掌心特征提取方法[J]. 计算机应用, 2015, 35(6): 1791-1794,1804.
- FAN W J, WANG M Y, YANG W J. Feature detection method of fingertip and palm based on depth image[J]. Journal of Computer Applications, 2015, 35(6): 1791-1794,1804.
- [11] 叶植超. 基于 Kinect 的空中手写系统[D]. 广州:华南理工大学,2014: 5-38.
- YE ZH CH. Writing in the air system base on Kinect sensor [D]. Guangzhou: South China University of Technology,2014:5-38.
- [12] BOLME D S, BEVERIDGE J R, DRAPER B A, et al. Visual object tracking using adaptive correlation filters[C]. IEEE Conference on Computer Vision and Pattern Recognition,2010:2544-2550.
- [13] HENRIQUES J F, CASEIRO R, MARTINS P, et al. High-speed tracking with kernelized correlation filters[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015,37(3):583-596.
- [14] DANELLJAN M, HAGER G, KHAN F S, et al. Accurate scale estimation for robust visual tracking[C]. British Machine Vision Conference, 2014, 65:1-11.
- [15] DANELLJAN M, HAGER G, KHAN F S, et al. Discriminative scale space tracking [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017,39(8):1561-1575.
- [16] SMARI K, BOUHLEL M S. Gesture recognition system and finger tracking with Kinect: Steps [C]. 7th International Conference on Science of Electronics, Technologies of Information and Telecommunications., 2016:544-548.
- [17] BAKAR M Z A, SAMAD R, PEBRIANTI D, et al. Finger application using K-Curvature method and Kinect sensor in real-time[C]. IEEE International Symposium on Technology Management and Emerging Technologies, 2015:218-222.
- [18] TAO H Y, YU Y L. Finger tracking and gesture recognition with Kinect [C]. IEEE 12th International Conference on Computer and Information Technology, 2012:214-218.
- [19] 孟明,杨方波,余青山,等.基于 Kinect 深度图像信息的人体运动检测[J]. 仪器仪表学报, 2015, 36(2): 386-393.
- MENG M, YANG F B, SHE Q SH, et al. Human motion detection based on the depth image of Kinect [J]. Chinese Journal of Scientific Instrument, 2015,36(2): 386-393.
- [20] SHI ZH, LI Y H, YU K, et al. Research of hand gesture using Kinect based on finger recognition[J]. Computer Modelling & New Technologies, 2014,18(10):158-162.
- [21] 秦超龙,宋爱国,吴常铖,等.基于 Unity3D 与 Kinect 的康复训练机器人情景交互系统[J]. 仪器仪表学报, 2017,38(3): 530-536.
- QIN CH L, SONG A G, WU CH CH, et al. Scenario interaction system of rehabilitation training robot based on Unity3D and Kinect [J]. Chinese Journal of Scientific Instrument, 2017,38(3): 530-536.
- [22] VINH T Q, TRI N T. Hand gesture recognition based on depth image using Kinect sensor [C]. 2nd National Foundation for Science and Technology Development Conference on Information and Computer Science, 2015:

34-39.

[23] WANG CH, LAI ZH Y, WANG H Y. Hand gesture recognition based on perceptual shape decomposition with a Kinect camera[C]. IEICE Transactions on Information and Systems,2013:2147-2151.

[24] 余铎,王耀南,毛建旭,等.基于视觉的移动机器人目标跟踪方法[J].仪器仪表学报,2019,40(1):227-235.

YU D, WANG Y N, MAO J X, et al. Vision-based object tracking method of mobile robot [J]. Chinese Journal of Scientific Instrument,2019,40(1):227-235.

作者简介



孟浩,1989 年于哈尔滨船舶工程学院获得学士学位,1992 年于哈尔滨船舶工程学院获得硕士学位,2003 年于哈尔滨工程大学获得博士学位,现为哈尔滨工程大学教授、博士生导师,主要研究方向为机器人与智能控制、机器视觉技术。

E-mail: menghao@hrbeu.edu.cn

Meng Hao received his B. Sc. degree and M. Sc. degree both from Harbin Shipbuilding Engineering Institute in 1989 and 1992 respectively, and received his Ph. D. degree from Harbin Engineering University in 2003. He is currently a professor and doctoral supervisor at Harbin Engineering University. His research interests include robot and intelligent control, machine vision detection technology.



尹维考 (通信作者),2017 年于长春大学获得学士学位,现为哈尔滨工程大学硕士研究生,主要研究方向为图像处理与目标跟踪。

E-mail: 598531842@qq.com

Yin Weikao (Corresponding author) received his B. Sc. degree from Changchun University in 2017. He is currently a M. Sc. student at Harbin Engineering University. His main research interests include image processing and object tracking.