Neurocomputing 158 (2015) 184–193

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A real-time hand detection system based on multi-feature

Kuizhi Mei*, Lu Xu, Boliang Li, Bin Lin, Fang Wang

Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, PR China

ARTICLE INFO

Article history:
Received 29 September 2014
Received in revised form
2 December 2014
Accepted 25 January 2015
Communicated by Feiping Nie
Available online 7 February 2015

Keywords:
Multi-feature
Hand detection
Adaboost

ABSTRACT

This paper describes a real-time hand detection system which can reach high speed and accuracy. The system is based on Gentle Adaboost and cascade classifier. To improve the performance of the system, three efficient features are selected to describe the visual properties of human hands. In addition, the detection is accelerated due to several optimization methods, including the method for fast calculation of HOG features, improved cascade classifier and skin-color pre-detection. Experiments were performed on our self-constructed dataset, the results showed that the detection rate of the system can reach 0.889 while the false rate is 0.010 at the speed of 32.6339 ms per frame on a Intel Core i5-2400 CPU running at 3.1 GHz.

1. Introduction

Gesture recognition provides a new approach for human–computer interaction and it is more natural and convenient than traditional ways. A real-time hand detection system with high accuracy is the prerequisite of gesture recognition. However, the task of hand detection is challenging, because of changing illumination, cluttered backgrounds and the diversity of the hand gestures. In contrast to faces, which have defined shapes, human hand have more than 25 degrees of freedom. Thus, hand detection is more difficult than face detection [1].

Feature selection is important in many machine learning applications [2], of which hand detection is not an exception [3]. Due to the diversity of hand motions, it is difficult to describe hand sufficiently with a single feature. In [4], the author compared two commonly used features: Haar-like and HOG. It shows that the Haar detector can detect hands roughly twice as fast as the HOG detector. However, since the length of a Haar feature vector is 30 times over that of a HOG feature vector, more memory space is needed for a Haar feature. Because different features describe different aspects of the visual characteristics [5], the method of multi-feature gains a good performance in many fields [6]. Zhou et al. have proposed a novel interactive segmentation method based on multiple-feature fusion [7]. Ma et al. have focused on the event detection of real-world videos combined with multiple features [8]. Yang et al. have presented a new algorithm, namely multi-feature learning via hierarchical regression for multimedia semantics understanding [9]. In [10], the single-sample face recognition problem has been solved by a multi-feature multi-manifold learning method, which have demonstrated to be efficient on many datasets.

Based on the observations, this paper puts forward a robust and real-time hand detection system based on multi-feature for interacting with smart devices, such as smart TVs and smartphones. We mainly focus on this task in indoor environments, where cluttered backgrounds and changing illumination may exist. After analyzing the components of hand, we select three efficient features to describe them. To implement the system, we use several methods, including the fast calculation of HOG, the "early termination mechanism" and the merging algorithm in detection. Furthermore, in order to improve the speed of detection, several optimization methods are proposed, including piecewise cascade and skin-color pre-detection. Then we compare the accuracy and speed among single feature systems, multi-feature systems and the optimized system. Finally, we test our method on the NUS hand posture dataset [11].

The remainder of this paper is organized as follows. In Section 2, the related works are provided. The framework of the system is given in Section 3. We analyze the features and propose the multi-feature method in Section 4. In Section 5, the implementations of training and detection are described and experimental results are displayed. Then the optimizations of the system are presented along with the results in Section 6. Finally, conclusions are drawn in Section 7.

2. Related works

Several hand detection and gesture recognition systems have been proposed so far. The early systems that are based on the gloves can achieve a good performance in terms of detection speed

* Corresponding author. Fax: +86 29 82668672.
E-mail address: meikuizhi@mail.xjtu.edu.cn (K. Mei).

http://dx.doi.org/10.1016/j.neucom.2015.01.049
0925-2312/

and accuracy [12,13]. Unfortunately, the use of additional equipment makes the system expensive and inconvenient. The later hand detection systems based on computer vision can be categorized into two classes according to the dimensions: (a) monocular vision based and (b) multi-vision based systems. Different from the monocular vision based systems that can only achieve 2-D information, the multi-vision can acquire depth information with depth sensors [14–16], such as infrared camera and kinect. Therefore they can achieve accurate modeling of movements and shapes. On the other hand, the multi-vision based system is complex in calculation and the costs of the depth equipment make the technology far away from the masses.

Different researchers have put efforts on monocular vision based hand detection systems so far. The key of these systems is to find out the specific features for human hands that can distinguish them from the chaotic environment. In [17], the authors explored the potential of four features: color, temporal motion, gradient norm, and motion residue for hand detection, as well as the potential of combinations of those four features. Skin color information can segment hands from the background [18,19]. However, it is not reliable when the light is insufficient or there are other skin-colored objects in the background. In [20], a model has been proposed with the integration of image saliency and skin information which greatly improved the detection accuracy of skin modes. However, it is not efficient enough when dealing with other body parts in the image (such as faces and arms). Motion flow models detect hands by segmenting moving objects from the captured images [21,22], but they are not applicable to non-stationary cameras. Athitsos et al. proposed a method for detecting shapes of variable structures in images with clutter by introducing the Hidden State Shape Models (HSSMs). It can detect and recognize hand shapes with high accuracy. However, the speed is not mentioned in the paper [23].

After the successful real-time face detection using Haar-like features and boosted classifiers proposed by Viola and Jones [24], many researchers have been inspired to employ this particular algorithm for hand detection. Chen et al. used the extended Haar-like features with Adaboost in training and parallelized the detectors for recognition of different gestures. The system performs well under the laboratory conditions, but still uncertain in dynamic environments [25]. In [26], a detector which use frequency spectrum analysis was trained. It achieved a detection rate of 92.23% with a low false rate, in the condition that the hand areas have uniform aspect ratios. In our system, Gentle Adaboost is used and multi-feature is selected to describe human hands. The experiments show a good result for interacting with smart devices.

3. The structure of the detection system

Boosting can learn a strong classifier based on a set of weak classifiers by re-weighting the training samples. Inspired by boosting classifier, Freund and Schapire proposed the Adaboost algorithm [27,28] and its more effective versions, Real AdaBoost and Gentle Adaboost. It focuses on the training samples that were incorrectly classified in the last iteration. The main differences among these Adaboost algorithms are the procedures of re-weighting training samples and their respective weak classifiers after each training iteration [29]. The regression stump (weak classifier of Gentle Adaboost) is defined as

$$g(x)=\begin{cases} a, & f(x)<\varphi \\ b & \text{otherwise} \end{cases} \tag{2.1}$$

where $f(x)$ is the feature value of sample x, and φ is its learned corresponding threshold. a and b are determined by a weighted conditional expectation on each side of the threshold

$$a=\frac{\sum_i^N w_i y_{i[x<\varphi]}}{\sum_i^N w_{i[x<\varphi]}}$$
$$b=\frac{\sum_i^N w_i y_{i[x>\varphi]}}{\sum_i^N w_{i[x>\varphi]}} \tag{2.2}$$

The output of $g(x)$ is continuous and bounded to $[-1,1]$. The classification error is calculated by weighted least-squares. Therefore, the weight of training samples will not increase as dramatically as in Discrete Adaboost, which make the algorithm more gentle and robust to noisy data. In [30], Lienhart achieved an object detection system based on the three Adaboost algorithms and compared their properties. It indicated that the Gentle Adaboost outperforms the others in both accuracy and costs. The algorithm is presented in Algorithm 1.

Algorithm 1. The framework of training a gentle adaboost classifier.

1: Get positive and negative samples from the dataset.
2: Calculate the features $f(x_i)$ of each sample image and form the training set $(f(x_1),y_1),\ldots,(f(x_N),y_N)$, $y_i\in -1,1$.
3: Initialize weights $w_i=1/N$, given The target detection rate d and false rate f.
4: Learn a weak classifier which fits the regression function $g_t(x)$ by minimizing Mean Square Error(MSE): $\epsilon_t=\sum_{i=1}^m w_i\cdot\left(y_i-g_t(x_i)\right)^2$.
5: Update $G(x)\leftarrow G(x)+g_t(x)$
6: Update $w_i\leftarrow w_i\exp[-y_i g_t(x)]$, and renormalize so that $\sum_i w_i=1$
7: Test the current classifier $G(x)$ and calculate the detection rate d_t and false rate f_t
8: **if** d_t, f_t can not reach d, f **then**
9: Return to 4.
10: **end if**
11: Output the classifier $\text{sign}[F(x)]=\text{sign}[\sum_t g_t]$

Cascade which can be treated as a degenerated decision tree is often used to improve the efficiency of a detection system. The structure of a cascade classifier is illustrated in Fig. 1. Only the samples that are considered as object by the early stages can pass through and be evaluated by the following stages. Since most

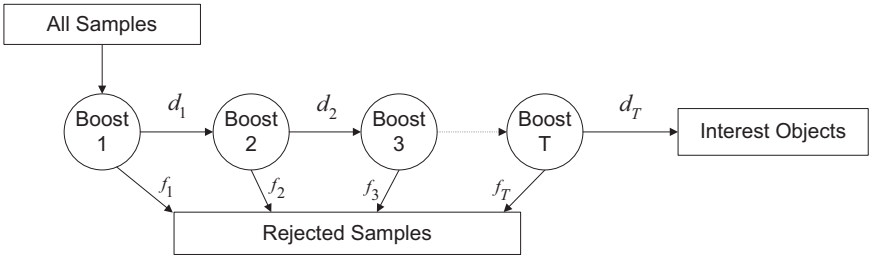

Fig. 1. A cascade classifier with detection rate $D=\prod_{t=1}^T d_i$ and false rate $F=\prod_{t=1}^T f_i$.