

一种快速高效的手势跟踪识别方法

全冬兵^{1,2} 程如中² 赵勇^{2,†} 魏江月² 梁浩² 魏益群³

1. 中国工程物理研究院计算机应用研究所, 绵阳 621900; 2. 北京大学信息科学技术学院, 北京 100084;
3. 深港产学研基地, 深圳 518057; † 通信作者, E-mail: yongzhao@pkusz.edu.cn

摘要 为了降低识别复杂度, 提高识别效率, 实现手势的快速高效跟踪, 提出一种分情况检测思想和搜索框概念。首先对图像进行细检测, 得到目标的准确位置, 然后通过粗检测与跟踪相结合的方式对目标跟踪, 并对跟踪结果进行修正和可信度判断。实验结果显示: 算法对图像手势的平均检测跟踪正确率可以达到97.36%, 且保证平均漏检率在5%以下, 对各种外界因素具有较好的鲁棒性; 算法对视频图像的处理速度达19.42 帧/秒, 满足人机交互系统中的实时性需求; 与 TLD 算法相比, 本算法在处理速度上有数量级的改善, 算法结果的正确率也有明显优势。

关键词 目标跟踪; 人机交互; 分情况检测; 搜索框

中图分类号 TP3

A Real Time and Effective Method for Hand-Gesture Detection and Tracking

QUAN Dongbing^{1,2}, CHENG Ruzhong², ZHAO Yong^{2,†}, WEI Jiangyue², LIANG Hao², WEI Yiqun³

1. Institute of Computer Application, China Academy of Engineering Physics, Mianyang 621900; 2. School of Electronics Engineering and Computer Science, Peking University, Beijing 100084; 3. PKU-HKUST ShenZhen-HongKong Institution, Shenzhen 518057;
† Corresponding author, E-mail: yongzhao@pkusz.edu.cn

Abstract An algorithm for gesture detection and tracking in HCI (human-computer interaction) is designed to meet real-time and accuracy requirements. An innovative conception, which includes using distinguishing detection methods to detect hand-gesture for different conditions and using searching-box to decrease the searching zone, is proposed. The result shows that the detection rate can reach 97.36% while the missing rate lowers than 5%. It is robust to various external factors. However, it also meets the real-time, as the frame rate can reach 19.42. Compared with TLD, this algorithm has not only magnitude improvement in processing speed but also obvious advantages in accuracy.

Key words object tracking; human-machine interaction; distinguishing detection; searching box

随着计算机视觉的发展, 图像手势识别在人机交互中的应用成为一大热点, 出现了各种各样的识别方法^[1-2]。Zhu 等^[3]运用肤色、手势轮廓及运动信息实现对 12 种定义手势的识别, 识别率达 90%。Fang 等^[4]使用肤色及运动信息实现手势的分割, 利用尺度空间信息及手势区域宽高比来实现手势的识别。Rautaray 等^[5]采用 LK 光流法来实现手

势的检测, 再通过 K 最近邻算法来实现手势识别。Brethes 等^[6]采用分水岭算法, 利用肤色信息来实现手势的分割, 将颜色块信息结合形状信息来实现手势的跟踪。Li 等^[7]设计的 WEGM 算法对手势的识别率达 97.08%。Hsieh^[8]等、Song 等^[9]和 Chen 等^[10]使用不同特征进行训练, 用得到的分类器来识别手势, 正确率分别达 94%, 75% 和 98%。除此之外, 还

有研究采用深度图信息^[11]、GMM 建模^[12]等方法来实现手势的分割,采用熵信息^[11]、骨架信息^[12]、3D 建模^[13]、全局与局部特征相结合^[14]等其他方法来实现手势的识别。

这些方法都存在各自的优势,比如识别正确率高、识别手势种类多等。但是还无法满足应用于人机交互的高要求。因为人机交互中的手势识别与跟踪,不仅要求较高的手势识别准确度,更要求识别率与实时性的平衡。

针对上述问题,本文提出一种解决方案:一方面,在现有手势检测方法的基础上,提出分情况检测思想与搜索框的概念来降低时耗,提高算法的实时性;另一方面,仅采用最易于识别的拳头和手掌两种手势,不仅可以保证识别的高准确度,还可以减少识别多种手势带来的算法高复杂度,从而进一步降低时耗。在实际操控时,操控命令由拳头运动轨迹来定义,而非传统的手势直接定义,操控的有效性由手掌来进行释放,从而实现整个人机交互操作过程。本文算法流程如图 1 所示。

1 分情况检测

基于常规检测与跟踪直接结合实现目标检测和跟踪的算法,本文提出分情况检测,即把检测模块分为粗、细检测。当系统初始化或粗检测结果不可信时,调用细检测模块实现高准确度的目标检测;在其他情况下,调用粗检测模块来实现目标的检测,并使用跟踪结果对检测结果进行修正及可信度判断。该思想的提出,不仅能保证目标检测与跟踪算法的高准确度,还能大幅提高算法的实时性。

1.1 粗检测

粗检测指一个笼统的概念,而不是具体的方法。它主要指一些简单、耗时少但能够大致定位出目标所属 ROI(region of Interesting)的方法。针对本文的手势检测,采用的粗检测方法包括肤色检测和搜索框检测。

研究表明,肤色信息在色调空间上比亮度空间能得到更好的区分^[15],但是 RGB 色彩空间的 3 个颜色通道都含有大量的亮度信息,所以,为了更好的将肤色信息提取出来,本文选择 HSV 色彩空间。

为了使肤色检测更加简单快捷,本文采用域值法:在 HSV 的每个通道,设定一个或者多个肤色范围的域值,当一个像素点同时满足每个通道的域值,则判为肤色点。本文采用 Sobottka 等^[16]的肤色检测阈值条件:

$$\phi = \begin{cases} 0 < H < 50, \\ 0.20 < S < 0.68, \\ 0.35 < V < 1.0, \end{cases}$$

其中, H , S 和 V 分别表示像素点在 HSV 空间的取值。满足上式阈值条件的像素点即属于肤色点,否则属于非肤色点。

$$\text{image}(i, j) = \begin{cases} \text{肤色点, 当 } \text{value}(i, j) \in \phi, \\ \text{非肤色点, 当 } \text{value}(i, j) \notin \phi, \end{cases}$$

其中, $\text{image}(i, j)$ 指图像中的像素点; $\text{value}(i, j)$ 为该像素点在 HSV 空间下的像素值。

粗检测模块的流程如图 2 所示。肤色检测结果与搜索框的融合规则为:若肤色检测框与搜索框之间的重叠区域大于可检测手势最小尺度,则取重叠部分为融合结果;否则,取搜索框为最终结果。

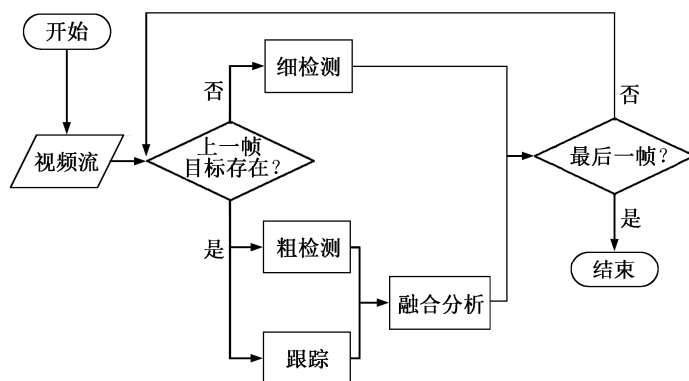


图 1 本文算法流程框图

Fig. 1 Flow chart of proposed algorithm

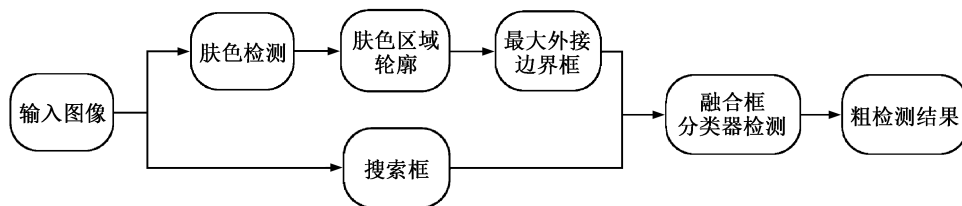


图2 粗检测模块流程图
Fig. 2 Flow chart of rough detection

1.2 细检测

细检测主要指一些高检测率的方法，对其耗时性等不作较高要求。

为了满足人机交互系统对检测的高准确度需求，本文针对手势检测的细检测模块，采用线下分类器的方法。为了保证检测结果的高准确度，选取最易于识别的两类手势：拳头和手掌。拳头手势用于手势跟踪及轨迹跟踪，使用拳头运动轨迹设置人机交互操控命令；手掌手势用于释放控制，即出现手掌手势表示人机交互命令输入结束。

本文采用 Chen 等^[10]的方法，首先对大量手势(拳头和手掌)和非手势样本提取 Haar-like 特征，使用 opencv 进行线下 Adaboost 训练，得到高检测准确度的手势(拳头和手掌)检测分类器。拳头手势样本尺寸为 18 像素×18 像素，训练得到的拳头分类器由 17 个强分类器和 385 个弱分类器组成；手掌手势样本尺寸为 18 像素×30 像素，手掌分类器由 28 个强分类器和 350 个弱分类器组成。程序调用细检测模块时，直接调用该分类器对图像进行手势

(拳头)检测和定位。细检测模块流程如图 3 所示。

2 搜索框

进行粗检测时，仅使用肤色检测对跟踪结果进行验证和修正是不够的。因为肤色检测结果范围有可能较大，会导致对跟踪结果修正不正确的情况，而这种不正确将影响到最终结果及对下一帧图像的跟踪。所以，基于视频中手势运动的连续性，本文提出搜索框的概念来完善跟踪结果的验证和修正，还可以进一步提高算法实时性。

对于视频图像中的物体，在连续两帧中，位置和尺度的变化不大。利用这个信息，就可以设定一个如图 4 所示的搜索区域。其中，灰色矩形框为目标框。黑色矩形框为需要搜索的范围。此处，黑色矩形框的大小为灰色矩形框的两倍。

在实验中，得到第 t 帧目标矩形框后，根据此矩形框确定下一帧待搜索的范围。这样可以减少搜索面积，节省时间，并且可以排除更多干扰区域，从而保证检测目标的准确性。

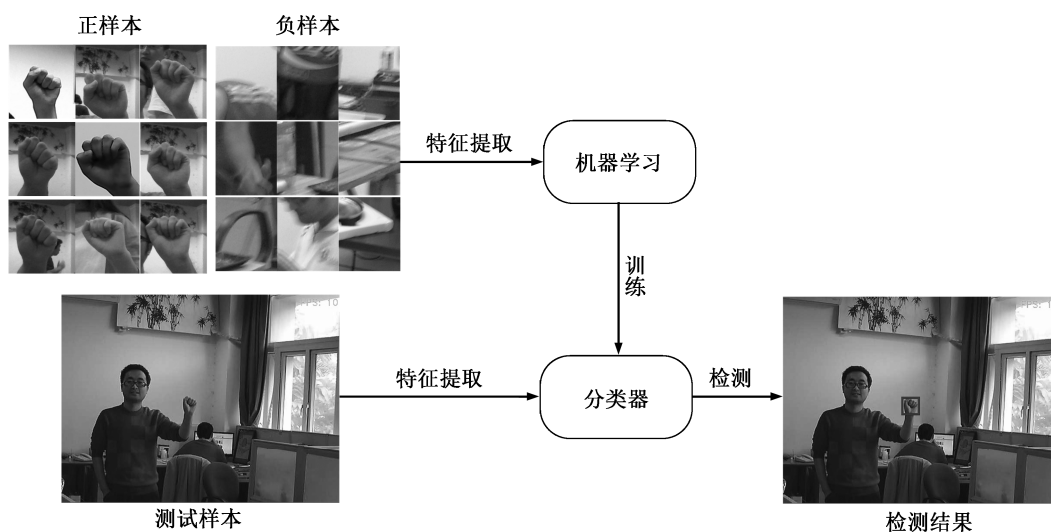


图3 细检测模块流程图
Fig. 3 Flow chart of fine detection



图 4 搜索框
Fig. 4 Searching-box

3 目标跟踪算法

本文提出的目标跟踪算法是将分情况检测思想、搜索框与光流跟踪方法相结合的一种算法。该算法使用从上一帧图像得到的目标所在位置作为初始信息,结合搜索框检测与光流跟踪,得到目标在当前帧图像中的位置。

3.1 光流跟踪

光流跟踪是一种高效的跟踪方法,但其耗时性较强。为了高效跟踪并解决耗时性的问题,本文采用在 TLD 算法中由 Kalal 等^[17-18]提出,Nebehay^[19]整理的对目标图像区域进行均匀取点跟踪的方法:1)对前一帧中目标图像块计算取点间隔,均匀地取约 10×10 个点;2)对取得的每个点 $P_{i,0}$ 进行前一帧到当前帧的光流跟踪;3)对跟踪得到的结果 $P_{i,1}$ 进行当前帧到前一帧的光流跟踪,得到跟踪结果点 $P_{i,2}$;4)选取有效跟踪点对 $\langle P_{i,0}, P_{i,1} \rangle$,统计点对的变换关系,计算目标在当前帧中的位置信息。

3.1.1 定义有效跟踪点

使用光流法对上一帧目标图像进行跟踪,定义有效跟踪点,对跟踪结果进行筛选,从而更加准确地估计目标在当前帧的大致位置。

参考前后向跟踪误差方法^[20],在跟踪过程中,用于第一次光流跟踪的初始点 $P_{i,0}$ 与经过两次光流跟踪后得到的跟踪点 $P_{i,2}$,二者组成的点对之间存在一定的误差。计算点对中两点之间的距离,作为点对的误差 d_i 。统计这些点对误差,并计算这些点对误差的均值 d_{avg} 。使用均值作为阈值,将点对误差小于阈值 d_{avg} 的所有点对对应的一次光流跟踪结果点作为可能的有效跟踪点。

$$d_i = \text{distance}(P_{i,0}, P_{i,2}),$$

$$d_{avg} = \frac{1}{n} \sum_{i=1}^n d_i,$$

其中, $P_{i,0}$ 为前一帧图像的均匀初取点。当然,为了避免极端性跟踪错误,算法约定:如果 d_{avg} 大于预定阈值 θ ,就认为这是一个不可信跟踪结果,而不给出任何结果。

为了有效地确定跟踪点是否有效,针对二次跟踪结果点对 $\langle P_0, P_2 \rangle$,分别以 P_0 和 P_2 点为中心,取大小为 11×11 的像素块 I_1 和 I_2 ,计算归一化相关系数(NCC):

$$NCC_i = \frac{1}{n-1} \sum_{x=1}^n \frac{(I_1(x) - \mu_1)(I_2(x) - \mu_2)}{\delta_1 \delta_2},$$

其中, δ_1 和 δ_2 分别表示 I_1 和 I_2 的标准差; μ_1 和 μ_2 表示 I_1 和 I_2 的均值。由上式可知,归一化相关系数的精度是像素级的。同理,计算所有点对的归一化相关系数的均值 NCC_{avg} ,舍弃 NCC 系数不大于 NCC_{avg} 的点对。

$$NCC_{avg} = \frac{1}{n} \sum_{i=1}^n NCC_i,$$

$$\langle P_{i,0}, P_{i,1} \rangle = \begin{cases} \text{有效跟踪点对, 当 } d_i < d_{avg}, d_{avg} < \theta, \\ \quad \quad \quad NCC_i > NCC_{avg}, \\ \text{无效跟踪点对, 其他。} \end{cases}$$

经过两轮筛选,得到满足上式要求的多组有效跟踪点对。

3.1.2 获取跟踪结果

使用对跟踪点的有效性定义,过滤得到有效的跟踪点对,从而得到从上一帧经一次光流跟踪到当前帧的有效跟踪点。使用最终过滤剩下的点来计算目标外接边界框的变换模型。

计算变换模型的方法是,计算跟踪前后所有有效点之间的距离,及其在跟踪前后的平均变化量,并将其作为尺度的变化量。其中 X 方向的位移通过计算全部点的水平位移均值得到, Y 方向的位移通过计算全部点的垂直位移均值得到。

3.2 检测与跟踪的融合

粗检测结果指的是融合肤色检测区域、搜索框得到缩小后的区域,然后对该区域使用分类器检测得到的结果。肤色检测结果与搜索框的融合方式是:计算肤色检测结果框与搜索框的重叠块信息,如果重叠面积不满足最小可用于手势检测的图像块条件(本文为 $18 \text{ 像素} \times 18 \text{ 像素}$),则取肤色检测结果

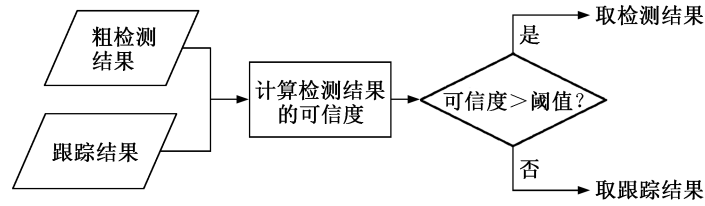


图 5 融合分析框图

Fig. 5 Method for merging the detecting and tracking results

块用于手势分类器检测；否则，取重叠部分图像块，用于手势分类器检测。算法的融合是对粗检测结果和跟踪结果进行综合分析，用于修正跟踪结果，或抛弃检测结果并重置跟踪初始状态信息。检测与跟踪的融合分析框图如图 5 所示。

具体实现步骤如下。首先，逐一计算粗检测与跟踪结果的重叠度，作为检测结果的可信度(如图 6 所示)，计算公式为

$$P = \frac{I_1 \cap I_2}{I_1 \cup I_2} = \frac{C}{I_1 + I_2 - C},$$

其中， I_1 为跟踪结果框； I_2 为检测结果框； C 为粗检测结果与跟踪结果的重叠部分。然后，取可信度最高的粗检测结果为最终检测结果。最后，根据最终检测结果的可信度来选择算法的最终结果。如果可信度 P 小于阈值 T (本文取值为 0.65)，则取跟踪结果为最终结果；否则，取检测结果为最终结果。如果跟踪模块没有跟踪结果，则调用细检测模块，取检测结果为最终跟踪结果，并作为下一次跟踪循环的初始量。

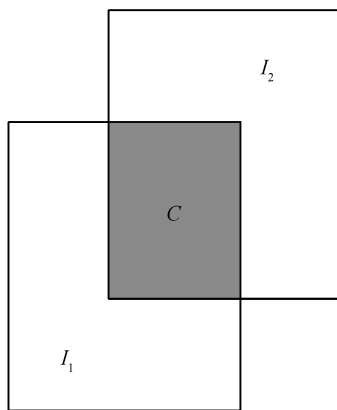


图 6 可信度计算示意图

Fig. 6 Chart for computing reliability

4 实验

为了测试算法的鲁棒性，使用该算法，对光照、背景复杂度、有无人为干扰、测试者与摄像头的距离等因素组合条件下拍摄的 16 个测试视频进行对比测试。

4.1 实验环境与方法

本文算法是在 VS2010 集成开发环境下，基于 OpenCV 用 C/C++ 开发，测试视频来源于普通摄像机拍摄，测试设备为普通笔记本电脑，其配置为 Intel(R) Core(TM) i5-3230M 处理器、2.60 GHz 主频，4 GB 内存，32 位 Win7 操作系统。

对约 1000 帧视频进行实验，并对实验结果进行统计分析。为了更好地衡量算法的性能，将算法的正确率、漏检率和误检率相结合作为衡量算法性能的标准。其中，算法正确率定义为算法成功检测跟踪到手势目标的数量与包含手势的帧数的百分比；漏检率定义为在包含手势的视频帧图像中，未检测到手势的总数量与视频帧图像总数量的百分比；误检率定义为在非手势区域检测到手势的数量与总视频帧数的百分比。

4.2 实验结果

4.2.1 分情况检测实验

为了验证使用分情况检测思想前后，算法性能的变化，对 1000 帧长度的视频进行实验测试，耗时及准确度如表 1 所示。由表 1 可知，使用分情况检

表 1 分情况检测前后算法性能对比
Table 1 Performance with or without distinguishing detection algorithm

分情况检测	总耗时/ms	平均单帧耗时/ms	准确度/%
使用	70489	70.489	91.30
不使用	211485	211.485	93.44

测后, 算法在耗时性上有很大的提升, 但对准确度的影响却很小。

4.2.2 搜索框实验

为了量化利用搜索框前后时间复杂度的变化, 进行了如下实验: 分别对 1000 帧视频图像, 使用搜索框前后进行耗时性分析实验; 对第 4 帧视频图像, 使用搜索框前后, 直接用分类器进行检测, 进行耗时性分析实验。

图 7 为使用搜索框后, 检测得到的结果。其中, 灰色框为检测结果框, 白色框为搜索框。图 8 为在不使用搜索框的条件下, 得到的检测结果, 如白色框所示。由图 7 和 8 可知, 使用搜索框前后检测结果基本一致。表 2 为采用搜索框前后算出耗时量的对比。从图 7 和 8 及表 2 可知, 通过使用搜索框缩小检测范围, 可以大大减少需要进行分类器检测的窗口数目, 缩短检测耗时, 且对检测结果的准确度影响较小。



图 7 使用搜索框的检测结果
Fig. 7 Result with searching-box



图 8 不使用搜索框的检测结果
Fig. 8 Result without searching-box

表 2 采用搜索框前后算法耗时量对比表

Table 2 Time-consuming with or without searching-box

A 搜索框	总耗时/ms	平均单帧 耗时/ms	单帧 纯耗时/ms	准确度/%
使用	71362	71.362	10	97.66
不使用	211485	211.485	197	99.13

4.2.3 算法性能测试实验

实验采集了 16 个测试视频, 采集环境及条件包括: 实验室室内; 白天/夜晚; 距离摄像头 3/4 m; 背景为均匀白墙/非均匀干扰; 有其他人的干扰/无人干扰。

图 9 为 16 个测试视频测试结果的正确率, 所有测试视频平均正确率为 97.36%。算法误检率统计结果如图 10 所示, 平均误检率为 1.93%。除了误检率, 作为衡量算法性能的另一重要指标, 就是漏检率。图 11 为 16 个测试视频的漏检率, 平均漏检率为 4.54%。

下面对算法的正确率进行分类对比分析。首先, 对白天与夜晚组进行对比统计, 得到图 12 所示的统计结果。结果显示, 算法针对白天的跟踪效果

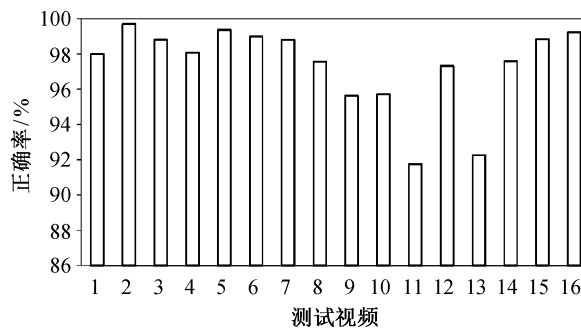


图 9 算法正确率
Fig. 9 Right rate of proposed algorithm

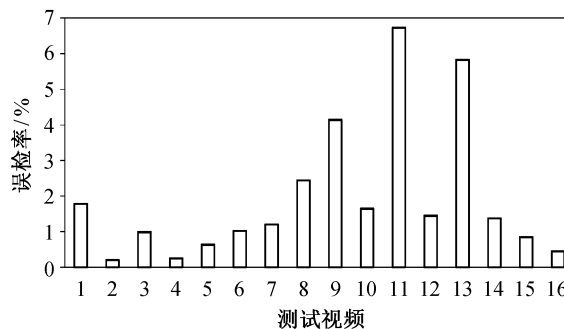


图 10 算法误检率
Fig. 10 Wrong rate of proposed algorithm

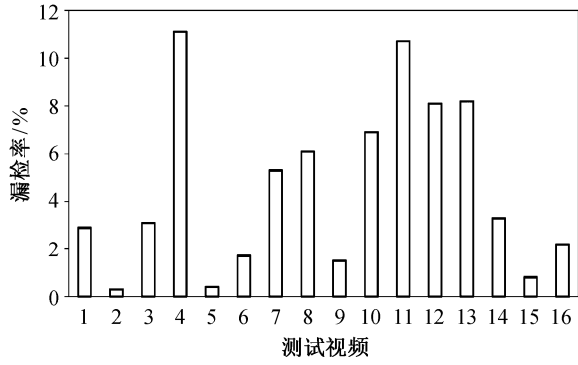


图 11 算法漏检率

Fig. 11 Missing rate of proposed algorithm

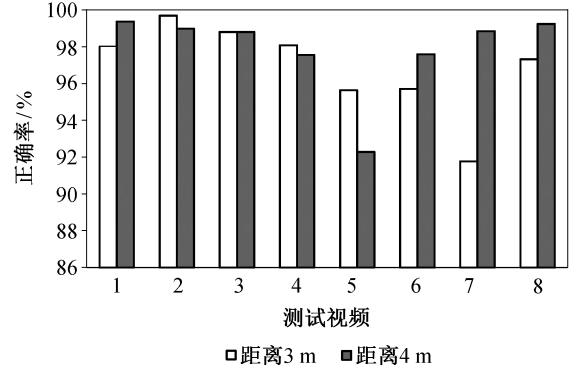


图 13 不同距离性能对比

Fig. 13 Comparison for different distance

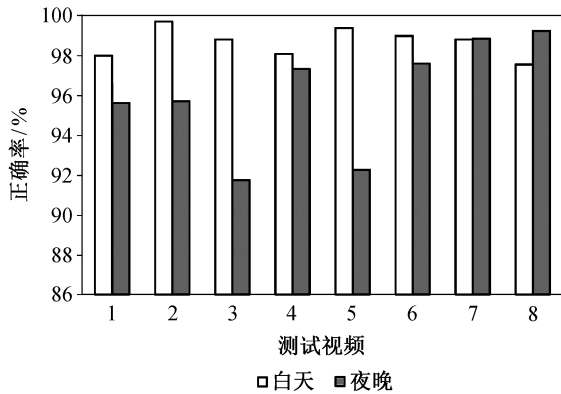


图 12 白天夜晚性能对比

Fig. 12 Comparison for day and night

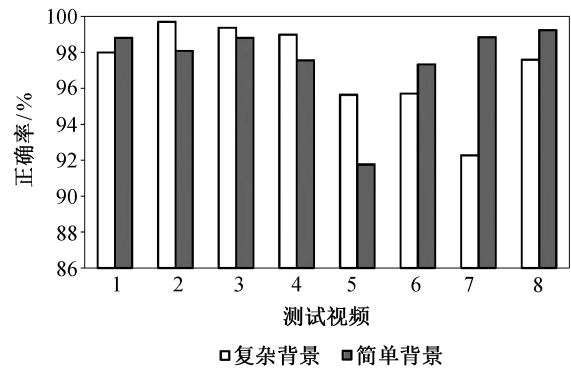


图 14 不同背景性能对比

Fig. 14 Comparison for different background

略优于夜晚。其次，将 16 个测试视频分为距离 3 m 和距离 4 m 两类测试视频组，对比结果如图 13 所示。测试结果表明，本算法针对两类视频的综合跟踪率虽然存在个体差异，但总体上大致相当。接着，把测试视频分为复杂背景和简单背景两类进行对照统计，统计结果如图 14 所示。其中，测试视频的简单背景指的是背景为颜色均匀的白墙，复杂背景指的是背景中存在书架、图画、器具等其他干扰物件。测试结果显示，两类统计结果存在某些个别差异，但平均正确率相差不大，详见表 3。最后，分为有人干扰组和无人干扰组进行对比统计，结果如图 15 所示。综合统计显示，无人干扰组算法效果比有人干扰组好。

综上所述，并结合平均正确率统计(表 3)，得出结论：针对各类测试视频组，算法测试正确率差异不大，平均正确率高达 97.36%，说明算法的抗干扰能力强，算法具有很好的鲁棒性。

4.2.4 其他算法性能对比实验

对 1000 帧视频分别用本算法 TDD (Tracking

表 3 对比组平均正确率比较

Table 3 Comparison of right rates for different conditions

对比因素	前 8 组平均 正确率/%	后 8 组平均 正确率/%
白天/夜晚	98.66	96.05
距离 3/4 m	96.88	97.83
复杂/简单背景	97.16	97.55
有/无人干扰	96.69	98.02

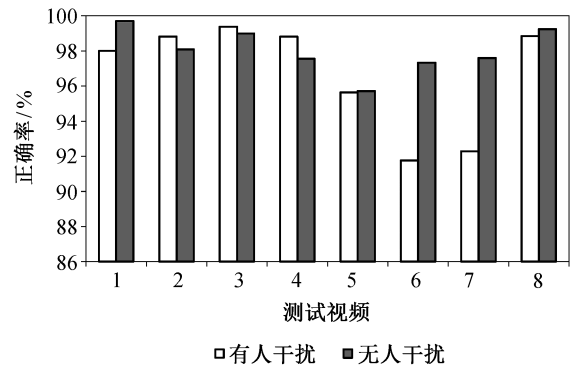


图 15 有无人干扰性能对比

Fig. 15 Comparison of performance with or without disturbing by person

表 4 本算法与 TLD 算法性能对比
Table 4 Comparison of performance for TLD and proposed algorithm

算法	速率/(帧·秒 ⁻¹)	正确率/%
TDD	19.42	98.30
TLD	1.50	77.47
CT	14.79	83.02
MS	7.96	31.00
CMT	1.48	94.12

and Distinguishing Detection)、TLD (Tracking, Learning and Detection)、CT (Compressive Tracking)、MS (MeanSift) 和 CMT (Consensus-based Matching and Tracking) 算法进行对比实验, 得到算法性能对比如表 4 所示。

考虑到其他算法无法对手势进行分辨, 也为了更好地说明 TDD 算法的优越性能, 在进行数据统计时, 只要其他算法跟踪到手部, 无论是拳头或手掌, 或二者都不是, 本文都归为正确跟踪。将其他算法的初始跟踪框手动设置为本算法初始检测到的手势位置。

由表 4 可知, 基于 TLD 算法设计的 TDD 算法在耗时性上具有明显优势, 与 TLD 算法在帧率上存在数量级的差异, TDD 算法使用搜索框检测方法对跟踪结果进行分析修正, 并用于下一帧目标跟踪, 所以在正确率上, 也明显优于其他算法。因此本算法的搜索框和分情况检测思想, 可以弥补丢弃 TLD 算法在线学习模块造成的损失。

5 结论

本文提出分情况检测思想和用于跟踪的搜索框概念, 并将其运用到手势的检测和跟踪中, 实现了手势的高效实时跟踪, 为基于视频图像的人机交互方式提供了一种解决思路与方案。本文还进行了多因素对比测试实验和多算法性能对比实验, 结果显示, 本算法对图像手势的平均检测跟踪正确率可以达到 97.36% 以上, 且平均漏检率在 5% 以下, 对各种外界因素具有较好的鲁棒性, 满足预期高准确度需求。本算法的运行速度达 19.42 帧/秒, 满足预期实时性需求。与参考 TLD 算法相比, 本算法在处理速度上有数量级的改善。与其他对比算法相比, 本算法在耗时性与准确度的平衡上做得更好, 针对人机交互应用, 本算法存在明显优势。但是, 人机

交互系统是一个要求苛刻的系统, 所以我们下一步的工作将改进检测模块的细检测方案, 进一步提高检测结果的准确度, 实现初始化的更高准确度。

致谢 感谢北京大学深港产学研基地和北京大学深圳研究生院移动视频研究中心提供的支持与帮助。感谢深圳迈乐数码为本研究提供的嵌入式样机, 为研究的产业化尝试提供了难得的条件。

参考文献

- [1] Rautaray S S, Agrawal A. Vision based hand gesture recognition for human computer interaction: a survey. *Artificial Intelligence Review*, 2015, 43(1): 1–54
- [2] Chaudhary A, Raheja J L, Das K, et al. Intelligent approaches to interact with machines using hand gesture recognition in natural way: a survey. *Int J Comput Sci Eng Survey*, 2011, 2(1): 122–133
- [3] Zhu Y, Xu G, Kriegman D J. A real-time approach to the spotting, representation, and recognition of hand gestures for human–computer interaction. *Computer Vision and Image Understanding*, 2002, 85(3): 189–208
- [4] Fang Y, Wang K, Cheng J, et al. A real-time hand gesture recognition method // *IEEE International Conference on Multimedia and Expo IEEE*. Beijing, 2007: 995–998
- [5] Rautaray S S, Agrawal A. A novel human computer interface based on hand gesture recognition using computer vision techniques // *Proceedings of the First International Conference on Intelligent Interactive Technologies and Multimedia*. New York: ACM, 2010: 292–296
- [6] Brethes L, Menezes P, Lerasle F, et al. Face tracking and hand gesture recognition for human-robot interaction // *IEEE International Conference on Robotics and Automation (ICRA)*. New Orleans, 2004: 1901–1906
- [7] Li Y T, Wachs J P. Recognizing hand gestures using the weighted elastic graph matching (WEGM) method. *Image and Vision Computing*, 2013, 31(9): 649–657
- [8] Hsieh C C, Liou D H, Lee D. A real time hand gesture recognition system using motion history image // *2nd IEEE International Conference on Signal Processing Systems ICSPS*. Dalian, 2010: 394–398
- [9] Song Y, Demirdjian D, Davis R. Continuous body and

- hand gesture recognition for natural human-computer interaction. *ACM Transactions on Interactive Intelligent Systems* *TIIS*, 2012, 2(1): Article 5
- [10] Chen Q, Georganas N D, Petriu E M. Hand gesture recognition using Haar-like features and a stochastic context-free grammar. *IEEE Transactions on Instrumentation and Measurement*, 2008, 57(8): 1562–1571
- [11] Lee D H, Hong K S. Game interface using hand gesture recognition // 5th IEEE International Conference on Computer Sciences and Convergence Information Technology (ICCIT). Seoul, 2010: 1092–1097
- [12] Roomi S, Priya R J, Jayalakshmi H. Hand gesture recognition for human-computer interaction. *Journal of Computer Science*, 2010, 6(9): 994–999
- [13] Garg P, Aggarwal N, Sofat S. Vision based hand gesture recognition // *Proceedings of World Academy of Science: Engineering & Technology*, 2009, 48(49): 972–977
- [14] Pereira F G, Vassallo R F, Salles E O T. Human–robot interaction and cooperation through people detection and gesture recognition. *Journal of Control, Automation and Electrical Systems*, 2013, 24(3): 187–198
- [15] Yang J, Lu W, Waibel A. *Skin-color modeling and adaptation*. Berlin: Springer, 1997
- [16] Sobottka K, Pitas I. A novel method for automatic face segmentation, facial feature extraction and tracking. *Signal Processing: Image Communication*, 1998, 12(3): 263–281
- [17] Kalal Z, Mikolajczyk K, Matas J. Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012, 34(7): 1409–1422
- [18] Kalal Z, Mikolajczyk K, Matas J. Face-TLD: tracking-learning-detection applied to faces // 17th IEEE International Conference on Image Processing ICIP. Hong Kong, 2010: 3789–3792
- [19] Nebel G. Robust object tracking based on tracking-learning-detection [D]. Vienna: Vienna University of Technology, 2012
- [20] Kalal Z, Mikolajczyk K, Matas J. Forward-backward error: Automatic detection of tracking failures // 20th IEEE International Conference on Pattern Recognition ICPR. Istanbul, 2010: 2756–2759