

# The Implementation of the Micro-Gesture Recognition on the Mobile Web

Meiyu Zhang, Jiacheng Chen, Wei Zhang, Chengfeng Jian

Computer Science and Technology College, Zhejiang University of Technology, Hangzhou Zhejiang  
Email: jiancf@zjut.edu.cn

Received: Jan. 7<sup>th</sup>, 2016; accepted: Jan. 26<sup>th</sup>, 2016; published: Jan. 29<sup>th</sup>, 2016

Copyright © 2016 by authors and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

For the requirement of Micro-gesture recognition on mobile terminal, this paper introduces a Micro-gesture recognition system based on HTML5, Javascript and webRTC. It obtains video through the web camera, then extracts the hand from the video, and divides the hand into palm and fingers. The position of palm would indicate the location of whole hand, while the fingers would indicate the information of gesture after skin color extraction, region segmentation and contour extraction. Compared with the predefined command, the system would work according to the current command as the gesture.

## Keywords

Web Client, webRTC, Micro-Gesture Recognition

---

## 面向移动WEB的微手势识别技术及系统实现

张美玉, 陈嘉诚, 张伟, 简琤峰

浙江工业大学计算机学院, 浙江 杭州  
Email: jiancf@zjut.edu.cn

收稿日期: 2016年1月7日; 录用日期: 2016年1月26日; 发布日期: 2016年1月29日

---

## 摘要

针对面向移动终端的微手势识别需求, 本文介绍了一种基于HTML5、Javascript和webRTC框架的移动

web端的微手势识别系统。系统通过移动终端摄像头获取视频信息，并从视频中提取人手，再将手分为手掌域和手指域。其中手掌域用于定位手部位置，手指域通过肤色提取、区域分割、轮廓提取等操作提取手势中的手指信息。通过与手势预定义进行解析对比，从而响应相应的操作行为。

## 关键词

移动WEB, webRTC, 微手势识别

## 1. 引言

移动硬件的普及，使得消费者在日常生活中所使用的应用渐渐集中到移动设备上，从而移动互联网应用的开发逐渐成为趋势。而随着移动应用的发展，消费者已经不再满足于点按的交互方式。而当手势识别技术被应用到移动端时，成本大小和便捷与否就应该被优先考虑[1]-[5]。从数据的采集方式考虑，传统的采用数据手套等的方式由于成本较高以及使用时的不便，所以并不适用于移动平台。而 Kinect 等三维信息采集设备虽然具有极大的发展前景[1] [6]，但就目前来说，并不适合用在移动平台上。由此，本文所介绍的系统只使用移动设备自带的网络摄像头获取视频信息。

当前市场上的移动平台并不统一，仅考虑身边最为常见的就有苹果和安卓这两大系统。这使得开发者有时需要针对不同系统开发不同版本的应用[7]。所以，为了使系统具备跨平台工作的能力，本系统以移动 WEB 应用的方式进行实现。它将与运行的移动平台无关，切实解决平台局限性问题。

此外，现有手势识别往往会忽略了对于较小动作手势的识别，这也是本系统所重视的问题之一。

## 2. 识别流程及关键技术

本文将介绍一种基于 web 的微手势识别系统。微手势识别的重点，在于对手指位置信息的确定。手指的外观形态随动作而变，而手掌的外观形态基本处于相似形状，故系统将手分为手指域和手掌域。系统对大量样本进行训练，得到基于 Haar 特征的级联分类器。这个分类器可以用来定位视频画面中的手掌位置，即定位到整个手势的位置。然后，系统获取手指的指尖和重心的位置，并通过追踪手指重心的前后位置差异，对手势进行解析。

基于移动 web 的微手势识别的流程如图 1 所示。首先通过网络摄像头获取手势图像；再使用已经训练完成的分类器，通过 objectdetect.js 对获取的图像进行对手掌区域的位置确定；然后根据手掌域位置的确定，获得手指域位置；接着根据 YCrCb 颜色空间的肤色模型[8]，对图像进行二值化处理，而后再经过一定的形态学处理，去除一定噪声；最后对已分割出的图像，采取微手势识别算法，提取出其中的手掌与手指重心，解析手势信息。

### 2.1. 图像采集与定位

图像采集基于 Google 提出的 WebRTC 框架，它基于 HTML5 标准，使用 Javascript API 来实现对摄像头的调用和浏览器的无插件视频播放。目前的主流浏览器，如 Chrome、Opera 以及 Firefox 等的最新稳定版本都支持这一标准。

对于网络摄像头所获取的视频信息，系统利用<canvas>获取视频的每一帧图像，用来检测视频中的手势。而在对视频中的手势进行检测和定位时，系统采用开源项目 objectdetect.js 来实现。它主要采用基于 Haar 特征的级联分类器对目标进行检测。

这里，本文的目标图像为手掌图像，通过对手掌域的定位，获取手势的大致位置。在大致位置(兴趣

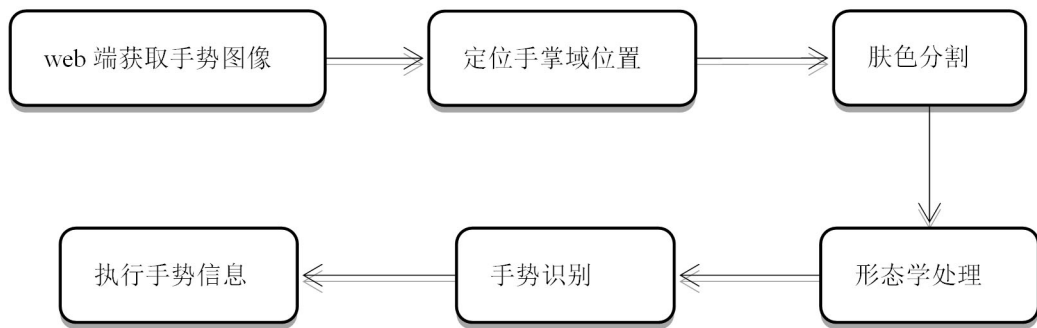


Figure 1. Micro-gesture recognition process

图 1. 微手势识别的流程

区域 ROI) 的确定之后, 后续的图像处理工作将被定在这块区域内。这样不仅排除了非手势区域的干扰信息, 并且大大加快了处理速度, 提高了系统的工作效率。

## 2.2. 肤色分割

相对 RGB, YCrCb 颜色空间可以将色度和亮度分离, 从而在肤色分割的过程中降低亮度的干扰。其中,  $Y$  表示亮度,  $Cr$  表示光源中红色分量,  $Cb$  表示光源中蓝色分量。RGB 向 YCrCb 的转换公式如下:

$$\begin{aligned}
 Y &= 0.2990R + 0.5780G + 0.1140B; \\
 Cr &= 0.5000R - 0.4187G - 0.0813B + 128; \\
 Cb &= -0.1687R - 0.3313G + 0.5000B + 128;
 \end{aligned}$$

一般来说, 根据某点  $Cr$  和  $Cb$  的值满足  $133 \leq Cr \leq 173$  且  $77 \leq Cb \leq 127$ , 那么该点可以被判断为肤色点, 否则不是。根据这个标准, 肤色区域像素点的灰度值被设为 255, 非肤色区域像素点的灰度值被设为 0, 生成肤色二值图像。

## 2.3. 手指识别

手指识别的作用在于对所得的手势图像进行解析, 获得其所表达的手势信息。它是整个流程中最重要的部分。目前手势识别的方法很多, 大致可分为基于模板匹配的方法和基于机器学习的方法。本系统的检测部分属于机器学习类算法。整个过程包括悬停判断, 肤色区域编号, 轮廓统计, 重心相对位置计算, 手势信息确定。其中, 重心相对法利用手掌域重心, 手指域重心的相对位置, 判断手指相对于手掌的运动情况, 再结合手指域重心的前后两者变化, 从而解析出手势信息。

## 3. 手势识别算法的研究

本系统的重点在于对手指的识别与追踪, 需要具有一定的高效、实时性。

### 3.1. 算法流程

算法流程图 2 所示。

### 3.2. 算法实现

#### 3.2.1. 获取手掌域重心

在处理过的肤色二值图像中, 肤色像素的灰度被设置为 255。这意味着手掌域内的肤色像素的  $(x, y)$  坐标点已经被确定, 然后手掌域的重心可以根据下方的公式算出。

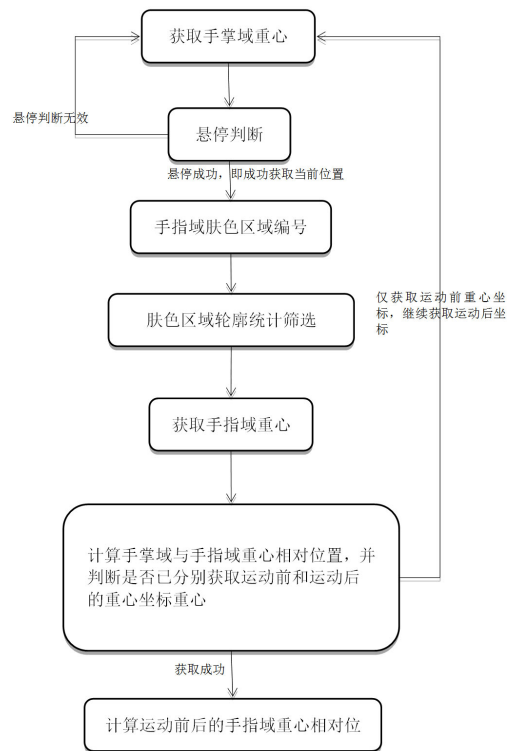


Figure 2. Algorithm flow

图 2. 算法流程

$$C(x, y) = \left( \frac{\sum_{i=0}^{N-1} P_i(x)}{N}, \frac{\sum_{i=0}^{N-1} P_i(y)}{N} \right)$$

其中,  $C(x, y)$  为重心坐标,  $P_i(x)$  为肤色像素点的横坐标,  $P_i(y)$  为肤色像素点的纵坐标,  $N$  为肤色像素点的总数目。

### 3.2.2. 悬停判断

web 端的手势识别, 在追求高效、实时性的同时, 会失去相应的准确性, 为了达到效率与精度的相平衡, 采用了一种悬停判断的方法。此方法的前提是手掌域的图像已经被识别, 且手掌域重心已经被获取到。将此时手掌重心的坐标记为  $(x_0, y_0)$ 。随着时间推移, 视频的每一帧的重心坐标可以组成一个序列, 即  $(x_t, y_t), t = 0, 1, \dots, 50$ 。若其中有 30 个坐标处于  $(x_0, y_0)$  的邻域内, 则该手势在这个位置保持了一定时间。它所代表的含义即为“选中, 执行下一步操作”。

### 3.2.3. 肤色区域编号

在经过肤色二值化操作后, 肤色像素在图中呈区域的形态。为了有效地处理其中的肤色信息, 系统对肤色像素区域进行编码, 从中提取出所需要的区域, 并排除不需要的区域。编码过程的大体准则为: 首先初始化整个手势图像的值; 接着按照从左到右、从上到下的方式对肤色像素区域进行编码。

#### (1) 初始化编号值

图像区域如图 3 所示: 其中 Total 为整个图像区域; Pre 为事先获得的手势图像区域; Ext 为将手势图像区域边界延伸 2 个像素大小所得的图像区域。

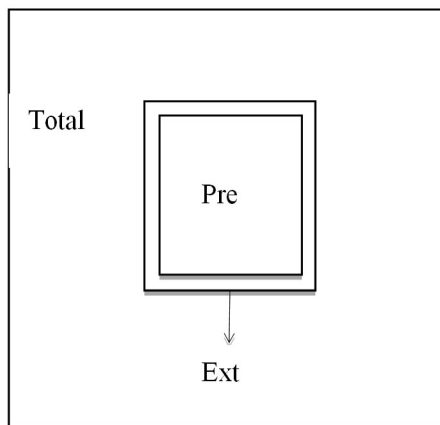


Figure 3. Area sketch map  
图 3. 区域示意图

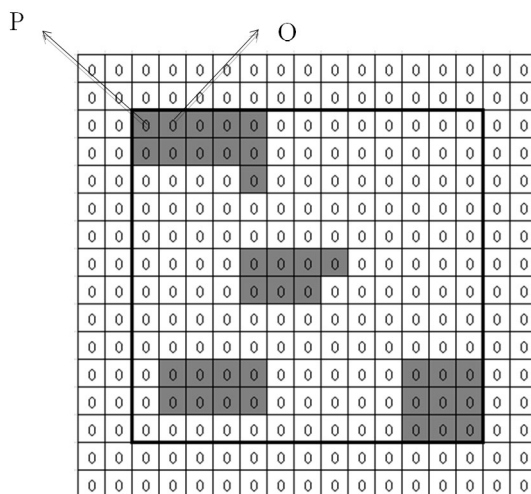


Figure 4. Encoding initialization  
图 4. 编码初始化

图像初始化是将 Ext 区域内的所有像素点的编号值初始化为 0，如图 4 所示：其中黑色边框为 Pre 区域，灰色框代表肤色像素点，白色框代表非肤色像素点

#### (2) 编号方法

如图所示，整个编码过程依照从左到右、从上到下的原则进行。假设肤色点的坐标为  $(x, y)$ ，码值为  $num$ 。系统需要判定  $(x-1, y)$ ， $(x-1, y-1)$ ， $(x, y-1)$  和  $(x+1, y-1)$  这四个点是否已被更改编码：若四个点的码值都是 0，则认定该像素点所处区域未被重新编码，然后将当前点的码值重新赋值为  $num = num + 1$ ；若其中至少有一个点被重新编号，则当前点的值应被赋值为那个被重新编号的码值。根据以上原则，如图 4 中所示，首先遇到的肤色像素点为 P 点，当前  $num = 0$ 。因为 P 点四个方向上的点编号值均为 0，故 P 点的编号值应为  $num = num + 1$ ，即为 1。同理，对于图中 Q 点，系统需判定 Q 点的左前方、左上方、上方、友上方四个点是否被重新编号，由于 P 点为 Q 点的左前方点，又因为 P 点已被重新编号为 1，那么 Q 点的编号值应与 P 点的编号值相同，即仍为 1；最终，图 4 中肤色区域编码后的结果如图 5。

#### 3.2.4. 轮廓统计

在不同区域的肤色被用不同的数字进行了编号之后，由于肤色分割存在一定缺陷，它会将与肤色相

近的颜色也误认为是肤色。所以在经过肤色二值化后，图像中仍会存在或大或小的干扰区域。而轮廓统计目的在于筛选出系统所需要的肤色轮廓。

此步骤首先对编号过的各个区域进行编号值统计，分别统计各个编号值的数量，通过数量上的大小，可以得知这个区域的大小。根据实验统计，手指轮廓的大小，所占目标区域的比例约为 5%~15%。然后根据轮廓大小，系统就可以得到所感兴趣的肤色轮廓区域，并排除了过大或过小的干扰区域。

### 3.2.5. 重心相对位置识别法

经过前面几个步骤，系统已经可以得出手指区域。与获取手掌域重心一样，系统采用相同的方法获取手指域重心坐标，并将重心点抽象出来，如图 6 所示。其中“状态一”指“手部运动开始前且通过悬浮判定成功获取重心坐标的状态”，“状态二”指“手部运动结束后且通过悬浮判定成功获取重心坐标的状态”。

### 3.2.6. 算法具体实现方法

#### (1) 构建重心坐标三角形

过掌心 A 点做一条垂线 L，过手指重心 B 点做 L 的垂线 L'且两条线交于点 C，得  $\angle ABC$ ，记 A 角度数为  $\theta$ 。同理可得  $\angle A'B'C'$ ，记 A'角度数为  $\theta'$ 。如图 7 所示

#### (2) 计算手指长度

在手指域部分，假设手指区域纵坐标最小点为 E，纵坐标最大点为 F，分别过 E、F 点做水平线，记水平线间距为 d，如图 8 所示。手指分为伸长和弯曲情况，弯曲情况下的手指长度大体为伸长情况下的二分之一。故根据这一点，假设状态一的手指长度为  $d_1$ ，状态二的手指长度为  $d_2$ ，max 为  $d_1$ 、 $d_2$  中的较

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	2	2	2	2	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	2	2	2	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	3	3	3	3	0	0	0	0	0	0	4	4	4	0	0	0	0
0	0	0	3	3	3	3	0	0	0	0	0	0	4	4	4	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	4	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 5. Encoding result  
图 5. 编码结果



Figure 6. States of hand movement  
图 6. 手部运动状态

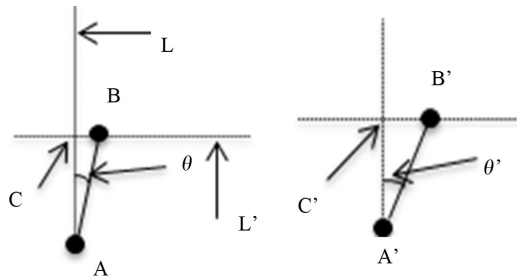


Figure 7. Formalize the states of hand movement  
图 7. 手部运动状态形式化

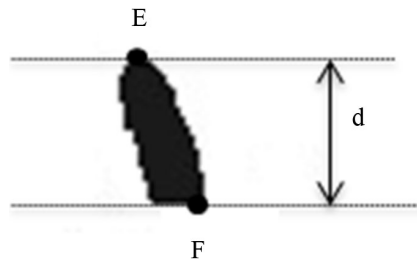


Figure 8. Finger bending  
图 8. 手指弯曲

大者，设  $m = |d_1 - d_2|$ 。

若  $m \geq 0.4 \max$ ，则认为手指前后发生了弯曲动作；若  $m < 0.4 \max$ ，则认为手指未进行弯曲动作，此时，系统需要从角度变化方面入手来解析手指的运动信息

(3) 计算角  $\theta$ 、 $\theta'$

通过事先构建的坐标三角形，系统可以利用三角信息计算出相应的夹角。假设 A 点坐标为  $(x_1, y_1)$ ，B 点坐标为  $(x_2, y_2)$ ，C 点坐标为  $(x_1, y_2)$ ，由此可得：

$$\theta = \arccos\left(\frac{y_1 - y_2}{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}}\right), \text{ 同理可得 } \theta'.$$

若  $|\theta - \theta'| \leq 10^\circ$ ，则认为手指前后未进行左右摆动，并且事先已知手指前后长度未改变，此时可认为手部微小动作前后未发生改变。

当  $|\theta - \theta'| > 10^\circ$ ，若  $\theta > \theta'$ ，则认为手指部分进行了向右的摆动运动；若  $\theta < \theta'$ ，则认为手指部分进行了向左的摆动运动。

综上，系统得到了手指的 4 个运动情况：弯曲、保持、向左和向右。然后系统将手指的 4 个动作与手掌部重心坐标结合起来，就可以解析出多种事先定义的手势信息。

## 4. 系统示例

### 4.1. “复制”命令

当视频中手指做一次向左移动，系统解析为开启矩形复制功能，如图 9。

### 4.2. “移动”命令

当视频中手指做一次向右运动时，系统解析为开启拖动矩形的功能，然后矩形随手移动而移动，如图 10。

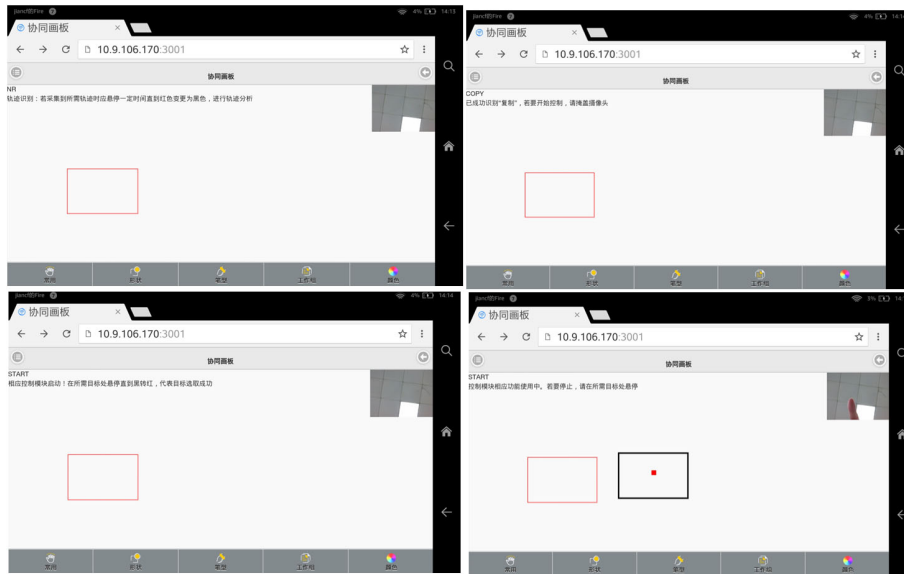


Figure 9. “copy” command  
图 9. “复制”命令

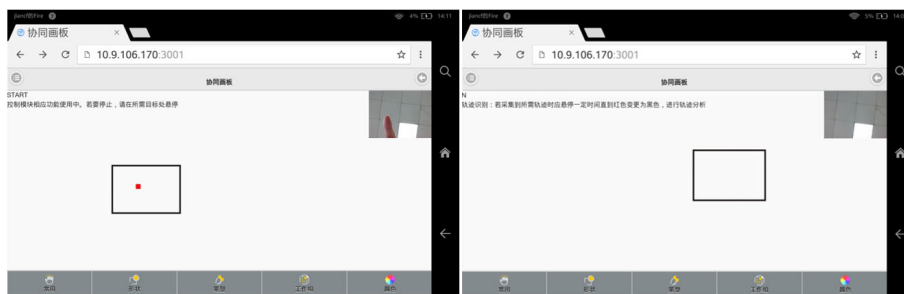


Figure 10. “move” command  
图 10. “移动”命令

## 5. 结束语

本系统通过网络摄像头获取手势视频, 使用基于 Haar 特征的级联分类器定位手掌位置, 再通过肤色分割、形态学处理等方法获取手部图像, 然后根据手掌和手指的信息分析获得当前手势所代表的命令。系统从数据采集到获取命令的整个过程, 都体现了低成本、高便捷性等特点, 对于移动设备具有较好的适用性。

## 基金项目

浙江省科技厅公益计划项目: 支持移动 WEB 手势识别的草图协同创意设计平台(No.2014C31081)资助。

## 参考文献 (References)

- [1] Chen, L.C., Wang, F., Deng, H. and Ji, K.F. (2013) A Survey on Hand Gesture Recognition. *International Conference on Computer Sciences and Applications*, Wuhan, 14-15 December 2013, 14-15. <http://dx.doi.org/10.1109/csa.2013.79>
- [2] Panwar, M. (2012) Hand Gesture Recognition Based on Shape Parameters. *International Conference on Computing, Communication and Applications*, Dindigul, 22-24 February 2012, 22-24. <http://dx.doi.org/10.1109/iccca.2012.6179213>
- [3] Yu, C.L., Wang, X., Huang, H.J., et al. (2010) Vision-Based Hand Gesture Recognition Using Combinational Features. *2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, Darmstadt, 15-17 October 2010, 543-546. <http://dx.doi.org/10.1109/IIHMSP.2010.138>



- 
- [4] Mei, K.Z., Xu, L., Li, B.L., *et al.* (2015) A Real-Time Hand Detection System Based on Multi-Feature. *Neurocomputing*, **158**, 184-193. <http://dx.doi.org/10.1016/j.neucom.2015.01.049>
- [5] 张秋余, 王道东, 张墨逸, 刘景满. 基于特征包支持向量机的手势识别[J]. 计算机应用, 2012, 32(12): 3392-3396.
- [6] Raheja, J.L., Chaudhary, A. and Singal, K. (2011) Tracking of Fingertips and Centers of Palm Using KINECT. 2011 *Third International Conference on Computational Intelligence Modelling Simulation*, Langkawi, 20-22 September 2011, 248-252. <http://dx.doi.org/10.1109/CIMSim.2011.51>
- [7] 赵健, 张冬泉. 基于 OpenCV 的数字手势识别算法[J]. 计算机应用, 2013, 33(S2): 193-196.
- [8] 朱志亮, 刘富国, 陶向阳, 刘晓山. 基于积分图和粒子群优化的肤色分割[J]. 计算机工程与应用, 2014(21): 171-174.