Neurocomputing 149 (2015) 79–85

Contents lists available at ScienceDirect

Neurocomputing

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

RGB-D action recognition using linear coding

Huaping Liu [a,b,c,*], Mingyi Yuan [a,b,c], Fuchun Sun [a,b,c]

[a] Department of Computer Science and Technology, Tsinghua University, Beijing, China
[b] State Key Laboratory of Intelligent Technology and Systems, Beijing, China
[c] Tsinghua National Laboratory of Information Science and Technology, Beijing, China

ARTICLE INFO

Article history:
Received 1 July 2013
Received in revised form
11 November 2013
Accepted 18 December 2013
Available online 1 August 2014

Keywords:
RGB-D
Action recognition
Linear coding

ABSTRACT

In this paper, we investigate action recognition using an inexpensive RGB-D sensor (Microsoft Kinect). First, a depth spatial-temporal descriptor is developed to extract the interested local regions in depth image. Such descriptors are very robust to the illumination and background clutter. Then the intensity spatial-temporal descriptor and the depth spatial-temporal descriptor are combined and feeded into a linear coding framework to get an effective feature vector, which can be used for action classification. Finally, extensive experiments are conducted on a publicly available RGB-D action recognition dataset and the proposed method shows promising results.

1. Introduction

Recognition of human actions has been an active research topic in computer vision. In the past decade, research has mainly focused on learning and recognizing actions from video sequences captured from a single camera and rich literature can be found in a wide range of fields including computer vision, pattern recognition, machine leaning and signal processing. Recently, there are some approaches using local spatio-temporal descriptors together with bag-of-words model to represent the action. Since these approaches do not rely on any preprocessing techniques, e.g. foreground detection or body-part tracking, they are relatively robust to the change of viewpoint, noise, background, and illumination. However, most existing work on action recognition is based on color video, which leads to relatively low accuracy even when there is no clutter.

Different from these work, our motivation is driven by the application of the famous mass-production consumer electronics device Kinect, which provides a depth stream and a color stream. Kinect has been applied in extensive fields including people detection and tracking [1,2]. Currently there exist very few work that utilize the color-depth sensor combination for human action recognition. For example, Ref. [3] used the depth information but totally ignored the depth information. In fact, as we will analyze, the color information and depth information can be complementary since the human actions are in essence three-dimensional. However, how to effectively fuse the color and depth information remains a great challenging problem. In this paper, we extract the local descriptors from the color and depth video and utilize the linear coding framework to integrate the color and depth information. The main contributions are summarized as follows:

1. The conventional STIP descriptor is extended by incorporating depth information to deal with depth video. Such descriptors are very robust to the illumination and background clutter.
2. A linear coding framework is developed to fuse the intensity spatial-temporal descriptor and the depth spatial-temporal descriptor to form robust feature vector. In addition, we further exploit the temporal intrinsics of the video sequence and design a new pooling technology to improve the description performance.
3. Extensive experiments are conducted on a publicly available RGB-D action recognition dataset and the proposed method shows promising results.

The organization of this paper is as follows: in Section 2 we introduce the feature extraction. Sections 3 and 4 present the coding and pooling methods, respectively. The experimental results are given in Section 5. Finally, Section 6 gives some conclusions.

2. Feature extraction

There are several schemes applied to time-consistent scene recognition problems. Some of them are statistics based approaches, such as Hidden Markov Models, Latent-Dynamic Discriminative Model [4], and so on. Differently, Space-Time Interest Points (STIPs) [5] regard the temporal axis as the same

* Corresponding author.
E-mail address: hpliu@tsinghua.edu.cn (H. Liu).

http://dx.doi.org/10.1016/j.neucom.2013.12.061
0925-2312/

as the spatial axes and looks for the features along the temporal axis as well. We prefer to the latter ones because the time parameter of the sample is essentially the same as the space parameters in the mathematics sense. Since we have plenty of reliable mathematics tools and feature construction schemes, the extensions of already existed feature schemes can be safely applied in such time-relevant problems. Meanwhile, those schemes can be naturally extended for more complex tasks.

STIPs is an extension of SIFT (Scale-Invariant-Feature-Transform) in 3-dimensional space and uses one of Harris3D, Cuboid or Hessian as the detector. For certain video, dense sampling is performed at regular positions and scales in space and time to get 3D patches. We perform sampling from 5 dimensions x, y, t, σ and τ where σ and τ are the spatial and temporal scales, respectively. Usually, the minimum size of a 3D patch is 18×18 pixel by 10 frames. Spatial and temporal sampling are done with 50% overlap. Multi-scale patches are obtained by multiplying σ and τ by a factor of $\sqrt{2}$ for consecutive scales. In total, there are 8 spatial and 2 temporal scales since the spatial scale is more important than the time scale. Different spatial and temporal scales are combined so that each video is sampled 16 times with different σ and τ parameters. The detector is applied in each video and locates interest points as well as the corresponding scale parameters. After that, the HOG-HOF (Histogram-Of-Gradient-Histogram-Of-optical-Flow) descriptors are calculated at those detected interest points and the sample features are generated.

In our work, the feature descriptors are extracted from both RGB image and depth image. For applying the STIPs detector and descriptor on the depth information, we scale the depth value from 16-bit unsigned integer to 8-bit unsigned integer by searching the maximum and minimum (above 0) of the depth value in the whole sample video, and transforming each depth pixel linearly as

$$d_{new} = \begin{cases} 0, & d = 0 \\ 255 \times \dfrac{d - d_{\min}}{d_{\max} - d_{\min}}, & d > 0, \end{cases} \tag{1}$$

where $d_{\max}$ is the maximal depth of the video sample and $d_{\min}$ is the minimal depth above 0 of the video sample. We save the matrices of d_{new} as the gray type depth video and use it in the same way as the RGB one. For a typical video, there are about several hundred frames of RGB and depth image pairs and thousands of STIPs descriptors detected. The STIPs descriptors are of 162 dimensional vector composed of 90-dimensional HOG [6] descriptor and 72-dimensional HOF [7] descriptor. The HOG and HOF descriptors are computed at the detected interest point with the associated scale factors. The STIPs descriptor describes the local variation characters well in the xy space as well as in the t space. Fig. 1 shows the features detected in one frame of both RGB image and depth image. The circles center at the interest points and the radius of the circle is proportional to the scale factor σ of that interest point. It can be seen that the STIPs features on RGB image and depth image cover different regions of the subjects because of the different pixel variations in the two types of data. In fact, the brightness of each pixel in the depth image has larger variation near the contour of the subject, including the head, arms and legs, etc. On the other hand, the variation of the brightness of the RGB image appears at the boundary of the texture of the subject. So the STIPs features in the RGB images disclose more detail characters of the subjects themselves while in the depth images they extract more characters of the shape of the subjects. In conclusion, both features are useful and equally important for classification.

3. Coding approaches

A popular method for coding is the vector quantization (VQ) method, which solves the following constrained least square fitting problem:

$$\min_{\mathbf{C}} \sum_{i=1}^{M} \|\mathbf{x}_i - \mathbf{B}\mathbf{c}_i\|_2^2 \quad \text{s.t. } \|\mathbf{c}_i\|_0 = 1, \|\mathbf{c}_i\|_1 = 1, \quad \mathbf{c}_i \geqslant 0, \ \forall i, \tag{2}$$

where $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_M]$ is the set of codes for $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M]$. The cardinality constraint $\|\mathbf{c}_i\|_0 = 1$ means that there will be only one non-zero element in each code $\mathbf{c}_i$, corresponding to the quantization id of $\mathbf{x}_i$. The non-negative, ℓ_1 constraint $\|\mathbf{c}_i\|_1 = 1$, $\mathbf{c}_i \geqslant 0$ means that the coding weight for $\mathbf{x}_i$ is 1. In practice, the single non-zero element can be found by searching the nearest neighbor.

VQ provides an effective way to treat an image as a collection of local descriptors, quantizes those descriptors into discrete "visual words", and then computes a compact histogram representation of the image for the classification tasks. For the action recognition task, one can use all the STIPs descriptors extracted from the video sample as the candidate vectors for building the codebook. In our implementation, the standard k-means [8] method is employed to cluster the feature descriptors and the cluster centers are selected as the codewords to form the codebook. The codebook size K could be small to save the computation time (i.e, $K < = 512$). For each feature descriptor, the nearest codeword in the Euclidean distance measurement is selected as the coding vector, in the form of a K-dimensional vector with all zero components but one set to 1. After a sum pooling (or sum normalization) [9] process, the local feature vectors are combined into a global histogram representation of the codewords. This is the framework of the well known Bag-of-Words model [10]. Finally, a χ^2-kernel [11] SVM is used to do the classification. The χ^2 kernel is denoted as

$$K(H_i, H_j) = e^{-(1/A)D(H_i, H_j)}, \tag{3}$$

where $H_i = \{h_{in}\}$ and $H_j = \{h_{jn}\}$ are two histogram vectors, $D(H_i, H_j) = \frac{1}{2}\sum_{n=1}^{K}(h_{in} - h_{jn})^2/(h_{in} + h_{jn})$ is the χ^2 distance of H_i and H_j, and A is the mean value of the distance between every two pooling vectors of all the samples.

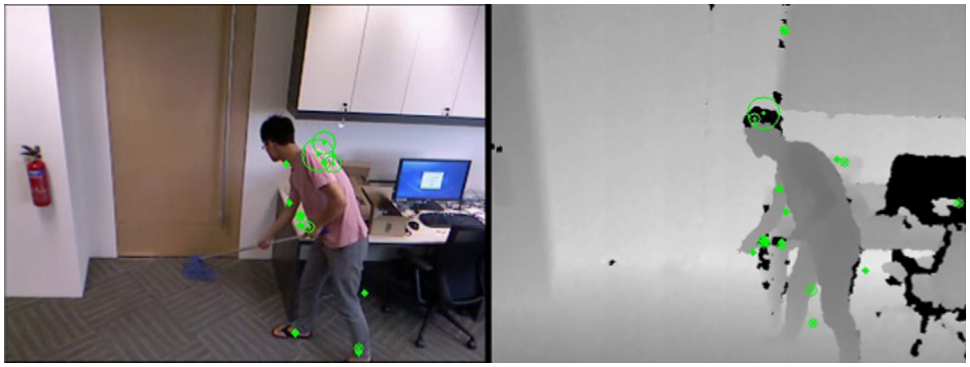

Fig. 1. STIPs features on RGB image (left) and depth image (right).