Sequential Max-Margin Event Detectors

Dong Huang, Shitong Yao*, Yi Wang*, and Fernando De La Torre

Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 15213. USA

Abstract. Many applications in computer vision (e.g., games, human computer interaction) require a reliable and early detector of visual events. Existing event detection methods rely on one-versus-all or multi-class classifiers that do not scale well to online detection of large number of events. This paper proposes Sequential Max-Margin Event Detectors (SMMED) to efficiently detect an event in the presence of a large number of event classes. SMMED sequentially discards classes until only one class is identified as the detected class. This approach has two main benefits w.r.t. standard approaches: (1) It provides an efficient solution for early detection of events in the presence of large number of classes, and (2) it is computationally efficient because only a subset of likely classes are evaluated. The benefits of SMMED in comparison with existing approaches is illustrated in three databases using different modalities: MSRDaliy Activity (3D depth videos), UCF101 (RGB videos) and the CMU-Multi-Modal Action Detection (MAD) database (depth, RGB and skeleton). The CMU-MAD was recorded to target the problem of event detection (not classification), and the data and labels are available at `http://humansensing.cs.cmu.edu/mad/`.

Keywords: Event Detection, Activity Recognition, Time Series Analysis, Multi-Modal Action Detection.

1 Introduction

Event detection in time series is a topic of growing interest in computer vision and machine learning. Many problems in surveillance [14], activity analysis [1], clinical monitoring [20] and human computer interaction [12] can be posed as detecting events in time series. While the type of data may vary (e.g., video, accelerometers, EEG, depth), the same techniques for event detection can be applied by changing the feature representation for each data type. At this point, it is important to notice that the vast majority of work on activity recognition in video does not address the detection problem, but rather focuses on classification (i.e., the start and the end of the action are given). This is surprising, since detection is of more practical use. While the methods developed for classification could be applied to detection by searching over different temporal scales and adding a null class (none of the existing classes), it is unclear how these methods will perform in practice. This paper focuses on developing a fast and efficient method to detect events from a large number of event classes, and shows its benefits on detecting human actions from a variety of sensing modalities (video, depth, skeleton).

* Both authors contributed equally to this paper.

D. Fleet et al. (Eds.): ECCV 2014, Part III, LNCS 8691, pp. 410–424, 2014.

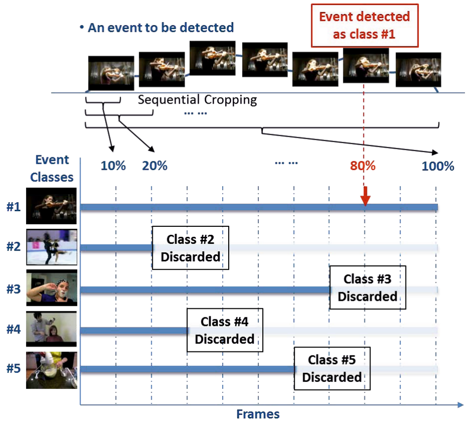

Fig. 1. Given a test event (sequence of a subject playing the violin in the top of the figure), SMMED sequentially evaluates partial events at $\{10\%, 20\%, \cdots, 100\%\}$. When SMMED is confident that the event is not from a given class, it automatically discards this class from further consideration. The blue bars illustrate that class #2(IceDancing), #4(BlowDryHair), #5(Blending), #3(Shaving) are sequentially discarded. Finally, the test event is identified as class #1(playing the violin): the remaining class (the longest blue bar), after 80% of the event has been evaluated.

Standard methods for event detection rely on one-vs-all classifiers [8, 11, 16]. Typically, for each event class, a one-vs-all detector is trained using temporal segments selected from training events. Applying one-vs-all detectors in the presence of a large number of event classes has three major drawbacks: (1) detection scores are not directly comparable, because they are not normalized, (2) detection is often slow because many detectors have to be run simultaneously, and (3) multiple detectors may fire at a given time because classes are not mutually exclusive. Using a multi-class event detector [3] would guarantee that the class label for a particular segment is unique. However, it still remains challenging to produce a consistent class label for non-overlapping but consecutive segments; typically the classifier with a higher score is selected. As a result, different consecutive segments of a given event might have different labels. This problem could be solved using off-line strategies (e.g., k-segmentation [16]), but it is unclear how to solve it online.

To address the above challenges, we propose Sequential Max-Margin Event Detectors (SMMED), which can efficiently scale to a large number of classes. Similar to [8, 21], SMMED is a maximum margin classifier learned using partial segments of training events. Unlike existing approaches, SMMED can sequentially select the most likely subset of classes while automatically enforcing a larger margin for the unlikely classes. As a result, SMMED can reliably discard many classes using only partially observed events.

Fig. 1 illustrates the basic idea of SMMED for multi-class event detection. In this case, we have five event classes (class #1-#5). The top part of Fig. 1 is the event to be detected. As it is common in online event detection, the frames of the event are provided sequentially. A five-class SMMED is sequentially run on partial segments of the event.

As SMMED observes more of the event, it is able to confidently discard classes such as #2 (Ice Dancing), #4 (Blow Dry Hair), #5 (Blending) and #3 (Shaving). Having ruled out all other classes, SMMED detected class #1 (playing the violin) after only seeing 80% of the event. Note how SMMED reduces the set of possible classes over time, making it a good candidate method for efficient detection when a large number of event classes exist.

We illustrate the benefits of our approach on the MSRDaily Activity (3D depth videos) [22] and UCF101 (RGB videos) [18] databases, where SMMED achieves slightly better performance than the multi-class SVM-based detectors, and does so in a more efficient manner. In addition, to evaluate multi-class event detection on continuous sequences, we have collected and labeled the CMU-Multi-modal Action Detection (CMU-MAD) database. This database contains RGB videos, 3D depth videos and body-joint sequences of 20 subjects performing 35 different actions. All data were recorded using a Microsoft Kinect sensor. We believe that one of the reasons for classification being more popular than detection is the limited number of databases with adequate labels. To encourage researchers to work on activity detection, we have released the CMU-MAD database (RGB, 3D depth and body-joint sequences) with frame-wise event labels and example codes to access the database.

2 Related Work

Activity recognition from video has been a long-standing problem in computer vision, and the vast majority of the literature deals with the problem of activity classification in video. Niebles et al. [13] used probabilistic Latent Semantic Analysis for unsupervised learning of human actions. Brand and Kettnaker [3] trained an HMM with entropy minimization and interpreted the hidden states for detection and segmentation of activities in video. [9, 19] modeled the temporal dynamics of activities for classification. Bengio et al. [2] applied tree-traversing to reduce the computational cost of multi-class activity classification in video.

The problem of detection, however, has been relatively unexplored. Early work of Sminchisescu et al. [17] used conditional models for human action detection. Ke et al. [10] detected human actions in videos of crowds. Oh et al. [15] proposed a parametric segmental switching linear dynamical system to model honey-bee behavior. This approach segments the video sequence off-line in a supervised manner. Recently, Gall et al. [6] used Hough Forests to segment the spatial-temporal cuboids of person from videos. Most related to our work is the work of Hoai et al. [7, 8]. [7] temporally segmented and detected events in video combining segment-based SVMs with Dynamic Programming (DP). [8] extended [7] to address the problem of early detection of events using a binary classifier trained to detect events as soon as possible. Unlike the aforementioned approaches, SMMED is a multi-class early event detection approach. SMMED sequentially discards the unlikely classes on consecutive segments, until a reliable class label can be identified from the remaining classes individually. Therefore, the number of detector scores that needs to be computed is reduced over time, making SMMED an efficient solution for a large number of event classes.

3 Structure Output SVM Event Detectors

Structured Output SVM (SO-SVM) [7, 8, 21] provides a natural formalism for event detection in time series, because the output of SO-SVM is the start and the end of the detected segment as well as the class label. More importantly, SVM is a discriminative model that is able to more efficiently model the null class [7] than generative models (e.g., HMMs), which is crucial in detection problems. This section reviews existing SO-SVM algorithms and reformulates SO-SVM as an unconstrained optimization problem.

Let c be the number of event classes, $\mathbf{x}_i^{y_i} \in \Re^d$ be a vector descriptor for the features of the i^{th} segment that starts at the frame s_i and ends at the frame e_i, i.e., the temporal interval is $y_i = [s_i, e_i]$. The goal of event detection is to determine the temporal interval y_i and the class label of an event in a testing video. Let us denote $\delta_{\tilde{y}_i} \in [0, 1]$ to be a scalar that measures the overlap between a temporal segment $\tilde{y}_i = [\tilde{s}_i, \tilde{e}_i]$ and the ground truth segment $y_i = [s_i, e_i]$ of an event. That is, $\delta_{\tilde{y}_i} = 0$ if there is no overlap with the ground-truth, and $\delta_{\tilde{y}_i} = 1$ if there is a perfect overlap (i.e. $\tilde{y}_i = y_i$). The cost function for training Multi-class SO-SVM (MSO-SVM) event detectors can be written as:

$$\begin{aligned} \min_{\mathbf{W}} \quad & \|\mathbf{W}\|_F^2 + \lambda \sum_{p=1}^{c} \sum_{i \in \mathcal{N}_p} \xi_i^2 \qquad (1) \\ s.t. \quad & \delta_{\tilde{y}_i} - (\mathbf{w}_p - \mathbf{w}_q)^T \mathbf{x}_i^{\tilde{y}_i} \leq \xi_i; \quad \xi_i > 0, \\ & \forall \tilde{y}_i; \;\; i \in \mathcal{N}_p \;\; ; \forall p \neq q; \;\; p, q = 1, \cdots, c, \end{aligned}$$

where the column vectors of $\mathbf{W} = [\mathbf{w}_1, \cdots, \mathbf{w}_c] \in \Re^{d \times c}$ are the classifier vectors for the c event classes, and $\mathcal{N}_p$ is the index set of event instances belonging to the p^{th} class. The constrains in Eq. 1 state that, for a segment feature $\mathbf{x}_i^{\tilde{y}_i}$ from the p^{th} class ($i \in \mathcal{N}_p$), the classification score with the p^{th} class $\mathbf{w}_p^T \mathbf{x}_i^{\tilde{y}_i}$ should be larger than the scores with any other class $\mathbf{w}_q^T \mathbf{x}_i^{\tilde{y}_i}$ ($p \neq q; p, q = 1, \cdots, c$) by a margin $\delta_{\tilde{y}_i}$. ξ_i is the slack variable that compensates for misclassification errors, and ξ_i^2 denotes the quadratic loss.

Although MSO-SVM is traditionally trained in the dual, we argue that formulating and optimizing the problem in the primal facilitates understanding and generalization of the method. Following Chapelle et al. [4] and using Lagrange multipliers, we can re-write Eq. 1 as an unconstrained optimization problem

$$\min_{\mathbf{W}} \|\mathbf{W}\|_F^2 + \lambda \sum_{p=1}^{c} \sum_{\{i \in \mathcal{N}_p; \tilde{y}\}} \|\mathbf{r}^{\tilde{y}_i} - (\mathbf{w}_p \mathbf{1}^T - \mathbf{W})^T \mathbf{x}_i^{\tilde{y}_i}\|_h, \qquad (2)$$

where the second term in Eq. 2 measures the mis-classification error with the quadratic loss: $\|\mathbf{x}\|_h = \|max(0, \mathbf{x})\|_2^2 = \sum_i max(0, x_i)^2$. $\mathbf{r}^{\tilde{y}_i} \in \Re^{c \times 1}$ is a vector of which all the elements are $\delta^{\tilde{y}_i}$ expect for the p^{th} element, which is zero because this element corresponds to $(\mathbf{w}_p - \mathbf{w}_p)^T \mathbf{x}_i^{\tilde{y}_i}$). The vector $\mathbf{r}^{\tilde{y}_i} - (\mathbf{w}_p \mathbf{1}^T - \mathbf{W})^T \mathbf{x}_i^{\tilde{y}_i}$ contains the scores for each of the c classes. Suppose that $\mathbf{x}_i^{\tilde{y}_i}$ belongs to the p^{th} class ($i \in \mathcal{N}_p$). If the q^{th} element of the scores is larger than 0, then $\mathbf{x}_i^{\tilde{y}_i}$ is considered a mis-classified sample between the p^{th} class and the q^{th} class. In addition, $\delta^{\tilde{y}_i}$ is larger for the higher overlapped segments, which enforces higher penalty to the mis-classification of segments with higher overlap.

During testing, two popular approaches can be used for inference using MSO-SVM (i.e., detecting the segments where the event occurs): (1) The off-line approach, e.g., k-segmentation [16], which automatically selects the k non-overlapping segments that maximize the response of the classifier. This approach was proven to be optimal and has no local minima. However, it can only be implemented off-line and the number of segments k must be given a priori. (2) The Dynamic Programming (DP) approach [7] which can be adapted for online detection. Given a sliding window (maximum length of the training events) along the test time series, online DP solves for the optimal segment configuration such that the sum of the MSO-SVM classification scores is maximal. It then updates the class labels of segments in the sliding window and moves the sliding window forward until the end of the time series. However, consecutive segments around the true event may have inconsistent labels, and it can be computationally intensive to evaluate all classifiers in $\mathbf{W}$. To address these issues next section proposes SMMED.

4 Sequential Max-Margin Event Detectors

This section describes SMMED to overcome the drawbacks of standard MSO-SVM for event detection.

4.1 Cost Function for SMMED

Given the i^{th} training event of the p^{th} class ($i \in \mathcal{N}_p$) with temporal segment $y = [s, e]$, we split the segment y into m sub-segments of equal length l, i.e., $l = (e - s)/m$. We use these sub-segments to construct a set of partially overlapped temporal segments $\{[s, s + l]$, $[s, s + 2l]$, ... ,$[s, e]\}$. In the left column of Fig. 2, we illustrate the partial temporal segmentation in four event classes.

The feature vector $\mathbf{x}_i^{y_f}$ is computed from the i^{th} event at segment y_f. Note that the temporal segment $y_{f_2} = [s, s + 2l]$ overlaps with the previous segment $y_{f_1} = [s, s + l]$. $\mathbf{x}_i^{y_{f_2}}$ thus contains more information than $\mathbf{x}_i^{y_{f_1}}$ for discriminating the true class (#1) from the other classes. Therefore, if a class can already be discriminated from the true class using segment y_f, it is not necessary to consider this class for the larger segment y_{f_1}. To learn this property, SMMED uses larger margins to penalize mis-classification with this class. For instance, in Fig. 2, at the segment $y_{f_1} = [s, s + l]$ the 4^{th} class can already be discriminated from the 1^{st} class. Thus at the next larger segment, $y_{f_2} = [s, s + 2l]$, of the event $\mathbf{x}_i^{y_{f_2}}$, the 4^{th} element in the margin vector $\mathbf{r}_1^{y_{f_2}}$ is increased by a positive scalar $\delta_{14}^{[s,f_2]}$. Similarly, at $y_{f_3} = [s, s + 3l]$, the term for the 3^{rd} class is increased by $\delta_{13}^{[s,f_3]}$. The classifier of the 1^{st} class $\mathbf{w}_1$ is learned by minimizing the sum of the three error terms above.

Including all partial event segments from the c classes in Eq. (2), the cost function of SMMED is

$$L(\mathbf{W}, \mathbf{r}_{(p)}^{y_f}) = \|\mathbf{W}\|_F^2 + \lambda \sum_{p=1}^{c} \sum_{f=s+l}^{e} \left\| \mathbf{r}_{(p)}^{y_f} \mathbf{1}^T - (\mathbf{W}\mathbf{S}_p)^T \mathbf{X}_{(p)}^{y_f} \right\|_h \tag{3}$$

where $\mathbf{X}_{(p)} = [\mathbf{x}_i]_{i \in \mathcal{N}_p}$ is the matrix containing the vector descriptors for all partial events belonging to the p^{th} class. $\mathbf{S}_p \in \Re^{c \times c}$ is a selection matrix for constructing the

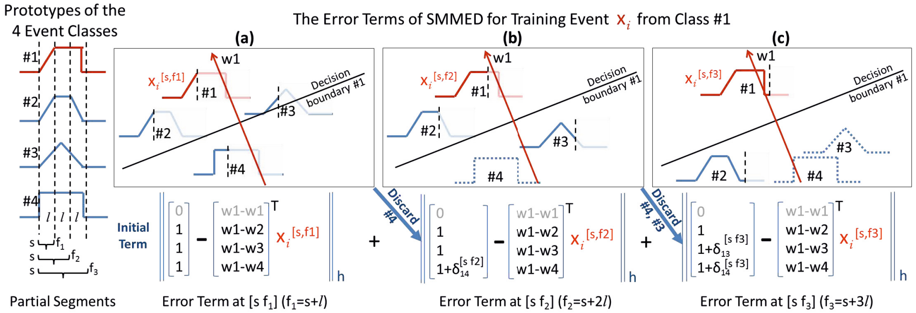

Fig. 2. A synthetic example that illustrates training SMMED. The sub-figures in the left column represent prototypes of four synthetic event classes. SMMED builds three partial segments for each class in the range $[s, f]$ ($f = s + l, s + 2l, s + 3l$). Let $\mathbf{x}_i$ be a training event instance from class $\#1$. The sub-figures (a)-(c) illustrate training SMMED using the temporal segments of $\mathbf{x}_i$. The vectors $\mathbf{w}_1$, $\mathbf{w}_2$, $\mathbf{w}_3$ and $\mathbf{w}_4$ are the classifiers of the 4 classes. Note that in **(a)**, at segment $f_1 = s + l$, all the partial segments from classes $\#2$ and $\#3$ have a ramp, so they cannot be discriminated from class $\#1$. The partial segment of $\#4$ has a step, and thus can be discriminated from $\mathbf{x}_i^{[s,f_1]}$. SMMED therefore enforces that for all subsequent (larger) partial segments $f_2 = s + 2l$ (in **(b)**) and $f_3 = s + 3l$ (in **(c)**) class $\#4$ remain discriminated from class $\#1$. In order to enforce this property the error term associated with $\mathbf{w}_4$ is penalized by increasing its corresponding margin by $\delta_{14}^{[s,f_2]}$ and $\delta_{14}^{[s,f_3]}$ respectively. Similarly, in **(b)**, the training partial segment of class $\#3$ has a peak and can be discriminated from $\#1$. Therefore, in **(c)**, for partial segment $[s, s + 3l]$, the term for $\mathbf{w}_3$ is also penalized. Finally, the total error term for $\mathbf{x}_i$ is the sum of the three terms above.

difference between the $\mathbf{w}_p$ and all other classifiers, such that the product of $\mathbf{W} \in \Re^{d\times c}$ (the matrix of the c classifiers) and $\mathbf{S}_p$ is $\mathbf{W}\mathbf{S}_p = [\mathbf{w}_p - \mathbf{w}_1, \mathbf{w}_p - \mathbf{w}_2, \cdots, \mathbf{w}_p - \mathbf{w}_c]$. The vector $\mathbf{r}_{(p)}^{y_f} \in \Re^{c\times 1}$ contains the margins for the partial events $y_f = [s, f]$ of the p^{th} class. As illustrated in Fig. 2, the key to SMMED Eq. (3) is to update the elements of the vector $\mathbf{r}_{(p)}^{y_f}$ to penalize those classes that can already be discriminated from the true class. For instance, if at y_f, the q^{th} class can be discriminated from p^{th} class, the q^{th} element of vector $\mathbf{r}_{(p)}^{y_{f+l}}$ is updated to be $1 + \delta_{pq}^{y_{f+l}}$, where $\delta_{pq}^{y_{f+l}}$ is a scalar to be computed below.

SMMED uses a quadratic loss for $\|\cdot\|_h$. In this case, the optimal value of $\delta_{pq}^{y_{f+l}}$ can be estimated using a simple and efficient method. Note that the L_2-based norm of a matrix is minimal when the value of its elements are uniformly distributed. Assuming the classification error in the second term of Eq. 3 is class-wise uniform at y_{f+l}, the q^{th} class is to be discarded from the p^{th} class, Γ_p is the index set of remaining classes. It follows that

$$\left\|(1 + \delta_{pq}^{y_{f+l}})\mathbf{1}^T - (\mathbf{w}_p - \mathbf{w}_q)^T\mathbf{X}_{(p)}^{y_{f+l}}\right\|_h = \frac{1}{|\Gamma_p|}\sum_{j\in\Gamma_p}\left\|\mathbf{1}^T - (\mathbf{w}_p - \mathbf{w}_j)^T\mathbf{X}_{(p)}^{y_{f+l}}\right\|_h .$$

The solution to $\delta_{pq}^{y f+1}$ is thus

$$\delta_{pq}^{y f+l} = \left\| \left(\frac{1}{|\Gamma_p|} \sum_{j \in \Gamma_p} \mathbf{w}_j - \mathbf{w}_q \right)^T \mathbf{X}_{(p)}^{y f+l} \right\|_2 . \tag{4}$$

Eq. (3) only considers the c labeled classes. However, in real applications, there are temporal segments not belonging to any class: the null event class. SMMED (Eq. 3) can be trained with a $(c+1)^{th}$ class. The temporal segments of the $(c+1)^{th}$ classifier are randomly selected from the unlabeled frames in the training sequences.

4.2 Solving SMMED

We solved SMMED (Eq. 3) in the primal following [4]. However, unlike [4] we used an efficient line search algorithm adapted to SMMED.

We initialized $\mathbf{W} = \mathbf{1}\mathbf{1}^T \in \Re^{d \times c}$ and $\mathbf{r}_{(p)}^{y_f} = \mathbf{1} \in \Re^{c \times 1}$ for all classes ($p = 1, \cdots, c$), and then iteratively updated the set of support vectors selected by $\| \cdot \|_h$, the classifiers $\mathbf{W}$, and the margin vectors ($\mathbf{r}_{(p)}^{y_f}$s) until convergence. The following steps were iterated:

(1) Identify the Mis-classified Sample-class Pairs. Let $\mathbf{H} = \mathbf{r}_{(p)}^{y_f}\mathbf{1}^T - (\mathbf{W}\mathbf{S}_p)^T \mathbf{X}_{(p)}^{y_f} \in \Re^{c \times |\mathcal{N}_p|}$ be the matrix evaluated with the quadratic loss $\| \cdot \|_h$ for the p^{th} class at the f^{th} segment in Eq. 3. The quadratic loss $\| \cdot \|_h$ splits the elements in $\mathbf{H}$ into two sets: elements that fall into the zero part of the quadratic loss function (Ω_0) and those that fall into the quadratic part (Ω_2). Recall that the non-zero values in the SVM quadratic loss measure the classification error. The set Ω_2 is defined as the mis-classified class-sample pairs, i.e., $\{(q, \mathbf{x}_j)\}$, ($q = 1, 2, \cdots, c$; $j = 1, 2, \cdots, n$). To simplify the derivation of the algorithm, in the following, we use the subindex "$\cdot_{\Omega_2}$" to denote both the classifier index p and the data index i that form a class-sample pair in the set Ω_2.

(2) Update the Classifiers W. Update $\mathbf{W}$ with gradient descent $\mathbf{W} = \mathbf{W} - \beta \frac{\partial L}{\partial \mathbf{W}}$, where

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{W}} = 2\mathbf{W} + \lambda \sum_{p=1}^{c} \sum_{f=s+l}^{e} \Big[& -2[\mathbf{X}_{(p)}^{y_f}]_{\Omega_2} \mathbf{1} ([\mathbf{r}_{(p)}^{y_f}]_{\Omega_2})^T [\mathbf{S}_p]_{\Omega_2}^T \\ & + 2[\mathbf{X}_{(p)}]_{\Omega_2}^{y_f} ([\mathbf{X}_{(p)}^{y_f}]_{\Omega_2})^T \mathbf{W} [\mathbf{S}_p]_{\Omega_2} [\mathbf{S}_p^T]_{\Omega_2} \Big] . \end{aligned} \tag{5}$$

Substituting $\mathbf{W} = \mathbf{W} - \beta \frac{\partial L}{\partial \mathbf{W}}$ into L, and computing the partial derivative of L w.r.t. β, we have:

$$\begin{aligned} \frac{\partial L}{\partial \beta} = tr & \left(-\mathbf{W}^T \frac{\partial L}{\partial \mathbf{W}} - \frac{\partial L}{\partial \mathbf{W}}^T \mathbf{W} + 2\beta \frac{\partial L}{\partial \mathbf{W}}^T \frac{\partial L}{\partial \mathbf{W}} \right) \\ & + \lambda \sum_{p=1}^{c} \sum_{f=s+l}^{e} tr[\mathbf{B}\mathbf{A}^T + \mathbf{A}\mathbf{B}^T + 2\beta \mathbf{A}\mathbf{A}^T], \end{aligned} \tag{6}$$

where the matrix $\mathbf{A} = [\mathbf{S}_p]_{\Omega_2}^T \frac{\partial L}{\partial \mathbf{W}} [\mathbf{X}_{(p)}^{y_f}]_{\Omega_2}$, and $\mathbf{B} = [\mathbf{r}_{(p)}]_{\Omega_2}^{y_f} \mathbf{1}^T - [\mathbf{S}_p]_{\Omega_2}^T \mathbf{W}^T [\mathbf{X}_{(p)}^{y_f}]_{\Omega_2}$. Then setting $\frac{\partial L}{\partial \beta^*} = 0$, the optimal step size is computed as

$$\beta^* = \frac{tr\left(\mathbf{W}^T \frac{\partial L}{\partial \mathbf{W}} + \frac{\partial L}{\partial \mathbf{W}}^T \mathbf{W} + \lambda \sum_{p=1}^{c} \sum_{f=s+l}^{e} \mathbf{B}\mathbf{A}^T + \mathbf{A}\mathbf{B}^T\right)}{tr\left(2 \frac{\partial L}{\partial \mathbf{W}}^T \frac{\partial L}{\partial \mathbf{W}} - \lambda \sum_{p=1}^{c} \sum_{f=s+l}^{e} 2\mathbf{A}\mathbf{A}^T\right)}. \quad (7)$$

(3) Update the Margin Vector $\mathbf{r}_{(p)}^{y_{f+1}}$ at Segment y_{f+1} (See Fig. 2). Let $\mathbf{H} = \mathbf{r}_{(p)}^{y_{f-1}} \mathbf{1}^T - (\mathbf{W}\mathbf{S}_p)^T \mathbf{X}_{(p)}^{y_{f-1}} \in \Re^{c \times |\mathcal{N}_p|}$ be the matrix for the p^{th} class at the segment $[s, f-l]$ in $\| \cdot \|_h$. Updating $\mathbf{r}_{(p)}^{y_f}$ consists of two sub-steps:

(a) Identify classes at $y_f = [s, f]$ that can already be discriminated from the true class. The q^{th} class is discriminated from the p^{th} class at y_f, if $\min(\mathbf{H}(q,:)) + \alpha(\max(\mathbf{H}(q,:)) - \min(\mathbf{H}(q,:))) < 0$, where α is a positive scalar, $\alpha \in [0, 1]$. For instance, $\alpha = 0.9$ means 90% of samples in $\mathbf{X}_{(p)}^{y_f}$ are not classified as the p^{th} class.

(b) Update the elements of $\mathbf{r}_{(p)}^{y_{f+1}}$ at segment $y_{f+1} = [s, f+1]$ to be $1 + \delta_{pq}^{y_{f+1}}$ where $\delta_{pq}^{y_{f+1}}$ is computed using Eq. 4.

Step **(1)**-**(3)** are repeated until the changes in $\mathbf{W}$ are small (i.e., $< 10^{-5}$).

4.3 Detecting Events in a Test Sequence

After the matrix $\mathbf{W}$ is learned, SMMED performs event detection as follows:

(1) Initialize Γ as the index set containing all $(c+1)$ classes, where the $(c+1)^{th}$ class is the null class. Recall that since this is a detection problem, many temporal segments will belong to the null class. Search the minimal temporal segment y_0 (e.g., 10% of the average length of training events, segment feature $\mathbf{x}^{y_0}$) and compute the classifier score of the null class $g_{c+1} = \mathbf{w}_{c+1}^T \mathbf{x}^{y_0}$. If g_{c+1} is smaller than the largest classifier score by 1, i.e., $g_{c+1} < \max_{p=1}^{c} g_p - 1$, remove the $(c+1)^{th}$ class from Γ; otherwise, label y_0 as a null class segment.

(2) If segment y_0 is not the null class, we sequentially construct a larger segment y by combining the segment y_0 with the incoming new frames. Remove the q^{th} class from Γ if the $g_q < \max_{p \in \Gamma} g_p - 1$.

(3) If no additional class is removed for a certain number of incoming frames, e.g., 30% of the average training event length, output the current segment as a detected event with a class label $arg \max_{p \in \Gamma} g_p$.

(4) After an event is detected, go to step (1) until the end of the test sequence.

5 Experiments

We evaluated SMMED against Multi-class SVM-based detectors (Eq. 2) on three databases; the MSR3D-Daily [22] database (3D depth videos), the UCF101 [18] database (RGB videos), and our collected Multi-Modal Action Detection (MAD) database (video, depth and the 3D body joints). The event data in both the MSR3D-Daily and UCF101 databases are organized in isolated clips, which is ideal for a controlled evaluation of

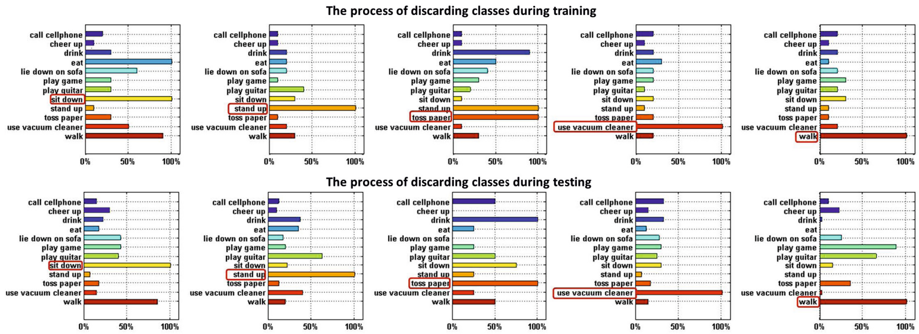

Fig. 3. Visualizing SMMED results on the MSRDaily database [22]. Each bar graph shows the portion of the event processed before the classes are discarded. The vertical axis depicts the class names (the red square highlights the true class), and the horizontal axis is the portion of the processed event. Each bar corresponds to one class. The end of the bar indicates when the class is discarded. For instance, for the class "sit down" (the top-left figure), the classes "call cellphone" and "cheer up" are discarded early because there is little overlap with the true action. **(Best viewed in color)**.

the detection performance. We used two metrics: (1) Percentage of Discarded Classes (PDC): the percentage of discarded classes when an event is detected; (2) Percentage of Early Labeling (PEL): the percentage of events that were reliably detected before the action ended (100% segment of an event). In addition, the MAD database has continuous events and the start and end of each action is provided, so the detection performance is easy to evaluate in a more realistic scenario.

5.1 MSRDaily Activity 3D Database

The MSRDaily Activity database [22] contains 3D depth clips of 10 subjects performing 16 daily activities. The resolution of the 3D depth frames is 640×480. Each subject performed each activity twice. There are a total of 320 activity instances organized in 10 groups (one group per person).

We used a fixed segmentation in each activity for a fair comparison. Each event was evenly split into 10 segments along time; The partial events were constructed as $[0\%, 10\%]$, $[0\%, 20\%]$,... ,$[0\%, 100\%]$ of each event for both training and testing, e.g., $[0\%, 100\%]$ includes all the frames of an event. For each partial event, we computed segment-based features using the DCSF (Depth Cuboid Similarity Feature) codes[1] provided by [23] and set the feature parameters according to [23]. As in [23], we used 12 of the 16 action classes. Note, our experiments are not directly comparable to [23](DCSF + SVM) because we are required to train and test on various temporal segments of the original event clips. In particular, [23] evaluated a classification problem: train and test only on the $[0\%, 100\%]$ segment of each event. Our experiments evaluated an event detection problem: the detectors must be trained over many temporal segments of each

[1] http://cvrc.ece.utexas.edu/lu/source_code.zip

event (the $[0\%, 20\%]$, $[0\%, 40\%]$,..., $[0\%, 100\%]$ segments), and test on many temporal segments in the test sequence.

We compared SMMED with the standard Multi-class SO-SVM (MSO-SVM), Eq. 1. MSO-SVM was trained using the Multi-Class SVM [5] of liblinear[2]. We used 5-fold-cross-validation on the 10 groups of sequences (2 groups per fold). For each cross-validation, 4 folds were used for training and the remaining 1 fold for testing. In the following experiments, we reported the best results for our SMMED (Eq.3) and the MSO-SVM detectors (Eq. 1) by tuning parameters over the 5-fold-cross-validation.

Fig.3 visualizes the training (the first row) and testing (the second row) process of discarding classes using SMMED. In each figure, the true classes are highlighted with a red square. Starting from the first column, the true classes are "sit down", "stand up", "toss papers", "use vacuum cleaner" and "walk". Each bar represents the percentage of time that an event class is considered as a candidate for the event. The end of the bar indicates the time when the action is discarded. Observe that in Fig. 3, many classes can be discriminated from the true class in the early stages (most bars stopped before 50% of the events). Moreover, the test bar graphs (the second row) and the training bar graphs (the first row) for the same true class show similar process of discarding classes. Recall that in SMMED, discarding a class means that the classification score for this class will not be computed in the later segments. This saves valuable time, especially when processing a large number of classes.

Table 1. Averaged recognition accuracy over 5-fold-cross-validation in MSRDaily database [22]. We compared the MSO-SVM detectors (different recognition results in different temporal segments) and our SMMED approach (the unique class labeling). Note, although we used the same DCSF codes provided by [23](DCSF+SVM), our experiment is not directly comparable to [23] because we address the detection problem not classification.

Segments	MSO-SVM	SMMED
$[0\%, 20\%]$	50.4%	**73.2**%
$[0\%, 40\%]$	63.8%	
$[0\%, 60\%]$	65.8%	
$[0\%, 80\%]$	68.8%	
$[0\%, 100\%]$	68.3%	

Table. 1 compares the average classification accuracy of SMMED against MSO-SVM. For all events, partial temporal segments were constructed using $[0\%, 20\%]$ to $[0\%, 100\%]$ frames of each event. Both MSO-SVM and SMMED were trained using all partial segments. Then MSO-SVM was tested on each partial segment of the test events respectively, and the recognition accuracy was computed at each segment by selecting the class with the highest score. SMMED, on the other hand, used all partial segments of the test events, and only discard classes until output a unique recognition accuracy at the last partial segment. Table. 1 shows that SMMED has higher recognition accuracy than the MSO-SVM for any segment. Moreover, we compared the detection

[2] `http://www.csie.ntu.edu.tw/~cjlin/liblinear/`

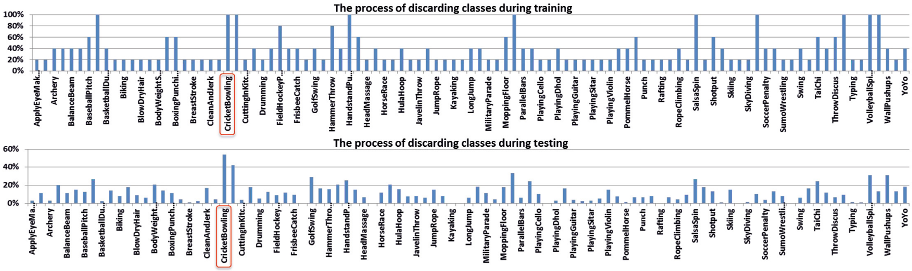

Fig. 4. Visualization of SMMED results of the true class "CricketBowling" in the UCF-101 dataset [18]. The horizontal axis lists the class names (the red square highlights the true class), and the vertical axis indicates the period or length that a particular class remains as a candidate action. Observe that most classes are discarded at very early stages (the short bars).

efficiency of SMMED against the MSO-SVM detectors in Table. 2. For a fair comparison against MSO-SVM, we used the same detection strategy described in Section 4.3 but the classifier matrix $\mathbf{W}$ trained by MSO-SVM. Table. 2 shows the average results over 5-fold-cross-validation for both SMMED and the MSO-SVM detector. Observe Table. 2, SMMED gets a higher Percentage of Discarded Classes (PDC) and Percentage of Early Labeling (PEL) than the MSO-SVM detector, being better suited for early activity detection. Specifically, 43.9% classes were discarded when test events were detected, and 39.2% of test events were identified without using all the frames.

Table 2. Averaged Percentage of Discarded Classes (PDC) and Percentage of Early Labeling (PEL) over 5-fold-cross-validation on MSRDaily [22]. We compared the MSO-SVM detectors and our SMMED approach.

	MSO-SVM	SMMED
PDC	29.1%	$\mathbf{43.9\%}$
PEL	13.8%	$\mathbf{39.2\%}$

5.2 UCF 101 Database

The UCF-101 database [18] contains 13320 video clips for 101 action classes. For each action class, the video clips were divided into 25 groups. Each group has 4-7 clips sharing common settings (e.g., similar background, same actors). All videos were recorded at 25fps and have a resolution of 320×240 pixels.

Similar to the MSRDaily experiment, we also constructed fixed partial segments for each event. The temporal segments used to evaluate the event detectors were $[0\%, 20\%]$, $[0\%, 40\%]$, $\cdots$, $[0\%, 100\%]$, where the interval $[0\%, 100\%]$ covers all frames of an event. We built a Bag-of-Words (BoW) representation with 4000-cluster codebooks by clustering 162-dimensional space-time interest points (STIP) descriptors provided by [18]. The segment-based feature of each temporal segment was computed as a histogram on the codebooks, the standard BoW. 5-fold-cross-validation was computed over the 25 groups of sequences (5 groups per fold). For each cross-validation, 4 folds were used for training and the remaining 1 fold for testing.

Table 3. Averaged recognition accuracy over 5-fold-cross-validation on UCF101 [18] for SMMED and MSO-SVM

Segments	MSO-SVM	SMMED
$[0\%, 20\%]$	35.0%	40.6%
$[0\%, 40\%]$	37.1%	
$[0\%, 60\%]$	39.4%	
$[0\%, 80\%]$	40.3%	
$[0\%, 100\%]$	$\mathbf{40.9}\%$	

Table 4. Averaged Percentage of Discarded Classes (PDC) and Percentage of Early Labeling (PEL) over 5-fold-cross-validation in UCF-101 [18] for SMMED and MSO-SVM

	MSO-SVM	SMMED
PDC	20.4%	$\mathbf{97.6}\%$
PEL	0.5%	$\mathbf{95.9}\%$

Fig. 4 visualizes the training (the first row) and testing (the second row) used by SMMED to partially discard classes. In this case, we have more classes than in the MSRDaily database. The horizontal axis depicts the class types and the vertical axis is the portion of the event for which the classes are active. We can see in Fig. 4 that most classes were discarded very early-on for both training and testing (most classes were discarded within 40% of the event).

Table. 3 shows the average recognition accuracy of SMMED (which output a unique class label after using all temporal segments), against MSO-SVM detectors (which output different classes on different temporal segments). As in the previous experiment, we used the classifier matrix $\mathbf{W}$ trained by MSO-SVM in the detection strategy (described in Section 4.3). The MSO-SVM detector performs similar to SMMED when using partial events higher than 80%. On the other hand, Table. 4 shows that SMMED gets much a higher Percentage of Discarded Classes (PDC) and Percentage of Early Labeling (PEL) than the MSO-SVM detectors. SMMED is better suited for online detection: 97.6% classes were discarded when an event was identified, and 95.9% events were detected before all frames were observed.

5.3 Multi-modal Action Detection (MAD) Database

The event sequences in both the MSRDaily and UCF101 databases are only available as isolated clips. In real applications, event detection is performed on streaming data. Manually concatenating the isolated clips results in discontinuous time series, and is not a very realistic scenario. Unfortunately, there are very few publicly available databases with labels for practical human action detection. This section describes the Multi-Modal Action Detection (MAD) database[3] for multi-class event detection. MAD contains 40 sequences of 20 subjects (2 sequences per subject) performing 35 activities in each of the sequences. The length of each sequence is around 2-4 minutes (4000-7000 frames).

[3] The MAD database and labels can be downloaded from `humansensing.cs.cmu.edu/mad/`.

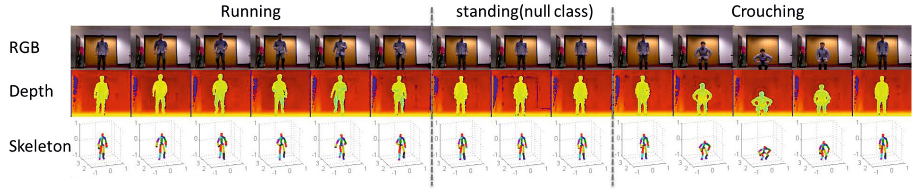

Fig. 5. Example frames of the Multi-Modal Action Detection (MAD) database

We recorded three modalities: RGB video (240 × 320), 3D depth (240 × 320), and a body-joint sequence (3D coordinates of 20 joints per frame). All data was recorded using the Microsoft Kinect sensor in an indoor environment. The 35 actions include full-body motion (e.g., Running, Crouching, jumping), upper-body motion(e.g., Throw, Basketball Dribble, Baseball swing), and lower-body motion (e.g., kicking). Each subject performs all the 35 activities continuously, and the segments between two actions are considered the null class (i.e., the subject is standing). Fig. 5 shows some example frames from the MAD database. The following experiments were performed using the 40 body-joint sequences.

We trained (35+ null class = 36)-class event detectors using both SMMED and MSO-SVM. The temporal segments used in training were the labeled event segments of the 35 event classes and the unlabeled segments (the null class, see the standing-by frames in Fig. 5). The feature for each temporal segment was computed in five steps: (1) Align all the body joints across frames using a 3D affine transformation; (2) Compute three descriptors using the aligned 3D-body-joints in each frame: the bone angles between joint pairs, differences of body-joint coordinates between the current and its previous frame, and average differences of body-joint coordinates between the current and its previous 10 frames; (3) Build a Bag-of-Word (BoW) with 100 codebooks for each of the three descriptors respectively; (4) Compute the frame features: For each frame, compute the three 100-dimensional BoW histograms, and concatenate them into a 100 × 3 = 300 dimensional frame feature vector; (5) Compute the segment features as the sum of frame features within each segment. After the SMMED and MSO-SVM classifiers were trained, SMMED-based action detection was done as described in Section 4.3. MSO-SVM was computed using [7] (i.e., MSO-SVM + Dynamic Programming(DP)), where DP searches the optimal temporal segmentation by enforcing the MSO-SVM objective. To allow MSO-SVM+DP [7] for on-line event detection, DP was solved in a sliding window (the maximum frame length of training events) moving through the test sequence.

For each method, we performed five-fold-cross-validation over the 20 subjects (4 subjects per fold). In each cross-validation, the labeled segments of four folds are used to train SMMED and MSO-SVM in [7]. The remaining sequence in the one fold is used for event detection. For instance, in the first cross-validation, the sequences of the 1^{st}-4^{th} subject are used for testing (4 × 2 = 8 sequences), and the sequences of the 5^{th}-20^{th} subject for training (16 × 2 = 32 sequences). Fig. 6 shows the frame-level detection results on 2 of the 8 test sequences. For each test sequence, the three bars are the ground truth frame-level labels, result of [7], and SMMED respectively. Different colors in the bars denote different class labels. Observe the bars, SMMED

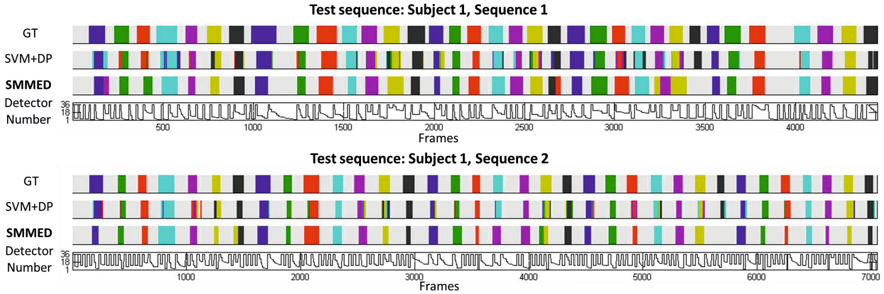

Fig. 6. Frame-level detection results on 2 test sequences in the MAD database. For each sequence, the three bars represent the ground truth frame-level labels (top), result of [7] (denoted as "SVM+DP" in the middle row), and SMMED respectively. In the curve figures below the detection bars shows the number of detectors used by SMMED detection. **(Best viewed in color)**.

produces fewer fragmented class labels than [7] around each true event. Quantitatively, we compared two event-wise measures: (1) Precision(Prec): the percentage of correctly detected events over all the detected events, the detected event is correct if it overlaps with 50% segment of the ground truth event; (2) Recall (Rec): the percentage of correctly detected events over all the ground truth events. Averaging over the 5-fold cross-validation, SMMED reached higher Precision ($Prec = 59.2\%$,$Rec = 57.4\%$) than [7] ($Prec = 28.6\%$, $Rec = 51.4\%$)) with comparable recall. The figures in the top of Fig. 6 shows the number of detectors used by SMMED detection. Observe that for most frames, the detector numbers are much less than the total number of event classes, i.e., 36 classes (35 activity classes + 1 null class).

6 Conclusion

We have proposed SMMED, a maximum-margin multi-class early event detection method. Unlike standard multi-class approaches, SMMED sequentially discards classes that can be early discriminated from the true class, being more efficient when detecting large number of classes. In our experiments, SMMED typically discarded about half of the classes before detecting the event. Experiments on databases with three different modalities, i.e., depth videos, RGB videos and body-joint sequences have shown that SMMED is more efficient, temporally consistent and accurate for multi-class event detection than MSO-SVM. In addition, we have released the CMU-MAD database a multimodal activity detection database with 20 subjects performing 35 actions. SMMED is a supervised detection system, in future work, we will explore the use of SMMED in an unsupervised and semi-supervised setting.

Acknowledgments. This work was partially supported by Samsung Electronics and by the National Science Foundation (NSF) under the grant RI-1116583. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.

References

1. Aggarwal, J., Ryoo, M.: Human activity analysis: A review. ACM Computing Surveys (CSUR) 43(3) (2011)
2. Bengio, S., Weston, J., Grangier, D.: Label embedding trees for large multi-class tasks. In: NIPS (2010)
3. Brand, M., Kettnaker, V.: Discovery and segmentation of activities in video. PAMI 22(8), 844–851 (2000)
4. Chapelle, O.: Training a support vector machine in the primal. Neural Computation 19(5), 1155–1178 (2007)
5. Crammer, K., Singer, Y.: On the algorithmic implementation of multi-class svms. JMLR, 265–292 (2001)
6. Gall, J., Yao, A., Razavi, N., Van Gool, L., Lempitsky, V.: Hough forests for object detection, tracking, and action recognition. PAMI 33(11), 2188–2202 (2011)
7. Hoai, M., Lan, Z., De la Torre, F.: Joint segmentation and classification of human actions in video. In: CVPR (2011)
8. Hoai, M., De la Torre, F.: Max-margin early event detectors. In: CVPR (2012)
9. Hongeng, S., Nevatia, R., Bremond, F.: Video-based event recognition: activity representation and probabilistic recognition methods. CVIU 96(2), 129–162 (2004)
10. Ke, Y., Sukthankar, R., Hebert, M.: Event detection in crowded videos. In: ICCV (2007)
11. Laptev, I., Marsza, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR (2008)
12. Mitra, S., Acharya, T.: Gesture recognition: A survey. TSMC-C 37(3), 311–324 (2007)
13. Niebles, J., Wang, H., Fei-Fei, L.: Unsupervised learning of human action categories using spatial-temporal words. IJCV 79(3), 299–318 (2008)
14. Niu, W., Long, J., Han, D., Wang, Y.F.: Human activity detection and recognition for video surveillance. In: ICME (2004)
15. Oh, S., Rehg, J., Balch, T., Dellaert, F.: Learning and inferring motion patterns using parametric segmental switching linear dynamic systems. IJCV 77, 103–124 (2008)
16. Simon, T., Nguyen, M., De La Torre, F., Cohn, J.: Action unit detection with segment-based SVMs. In: CVPR (2010)
17. Sminchisescu, C., Kanaujia, A., Li, Z., Metaxas, D.: Conditional models for contextual human motion recognition. In: ICCV (2005)
18. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human action classes from videos in the wild. In: CRCV-TR-12-01 (2012)
19. Swears, E., Hoogs, A.: Learning and recognizing complex multi-agent activities with applications to american football plays. In: IEEE Workshop on the Applications of Computer Vision (2012)
20. Tapia, E., Intille, S., Haskell, W., Larson, K.: Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor. In: IEEE Int. Symp. Wearable Computers (2007)
21. Tsochantaridis, I., Joachims, T., Hofmann, T., Atun, Y.: Large margin methods for structured and interdependent output variables. JMLR 6, 1453–1484 (2005)
22. Wang, J., Liu, Z., Wu, Y., Yuan, J.: Mining actionlet ensemble for action recognition with depth cameras. In: CVPR (2012)
23. Xia, L., Aggarwal, J.: Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In: CVPR (2013)