Large scale continuous visual event recognition using max-margin Hough transformation framework

Bhaskar Chakraborty, Jordi Gonzàlez and F. Xavier Roca

{*bhaskar,poal,xavir*}*@cvc.uab.es*
Computer Vision Center, Universitat Autònoma de Barcelona, Catalonia (Spain)

Abstract

In this paper we propose a novel method for continuous visual event recognition (CVER) on a large scale video dataset using max-margin Hough transformation framework. Due to high scalability, diverse real environmental state and wide scene variability direct application of action recognition/detection methods such as spatio-temporal interest point (STIP)-local feature based technique, on the whole dataset is practically infeasible. To address this problem, we apply a motion region extraction technique which is based on motion segmentation and region clustering to identify possible candidate "event of interest" as a preprocessing step. On these candidate regions a STIP detector is applied and local motion features are computed. For activity representation we use generalized Hough transform framework where each feature point casts a weighted vote for possible activity class center. A max-margin frame work is applied to learn the feature codebook weight. For activity detection, peaks in the Hough voting space are taken into account and initial event hypothesis is generated using the spatio-temporal information of the participating STIPs. For event recognition a verification Support Vector Machine is used. An extensive evaluation on benchmark large scale video surveillance dataset (VIRAT) and as well on a small scale benchmark dataset (MSR) shows that the proposed method is applicable on a wide range of continuous visual event recognition applications having extremely challenging conditions.

Keywords: Continuous visual event, Large scale, Max-margin Hough transform, Event detection

1. Introduction

Visual event recognition i.e. recognition of semantic spatio-temporal visual patterns such as "waving", "boxing", "getting into vehicle" and "running" is a fundamental Computer Vision problem. An enormous amount of work on this topic can be found in literature survey [1–4]. Recently, research in this field is directing towards *continuous visual event recognition* (CVER) where the goal is to both recognize an event and to localize the corresponding space-time

volume from large continuous video [5] like in object detection in images where only spatial location is important. This area is more closely related to the real world video surveillance analytics need than the current research which aims to classify a prerecorded video clip of a single event. Accurate CVER would have direct and far reaching impact in surveillance, video-guided human behaviour analysis, assistive technology and video archive analysis.

Figure 1: Examples of 6 scenes present in the VIRAT dataset [5] for large scale activity detection which explains different challenges: realism and natural scenes, diversity, quantity and wide range of scene resolution.

The task of CVER, i.e. the activity detection on large scale real world video surveillance dataset, is an extremely challenging task and current state-of-the methods for 2D small scale action recognition become infeasible to apply. One of the main challenges for CVER is the scalability, e.g. a CVER dataset like VIRAT dataset [5] contains 23 event types distributed throughout 29 hours of video. The other difficulties are due to i) natural appearance since the events are recorded in a real world scenario, ii) huge spatial and temporal coverage which affects the video resolution, e.g. the human heights within videos range $25 \sim 200$ pixels constituting $2.4 \sim 20\%$ of the heights of the recorded videos with an average being about 7%, iii) diverse event types and iv) huge variability in view-points, scenes and subjects (See Figure 1).

Among all these above mentioned difficulties, action detection in video (both small and large scale) is a challenging problem mainly due to the scalability of its search space. Without knowing the location, temporal duration and the spatial scale (spatial resolution of the activity) of the action, an exhaustive search is a NP-hard problem. For example, a one minute video sequence of size $160 \times 20 \times 1800$ contains more than 10^{14} spatial sub-volumes of various sizes and locations [6]. To solve this issue there are methods like discriminative sub-volume search [6], unsupervised random forest indexing [7]. Although promising, these works always use small scale video datasets like KTH[1] and MSR action

[1] *http : //www.nada.kth.se/cvap/actions/*

datasets [6] where the challenges present in CVER, mentioned above, are absent.

To solve the search space complexity of CVER, it is necessary to apply a motion region identifier to roughly detect the motion *region of interest* where the events to be searched may appear. Oh et al. [5] apply a multi-object tracking using frame difference, and the obtained tracks are divided into detection units resulting over $20K$ units, as a preprocessing step. This division of detection unit by a fixed amount always misses some events that are having different duration. On the other hand, in our approach first a motion segmentation method similar to [8] is applied to obtain the primary candidate region set. The obtained regions are further joined using a region clustering technique based on action heuristics. Finally, we obtain on an average $3K$ candidate regions as opposed to $20K$ by [5] with a greater recall rate. This has a major impact towards the search space reduction and on achieving faster event detection in large scale.

Our method for event detection is related to several ideas recurring in the literature. Firstly, we use STIP detector which is successfully applied in 2D action recognition problems [9–13]. Several local features are computed such as histogram of oriented optical flow (HOF) [14], histogram of oriented gradient in 3D (HOG3D) [15] and extended SURF (ESURF)[16] at the detected STIPs. We use the idea of *local appearance codebook*[17] including bag-of-word approach [18, 19] to group the detected features into a set of *visual words* that represent an event class.

The next idea is to use the generalized Hough transformation (GHT) framework for object detection in images into event detection in videos. Originally developed for detecting straight line [20], Hough transforms are generalized to use for detecting generic parametric shapes [21]. Recently, GHT scheme is successfully used for detecting object class instances tracking and action recognition [22–27].

The concept of GHT usually refers to any detection process based on an additive aggregation of evidence, Hough votes, coming from local image/video elements. Such aggregation is performed in a parametric space called as Hough space, where each point corresponds to the existence of an instance in a particular configuration. The Hough space may be a product set of different locations, scales and aspects etc. The detection process is then reduced to finding maxima peaks in the sum of all Hough votes in the Hough space domain, where the location of each peak gives the configuration of a particular detected object/event instance.

The implicit shape model of Leibe et al. [23] and the max-margin hough transformation of Maji et al. [25] serve a baseline for our work. These works mainly focus on object detection. During training, they augment each visual words in the codebook with the spatial distribution of the displacements between the object center and the respective visual word location. Using max-margin setup the weights of each visual words are learned. At the detection time, these spatial distributions are converted into Hough votes withing the Hough transformation framework. The weights of the visual words are also used for extra information to the Hough votes.

To incorporate this idea into CVER, we need to extend the dimensionality of

the voting space since now each STIP will vote for a parallelepiped center i.e. the event center. To make it easier to understand, we scale each candidate event into a normalized cube and during training the interest point (feature) distributions along the cube center is learned for each event class. The scale information is also saved so that by using a simple reverse conversion the normalized cube can be transformed into the actual event parallelepiped. After obtaining a set of visual words from detected event features, a max-margin frame work similar to [25] is applied for learning weights of each visual words for each event class. For a test candidate region, the detected interest points (features) are matched with the event class visual words and weighted votes for the possible event center are obtained in the Hough voting space. The votes corresponding to the peaks of Hough space reveal the possible hypothesis of the detected events in the actual video. Finally, a verification Support Vector Machine (SVM) designed for the particular event class is used to obtain the recognition score.

The main advantage of using a GHT framework is, it avoids the need of exhaustive search like in sliding window technique, which is infeasible to apply in CVER. GHT directly works on the STIPs and the local features that are extracted from the candidate motion regions. An instant probabilistic score can be obtained on the activity center and based on which an activity hypothesis can be generated. By using a verification SVM a more robust recognition is obtained, once the activity hypothesis is generated by GHT.

To test our approach we use large scale CVER dataset, VIRAT, proposed by [5]. Our result shows the state-of-the-art performance. To show the wise applicability of our method, we choose small scale video search dataset MSR [6] and also obtained above state-of-the-art result.

2. Related work

Action categorization/recognition and detection are important research topics and a large number of work have been found in the literature [1–4]. One type of approaches uses motion trajectories to represent actions and it requires target tracking [28, 29]. Another type of approaches uses background subtraction to obtain a sequence of silhouettes or body contours to model actions [2, 30]. Recently, action categorization use local spatio-temporal features computed on the detected spatio-temporal interest points (STIPs) to characterize the video and perform classification over the set of local features using a bag-of-word (BoW) approach [9–13, 16, 31]. Different shape and motion features are also applied to improve the action recognition [14, 15, 32–36]. Also, motion segmentation methods [37, 38] are sometimes used prior to local feature based methods [39, 40].

For action localization/detection, methods such as [41–44] apply a spatio-temporal template matching technique where actions are represented as a spatio-temporal template such as motion history[45] and space-time shapes[46]. Other methods employ multiple instance learning framework to obtain rough annotation [47], sub-volume search [6] using branch and bound method [48], Gaussian mixture model [49] and Random forest indexing [7] for action detection. Most

of these methods use STIP for action representation. In our method also a STIP detector, as described in [9], with 3 local features HOF [14], HOG3D [15] and ESURF [16] computed on each STIP is used for action representation. Having inspired by the success of the Hough transform-based methods in object detection [23–27, 50], methods like [22, 51] use Hough transform-based voting framework for action classification and localization where dense features are first computed and then a Hough forest is built to train the actions.

Although great success have been achieved in action recognition and detection, research towards large scale action detection (CVER) from video is less explored. Few works like [5, 49] show some progress in this area. For example, in the work of Cao et al. [49] result on TRECVID 2008 dataset [52] only using one action type, "running", is presented.

Our contribution: To address this gap, in this paper, we present a novel approach for large scale action detection. We propose

i) A generalized max-margin Hough transformation framework for activity detection which extends a similar framework [23, 25] applied to object detection to spatio-temporal domain.

ii) To reduce the initial search space we apply a region clustering based motion segmentation algorithm, which performs better than the tracking based region extraction technique proposed by Oh et al. [5].

iii) We propose a verification action recognition SVM to boost the final score of the detected event by the Hough transformation

iv) Finally, we test our approach on large scale video benchmark dataset [5] as well as on small scale video benchmark dataset [49]. The result shows state-of-the-art performance and validates the robustness of our method.

Figure 2 shows an overview of our proposed system.

The rest of the paper is structured as follows. We present an overview of the region clustering based motion segmentation in Section 3. In Section 4 the max-margin Hough transformation framework for event detection is described. We present our experiments in Section 5 and conclude in Section 6.

3. Region clustering based motion segmentation

To tackle the scalability issue of CVER it is important to reduce the action search space. Towards this goal, we apply a motion segmentation technique to identify roughly the motion regions where the *event of interest* may appear.

This step is important as it is reducing the search space. Due to the higher video resolution it is practically infeasible to apply any state-of-the-art STIP detector like [9–13, 16, 31]. But after the region extraction process, the candidate activity regions are usually smaller ($\sim 300 \times 300 \times 100$) and direct application of the STIP detector can be done easily.

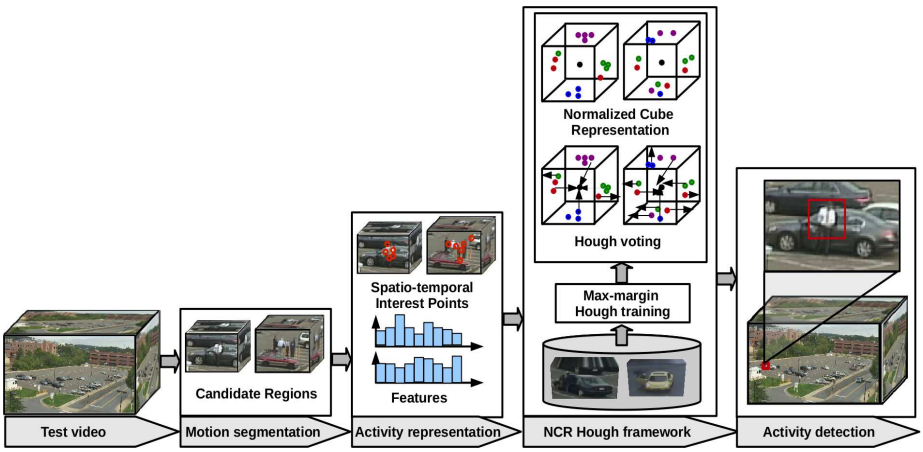


Figure 2: Large scale activity detection framework. Candidate activity regions are identified from the test video. STIPs and local features are computed and each candidate activity is represented by a normalized cube where each STIP (feature) votes for activity class center. Peaks of the Hough voting space are used for activity class recognition.

3.1. Background subtraction

As the first step of motion segmentation a background segmentation technique is applied as in [8]. In this method, each image pixel is modeled as a mixture of Gaussians and use an on-line approximation to update the model. The Gaussian distributions of the adaptive mixture model are then evaluated to determine which are most likely to result from a background process. Each pixel is classified based on whether the Gaussian distribution which represents it most effectively, is considered part of the background model. We apply background segmentation (See Algorithm 1) and after obtaining the result of background subtraction, a connected component algorithm [53] is applied to obtain motion regions per frame.

Algorithm 1 Background segmentation using mixture of Gaussian model. This function performs a GMM based background subtraction and calls standard connected component analysis, *connectedComponent* which returns region information and number of connected component.

Require: imStack: Having N frames of size $(H \times W)$.
Ensure: Motion regions (foreground) per frame.
1: Initialize Gaussian parameters, gP.
2: **for** $i \in N$ **do**
3: **if** $(i == 1)$ **then**
4: $bgModel = cvCreateGaussianBGModel(imStack(:,:,i), gP)$.
5: **else**
6: $cvUpdateBGStatModel(imStack(:,:,i), bgModel)$.
7: $fG = bgModel.foreground$
8: $connectedComponent(fG, \&regionInfo, \&noComponent)$.
9: $save(regionInfo, noComponent)$.
10: **end if**
11: **end for**
12: $cvReleaseBGStatModel(bgModel)$.

3.2. Region clustering and candidate event identification

Motion regions obtained by using the above step contain a large number of broken parts. To join these parts to a valid candidate region, a region clustering method is applied as described in Algorithm 2. In this approach, we first sort the obtained initial broken regions based on their Y-axis coordinate. Different broken parts are joined if they are within a horizontal and vertical thresholds, τ_x, τ_y, respectively. These thresholds are average bounding box dimensions of the *human* or *(human, object)* that are participating in the event of interest, which can be obtained from the training. We use the values $(\tau_x, \tau_y) = (80, 70)$ in our experiments. These values are obtained by computing the average bounding box sizes of the ground truth data of MSR and VIRAT datasets.

Algorithm 2 Region clustering algorithm

Require: regionInfo, noRegion: Region information per image frame.
Ensure: Pruned motion region of each image frame.

$regionInfo = sort(regionInfoY)$.
repeat
  $clusterNo = 0$.
  for $regCurr \in regionInfo$ **do**
    $clusterNo = clusterNo + 1$.
    $[x_C, y_C] = getCenter(regCurr)$.
    $regCurr.cluster = clusterNo$.
    for $regNext \in (regionInfo \setminus regCurr)$ **do**
      $[x_{C_N}, y_{C_N}] = getCenter(regNext)$.
      $x_D = fabs(x_C - x_{C_N})$.
      $y_D = fabs(y_C - y_{C_N})$.
      if $(x_D \leq \tau_x) \wedge (y_D \leq \tau_y) \wedge (\sim regNext.cluster)$ **then**
        $regNext.cluster = regCurr.cluster$.
      end if
    end for
  end for
  Initialize $flag$.
  for $clusterInd \in clusterNo$ **do**
    for $reg \in regionInfo$ **do**
      if $\sim flag(clusterInd)$ **then**
        $[x_L, y_L, x_R, y_R] = getCoordinate(reg)$
        $flag(clusterInd) = 1$
      else
        $[x_L, y_L] = min([x_L, y_L], getCoordinateL(reg))$
        $[x_R, y_R] = max([x_R, y_R], getCoordinateR(reg))$
      end if
    end for
    $clusterInfo.clusterNo = clusterInd$.
    $clusterInfo = putInfo([x_L, y_L, x_R, y_R])$;
  end for
  $regInfo = clusterInfo$.
  if $clusterNo == noRegion$ **then**
    $CONVERGENCE = TRUE$.
  end if
  $noRegion = clusterNo$.
until $CONVERGENCE \vee MAXITER$

After the region joining, a candidate region extraction as described in Algorithm 3, 4 is applied. Candidate region extraction is based on the *action heuristics*. The event of interest in VIRAT and MSR datasets are not moving in consecutive frames, since the actions are of type, "getting inside car", "open the trunk of a car" and "loading objects in the car" etc. or "clapping", "waving" and "boxing". So if these events are occurring along with other moving actions

like "walking" and "running" we are guaranteed to obtain fixed regions, corresponding to events of interest, along with some moving regions , corresponding to the moving actions, in consecutive frames. Based on these heuristic, identifying motion region is simply to identify region having some permissible region overlap in consecutive frames. To realize this goal, we apply a region search

Algorithm 3 Find candidate event/video of interest

Require: $pRegInfo$, $noPReg$: Pruned region information per image frame.
Ensure: Candidate event/video of interest.

for $imC \in N$ **do**
  for $reg \in pRegInfo_{imC} \wedge notChecked(reg)$ **do**
    $checked(reg)$.
    $[x_C, y_C] = getCenter(pRegInfo_{imC})$.
    $putInfo(regTack, getInfo(pRegInfo_{imC}))$.
    for $imNC \in (N \setminus imC)$ **do**
      $flag = 0$.
      for $regN \in pRegInfo_{imNC} \wedge notChecked(regN)$ **do**
        $[x_{C_N}, y_{C_N}] = getCenter(pRegInfo_{imNC})$.
        if $dist([x_C, y_C], [x_{C_N}, y_{C_N}]) \leq \tau_d$ **then**
          $checked(reg)$.
          $flag = 1$
          $putInfo(regTack, getInfo(pRegInfo_{imNC}))$.
        end if
      end for
      if $flag == 0$ **then**
        $extractCandidates(candidate, regTrack)$.
        $break$.
      end if
    end for
    if $flag == 1$ **then**
      $extractCandidates(candidate, regTrack)$.
    end if
  end for
end for
for $reg \in candidate$ **do**
  for $regN \in (candidate \setminus reg)$ **do**
    if $overlap(reg, regN)$ **then**
      if $(abs(reg.End) - regN.Start) \leq \tau_l)$ **then**
        $candidate = merge(reg, regN)$.
      end if
    end if
  end for
end for
$saveCandidate(candidate)$

Algorithm 4 Extract candidate event/video of interest

Require: $regTrack$: Information of motion region in consecutive frames having some permissible overlap.
Ensure: Candidate event/video of interest.

for $reg \in regTrakc$ **do**
  $regNext = getNext(regTrack, reg)$.
  if $overlap(reg, regNext) \leq \tau_a$ **then**
    $merge(reg, regNext)$.
  end if
  $candidate = getInfo(reg)$.
end for

technique to first obtain a chain of motion region that have a permissible region overlap within a threshold, τ_d. In our experiment, τ_d is in between 15%. This chain of motion region may contain some false alarm like, a person walking

at a slow rate. To avoid such outliers, we apply a second method (Algorithm 4) which takes only the regions having higher overlapping (τ_a) in consecutive frames. As a final step, we apply a region merging by putting a considerable frame gap (τ_l). With this, two candidate regions having τ_l frame gap and overlapping area withing τ_a are merged together. We use $\tau_a = 45\%$ and $\tau_l = 5$ in all our experiments.

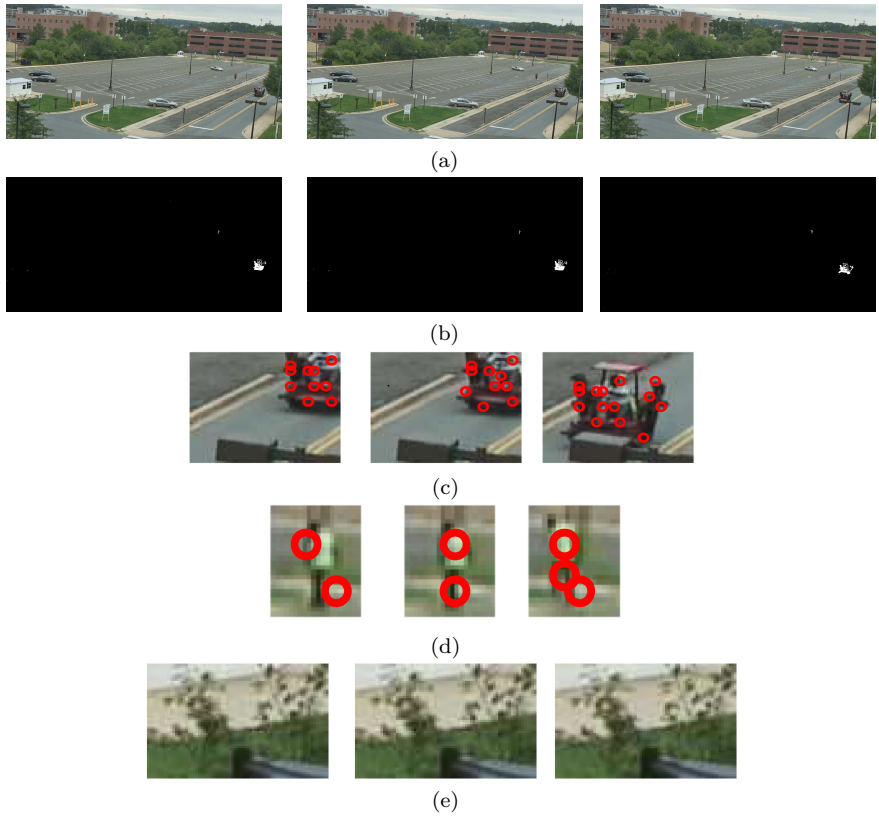


Figure 3: Output of the region clustering based segmentation algorithm. (a) Actual frames, here frame number 375, 376 & 405 are shown. (b) Corresponding motion segmentation regions. (c) & (d) The regions that have motion due to moving cars and moving human. (e) False region extraction due to sudden illumination change. Note that false motion regions where very little or no motion is present, no STIP is detected. But, motion regions due to moving cars or humans contain STIPs.

The proposed algorithms generate many motion regions from a video. Due to the use of background subtraction algorithm, it may suffer from illumination changes and may result in extracting several false motion regions. In our case, this limitation is automatically handled by the next phase of STIPs detection and motion features extraction. Since all the extracted motion regions obtained from the proposed preprocessing step are passed through a STIP detector and feature extraction phase, false motion regions due to sudden illumination change

would automatically be removed due to low number of detected STIPs on them. Figure 3 shows some example of the region clustering based motion segmentation technique. Here we present actual frames, corresponding motion segmentation region and candidate motion regions with detected STIPs (Figure 3.a-d). Note that the false motion detections due to illumination changes or low motion component do not contain any STIPs (Figure 3.d) and are automatically discarded. Figure 4 shows the regions containing an activity "person getting out

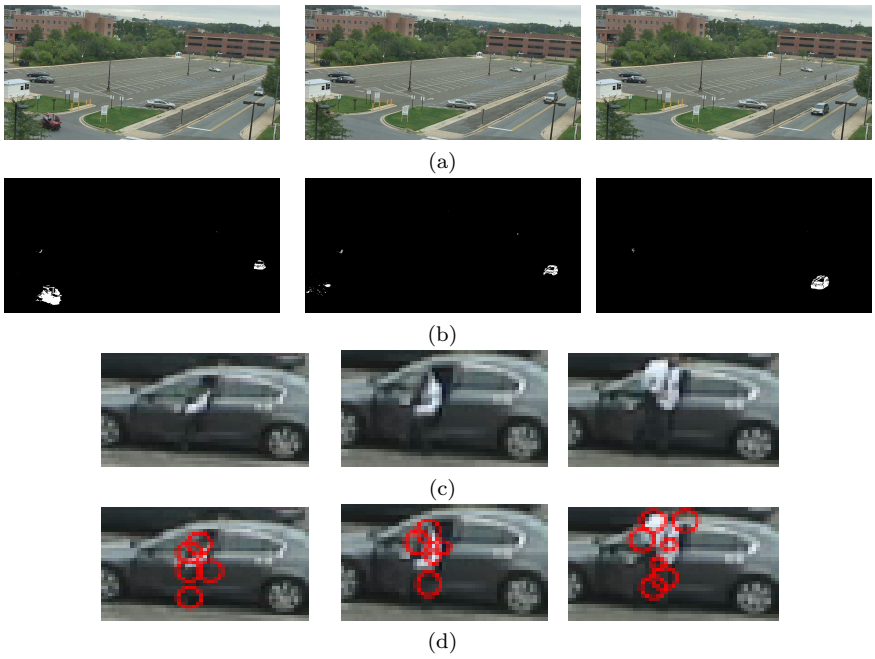


Figure 4: Output of the region clustering based segmentation algorithm with proper activity. (a) Actual frames, here frame number 659, 682 & 725 are shown. (b) Corresponding motion segmentation regions. (c) Person getting our of the car activity region (d) Detected STIPs.

of the car". Figure 4.a-d presents 3 samples of the actual frames, corresponding region segmentations, actual activity region and the detected STIPs.

4. Max-margin Hough transform framework for event detection

The general idea to apply a Hough transformation framework [23] into an action detection problem is to compute the probabilistic score which is obtained by adding up the votes from D-dimensional feature vectors extracted from a candidate video event in a Hough space $\mathcal{H} \subseteq \mathbb{R}^H$. In our case we apply a spatio-temporal interest point (STIP) detector [9] on candidate event (Figure 5) and the feature vector is the concatenation of HOF, HOG3D and ESURF.

So formally, let $\mathcal{A}$ be a candidate event having center at $r = \{x, y, t, s\}$ where $\{x, y, t\}$ is the coordinate of the center and s is the scale of the detection and

(a) (b)

Figure 5: Detected STIPs on the two events (a)"loading" and (b)"getting into vehicle" activity of the VIRAT dataset..

f_i is the feature vector computed at a location l_i. l_i is basically associated with a STIP $\{x_i, y_i, t_i, s_i\}$. So, the probabilistic score $S(\mathcal{A}, r)$, i.e. the confidence value that the center r is associated to the candidate event $\mathcal{A}$, can be obtained following [23, 25]

$$S(\mathcal{A}, r) = \sum_j p(\mathcal{A}, r, f_j, l_j) \quad (1)$$

$$= \sum_j p(f_j, l_j) p(\mathcal{A}, r | f_j, l_j) \quad (2)$$

Let C_i denotes the i^{th} codebook entry of the vector quantized space of features f. Assuming a uniform prior over features, f_j, local patches, l_j together with codebook entries, C_i, the score we get:

$$S(\mathcal{A}, r) = \sum_j p(\mathcal{A}, r | f_j, l_j) \quad (3)$$

$$= \sum_{i,j} p(C_i | f_j, l_j) p(\mathcal{A}, r | C_i, f_j, l_j) \quad (4)$$

This equation can further be simplified using the argument that $p(C_i|f_j, l_j)$ is equivalent to $p(C_i|f_j)$ since the codebook entries, C_i, are based only on the features, f_j. Furthermore, the term $p(\mathcal{A}, r|C_i, f_j, l_j)$ depends only on the matched codebook C_i and l_j,

$$S(\mathcal{A}, r) \propto \sum_{i,j} p(C_i | f_j) p(\mathcal{A}, r | C_i, l_j) \quad (5)$$

$$= \sum_{i,j} p(C_i | f_j) p(r | \mathcal{A}, C_i, l_j) p(\mathcal{A} | C_i, l_j) \quad (6)$$

The first term, $p(C_i|f_j)$ is the likelihood that the feature f_i is associated with the codebook entry C_i. This we define as,

$$p(C_i|f) = \begin{cases} \frac{1}{Z} exp(-\gamma sim(C_i, f)) & \text{if } sim(C_i, f) \leq t \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where Z is the constant of the probability distribution $p(C_i|f)$ and $sim(C_i, f) = \frac{1}{d(C_i, f)}$ where $d(C_i, f)$ is the distance between feature f and the codeword C_i. γ and t are positive constant.

The second term, $p(r|\mathcal{A}, C_i, l_j)$ is the probabilistic Hough vote for the activity center r which is estimated in the training phase by observing the distribution of the location of the codebooks relative to the activity center. The third term, $p(\mathcal{A}|C_i, l_j)$ is the weight of the codebook entry emphasizing the confidence of the codebook C_i at location l_j that matches the activity $\mathcal{A}$. Final term, $p(\mathcal{A}|C_i, l_j)$, can further be simplified by assuming that the probability $p(\mathcal{A}|C_i, l)$ is independent of the location,

$$p(\mathcal{A}|C_i, l) = p(\mathcal{A}|C_i) \propto \frac{p(C_i|\mathcal{A})}{p(C_i)} \tag{8}$$

While applying this framework in CVER (large scale activity detection) the computation of the second term, $p(r|\mathcal{A}, C_i, l_j)$, becomes complicated due to the in-feasibility to apply any state-of-the-art STIP detector or feature extraction method directly on the large scale video. We must work on the possible candidate regions either obtained from the ground truth or by applying a motion segmentation method (see Section 3) to extract motion regions where the candidate event of interest may appear. Then $p(r|\mathcal{A}, C_i, l_j)$ is the collection of distances $\{d_{x_j}, d_{y_j}, d_{t_j}, d_{s_j}\}$ between the STIP $\{x_j, y_j, t_j, s_j\}$ associated to l_j and the activity parallelepiped center r. These distances are the *votes* of l_js for r.

4.1. Learning the codebook weight using max-margin framework

The Equation 6 can further be simplified as a weighted vote for event video location over all codebook entries C_i. The key idea, as described in [25], is to observe that the score $S(\mathcal{A}, r)$ is a linear function of $p(\mathcal{A}|C_i)$ (Equation 8). Using this idea Equation 6 can be expressed as,

$$\begin{aligned} S(\mathcal{A}, r) &\propto \sum_{i,j} p(r|C_i, l_j) p(C_i|f_j) p(\mathcal{A}|C_i, l_j) && (9) \\ &= \sum_{i,j} p(r|C_i, l_j) p(C_i|f_j) p(\mathcal{A}|C_i) && (10) \\ &= \sum_{i} p(\mathcal{A}|C_i) \sum_{j} p(r|C_i, l_j) p(C_i|f_j) && (11) \\ &= \sum_{i} \lambda_i \times q_i(r) = \lambda^T Q(r) && (12) \end{aligned}$$

where $\lambda_i = p(\mathcal{A}|C_i)$ is the probability of the activity $\mathcal{A}$ given the codebook C_i. This is considered as a codebook weight. $Q^T = [q_1 q_2 \dots q_k]$ is the activation vector and q_i is given by,

$$q_i = \sum_j p(r|C_i, l_j) p(C_i|f_j) \tag{13}$$

For a given event and identity the summation over j is constant and is only a function of the observed features, locations (STIPs) and the estimated distribution over the centers for the codebook entry C_i.

To learn the weight vector λ, a max-margin optimization approach as describe in [25] is used. Starting from a set of training examples, $\{(q_i, y_i)\}_{i=1}^{L}$, where $y_i \in \{+1, -1\}$ is the label and q_i is the i^{th} training activity, we compute the activations $Q_i = Q(q_i)$ for each example by adding up the votes for each feature f_j extracted at location STIP l_j according to the Equation 13. So, the score assigned by the model to the instance i is $\lambda^T Q_i$. Weights are learned by maximizing this score on correct classification of events over the incorrect ones. This is done by using a max-margin frame work ([25]),

$$min_{\lambda,b,\xi} \qquad \frac{1}{2}\lambda^T\lambda + K\sum_{i=1}^{M}\xi_i \tag{14}$$

$$s.t.: \qquad y_i(\lambda^T Q_i + b) \geq (1 - \xi_i) \tag{15}$$

$$\lambda \geq 0, \xi_i \geq 0, \forall i = 1, 2, \dots L \tag{16}$$

This optimization is similar to the optimization problem of a linear Support Vector Macine [54], with an additional positive constrain on the weights. We use a traditional optimization package, CVX [2] [55], for solving this problem.

4.2. Overall detection technique

The proposed max-margin GHT frame work is run on the each extracted candidate region by the region extraction algorithm proposed in Section 3. After obtaining the votes in our Hough space $\mathcal{H} \subseteq \mathbb{R}^4$, a mean shift based clustering algorithm [56] is used to identify the location of the peaks in $\mathcal{H}$. By using these peaks, initial hypothesis of the event in actual video coordinate is obtained. This is computed based on the information of the participating STIPs in the peaks of $\mathcal{H}$.

From this initial hypothesis, the test candidate region is extracted and by using a verification SVM its class label is identified. This verification SVM is similar to the action recognition SVM approach in [9]. This is learned using the training activity features. In our experiment, we follow a leave-one-video out technique.

[2] http://standford.edu/∼boyd/cvx

Let $V = \{v_1, v_2, \ldots, v_n\}$ be the n videos where r activity classes are distributed as $A = \{a_1, a_2, \ldots, a_r\}$. Each activity class a_i contains k_i number of training samples. So to perform the activity detection in the video v_i, all the activity regions in v_i is removed from S to obtain a reduced activity set $\hat{A}$:

$$\hat{A} = \{a_j \backslash AR_{v_i}\} \tag{17}$$

where AR_{v_i} stands for the activity region in the video v_i. Using this set $\hat{A}$, r activity SVMs are learned by using [9]. To this end, first the features are grouped from each training activity regions in A. General k-means clustering algorithm in applied to this feature group to obtain the initial vocabularies. These vocabularies are compressed by using an Agglomerative Information Bottleneck (AIB) technique as described in [57] to obtain a compact activity visual codewords. Each sample activity is represented by using a histogram of these compact activity visual codewords as $hist_{a_{i_j}}$, where a_{i_j} denotes j^{th} training sample of the activity a_i. For each activity class i, a SVM is trained by using the $hist_{a_i}$. As mentioned above, after obtaining the initial hypothesis of the test activity region a_{test} from the proposed max-margin GHT technique, it is represented as a histogram of compact activity visual codewords as $hist_{a_{test}}$. We apply this histogram to the activity class SVM to obtain the final score of the activity region. In our experiment, we use intersection kernel for the SVM. The score obtained from the verification SVM is used as a final score of the detected region. This score is later used to compute ROC curves and average precision (AP) values.

Note that, in our approach max-margin Hough transformation is very important step since it gives actual activity location in a space-time domain, i.e. the start frame, end frame and the spatial location of the activity. So, Hough transformation is actually detecting the activity. For a robust recognition, the activity SVM is proposed. In the experimental result section (Section 5), we present the results of both with and without the SVM based verification and we obtain a significant gain in the AP values by using the proposed verification classifier.

5. Experimental results

To validate our proposed approach, experiments on two benchmark datasets are performed: VIRAT dataset [5] is used for large scale event detection and Microsoft Research Action (MSR) Dataset II [3], [6, 49] is used for small scale activity detection.

VIRAT video dataset: In our experiments we use the Release 1.0 [4] of the dataset which was publish in the CVPR'11 activity recognition challenge [5]. It contains 66 videos as *training set* with available ground truth annotation

[3] $http://research.microsoft.com/en-us/um/people/zliu/actionrecorsrc/default.htm$
[4] $http://www.viratdata.org/virat/virat_archive1.html$
[5] $http://www.umiacs.umd.edu/conferences/cvpr2011/ARC/$

and *scoring software*. The *training set* contains 3 scenes. Besides, in the above mentioned activity recognition challenge, 128 videos were latter released as *test videos* where total 6 scenes are present, three same scenes like in *training set* with three additional scenes (see Figure 1). The *test set* does not have *scoring software* or ground truth annotations. These videos are captured by stationary HD cameras (1080p or 720p). Some of them contains slight jitter due to environmental condition. Heights of humans within videos range $25 \sim 200$ pixels, constituting $2.3 - 20\%$ of the heights of recorded videos with average being about 7%. There are total 6 activities, i) *person loading an object to a vehicle*, ii) *person unloading an object from a vehicle*, iii) *person opening a vehicle trunk*, iv) *person closing a vehicle trunk*, v) *person getting inside into a vehicle* and vi) *person getting out of a vehicle*. This dataset is extremely challenging.

MSR action dataset II: This is an extended version of the Microsoft Research Action Data Set. It consists of 54 video sequences recorded in a crowded environment. The video resolution is 320×240 and frame rate is 15 fps. Each video sequence consists of multiple instances of the three actions: *hand waving*, *hand clapping*, and *boxing* multiple actions. There are in total 203 action instances of these three actions distributed in all the 54 video sequences. This dataset is a small scale activity recognition dataset compared to the VIRAT video dataset.

5.1. Large scale activity detection

For this experiment 66 videos are used where the ground truth annotation and *scoring software* are available. To prune the search space we apply the region extraction algorithm (Section 3). Region pruning algorithm (Algorithm 2) extracts $\sim 3K$ candidate regions from each video. To validate the candidate region we perform a "recall test". To this end, the region extraction algorithm is applied to each of the 66 videos of VIRAT. If there is a overlap of 75% between a candidate region and the ground truth, it is considered as *hit*. Table 1 shows the number of *hit* w.r.t the ground truth. We obtain quite high recall in each category (See Table 1). Our method extracts 60% more ground truth regions compared to the tracking based region extraction algorithm [5], where first tracking is used and then tracks are divided into units of 3-4 seconds segments (with 2 seconds overlap) each, resulting more than $20K$ detection units. This approach fails to detect the activity that are happening during longer duration than the detection units. On the other hand, as mentioned above, our region pruning method extracts on an average $3K$ regions per video and obtains a high recall rate (0.9680). Note that this recall test is very important to show the robustness of the region extraction algorithm. The purpose of the region extraction algorithm is to prune the initial search space. Recall test gives us a clear a idea of how good is the region extraction algorithm. Recall close to 1.0 means all the extracted region obtained by using the region extraction algorithm contains at least the ground truth activity regions along with the other motion regions.

Table 1: Comparison of recall test performed in the VIRAT dataset. Proposed region extraction algorithm outperforms the tacking based method of [5] to identify initial motion regions where activity of interest may present. Events categories are, (1) loading, (2) unloading, (3) opening trunk (4) closing trunk, 5 getting into vehicle and (6) getting out of vehicle. The numbers are presented as: recall value (# Ground truth region extracted / # Ground truth)

Event category	# Ground truth	Recall	
		Our approach	Oh et al. [5]
1	11	0.9090(10/11)	0.5454(6/11)
2	16	0.9375(15/16	0.5(8/16)
3	18	0.8888(16/18)	0.4444(8/18)
4	19	1.0(19/19)	0.4736(9/19)
5	61	0.9836(60/61)	0.2950(18/61)
6	63	0.9841(62/63)	0.2222(14/63)
Total	188	0.9680(182/188)	0.3351(63/188)

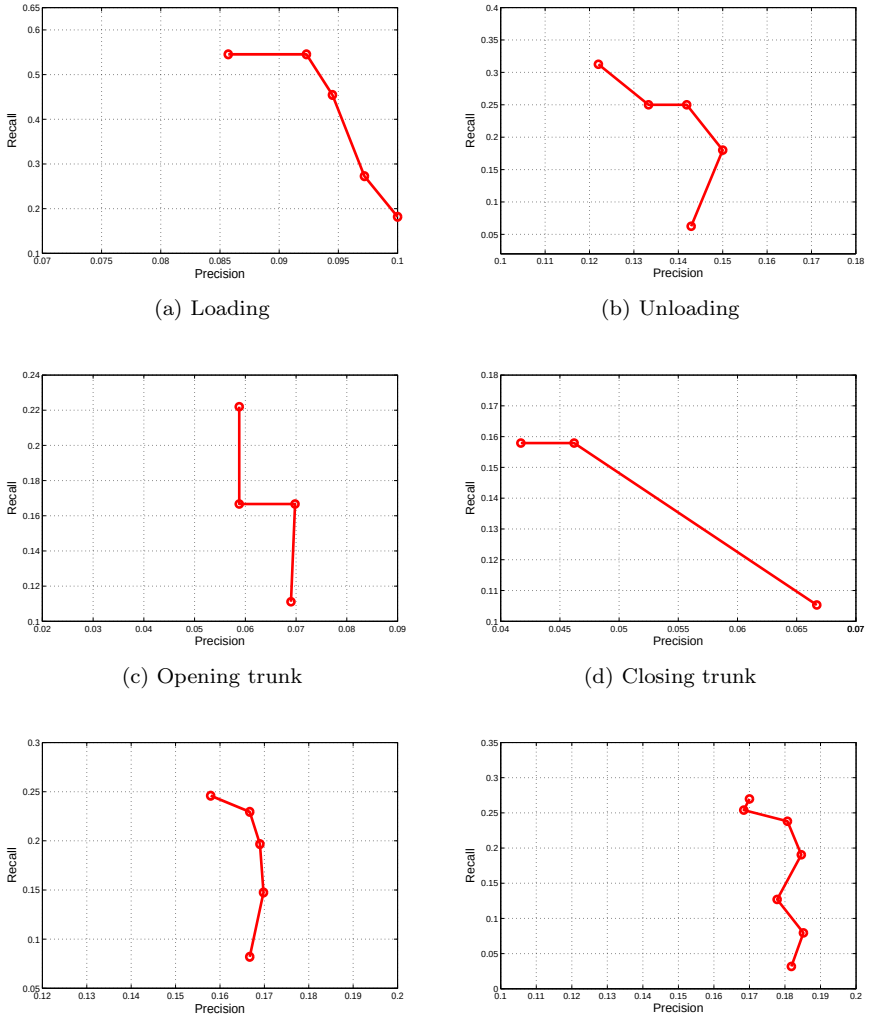
5.2. Detection scores

To evaluate the performance of our generalized Hough transform approach we use the *scoring software* and generate ROC curves for different activities by varying a threshold on the detection scores. In our experiment, we use a leave-one-video out cross validation technique to obtain the AP values for each activity. Figure 6 shows the obtained ROC curves and Table 2 presents the AP values of the 6 activity classes. We obtain best scores in "getting out of vehicle", "getting into the vehicle" and "unloading" activity classes. Our method outperforms the detection score of [5] where the score of only one activity class, "getting into the vehicle" is presented. The AP value of the activity class "getting into the vehicle" presented by Oh et al. [5] is ~ 0.007, where as in the same class we obtain an AP value 0.1660.

Table 2: Comparison of the AP values obtained from max-margin GHT + SVM and only max-margin GHT based activity detection in VIRAT dataset.

Activity class	Max-margin GHT + SVM	Max-margin GHT
Loading	0.0939	0.0345
Unloading	0.1380	0.0875
Opening trunk	0.0641	0.0548
Closing trunk	0.0515	0.0479
Getting in	0.1660	0.1035
Getting out	0.1784	0.1295

The main reason to obtain low detection rate in all the activity classes is the small number of activity samples in each class. For example, in all 66 videos there are only 11 samples for "loading" activity class, 18 and 19 samples for "opening trunk" and "closing trunk" activity classes respectively. Moreover, often the activity samples are highly occluded by other objects and the visible



(a) Loading (b) Unloading

(c) Opening trunk (d) Closing trunk

(e) Getting in (f) Getting out

Figure 6: ROC curves of the all six activities obtain from 66 videos. The activities are (a) loading (b) unloading (c) opening trunk, (d) closing trunk (e) getting into vehicle and (c) getting out of vehicle. We obtain best result in "getting out of vehicle" activity.

area is quite small (See Figure 7). Due to this problem enough motion features are not obtained and affects the performance of max-margin GHT and verification SVM. Note that, the activity classes contain higher number of samples like "getting into the vehicle" (61 samples) and "getting out of vehicle" (63 samples), the AP values are higher compared to other classes.

In Table 2, we also show the AP values without verification SVM. We gain

Figure 7: Activity "getting into vehicle", where the human is occluded by other cars and the activity is hardly visible. Due to this occlusion not enough motion features are obtained and hence, overall detection performance suffers.

$\sim 4-6\%$ in AP values for all the 6 activity classes by using the verification SVM. Figure 8 shows sample frames with detection of "getting into vehicle" and "getting out of vehicle" activities.

5.3. Computational time

For large scale activity detection the computational time is an important factor. We present a thorough analysis of the computational time of different phases of our overall activity detection system. The computational time of the region clustering based motion segmentation algorithm (Section 3) is highly depend on the total number of frames per video. The region extraction algorithm takes $\sim 2.5 hours$ per video, for the videos with more than $10K$ frames. For the videos with less than $10K$ frames, the algorithm takes $\sim 1.8 hours$ per video. The computation time of the feature computation is $\sim 15 mins$ per candidate region. The max-margin GHT takes $\sim 7 mins$ per candidate region for the hypothesis generation.

5.4. Activity detection in small scale

To perform experiments on small scale dataset we use MSR action dataset II. Most state-of-the-art approaches like, [6, 7, 49] use this dataset as cross-data action recognition where KTH [6] is used as training dataset and MSR is used as test dataset. All these methods apply *model adaptation* to perform the cross-dataset action detection. Note that although the work of Cao et al. [49] is about cross-data action recognition, the final detection score is obtained after the ground truth adaptation.

Since our approach does not design to perform cross-dataset action detection rather our goal is to present max-margin GHT framework on action detection,

[6] $http://www.nada.kth.se/cvap/actions/$

(a) Getting into vehicle

(b) Getting out of vehicle

Figure 8: Sample frames with detection bounding box of two activity classes, (a) getting into vehicle and (b) getting out of vehicle.

we split MSR action dataset II into two groups: first 16 videos as *training set* and rest 38 videos as *test set*. Ground truth annotations are used to separate actions in the *training set*. We first apply our motion segmentation approach described in Section 3 and apply recall test using ground truth annotation to validate the proposed region extraction method. We obtain 100% recall in all three actions, *hand waving*, *hand clapping* and *boxing*, present in *test set*.

To evaluate the the detection results of our algorithm, we follow the same technique as proposed by Cao et al. [49]. Let $\mathbf{Q}^g$ be the ground truth instances, $\mathbf{Q}^g = \{Q_1^g, Q_2^g, \ldots, Q_m^g\}$, and $\mathbf{Q}^d$ be the instances detected by the algorithm, $\mathbf{Q}^d = \{Q_1^d, Q_2^d, \ldots, Q_n^d\}$. $H(Q_i^g)$ denotes whether a ground truth instance Q_i^g is

detected and $T(Q_j^d)$ denotes if a detected instance Q_j^d is properly matched with the ground truth set $\mathbf{Q}^g$. These values can be calculated as,

$$H(Q_i^g) = \begin{cases} 1 & \text{if } \exists Q_k^d, \text{ s.t. } \frac{|Q_k^d \cap Q_i^g|}{|Q_i^g|} > \delta_1 \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

$$T(Q_j^d) = \begin{cases} 1 & \text{if } \exists Q_k^g, \text{ s.t. } \frac{|Q_k^g \cap Q_j^d|}{|Q_j^d|} > \delta_2 \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

where $|\cdot|$ denotes the area of the video instance and δ_1, δ_2 use to judge the overlapping ration. δ_1 and δ_2 are set to 0.125 as proposed by Cao et al. [49].

Given a set of detected instances the *precision* and *recall* can be computed using the values of H and T,

$$Precision = \frac{\sum_{i=1}^{m} H(Q_i^g)}{n} \tag{20}$$

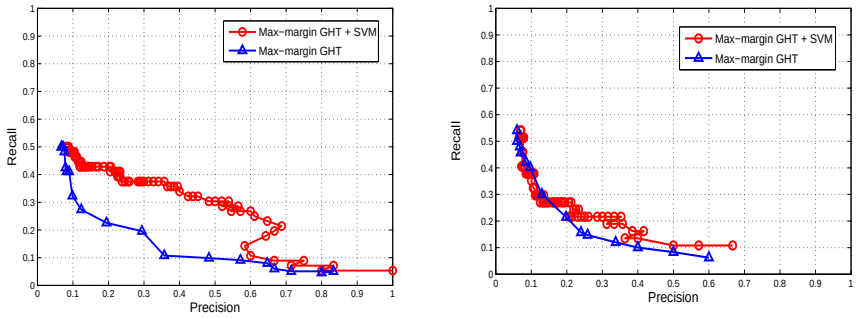
$$Recall = \frac{\sum_{j=1}^{n} T(Q_j^d)}{m} \tag{21}$$

Table 3: Comparison of the average precision value using max-margin GHT + SVM and only max-margin GHT based activity detection in MSR dataset.

Methods	Boxing	Hand clapping	Hand waving
Max-margin GHT + SVM	0.4571	0.2327	0.4938
Max-Margin GHT	0.3325	0.2217	0.3555

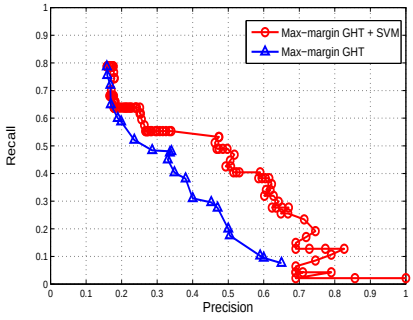
Varying the threshold on the detection score ROC curve can be obtain for three actions in MSR dataset (See Figure 9). We compare average precision (AP) of the three actions in MSR with other state-of-the-art approaches in Table 4. We obtain higher AP in *boxing* and *hand waving* actions. The AP value for hand clapping is low compared to the other state-of-the art approach. This due the low number of samples in this category and often clapping action is performed together with waving and boxing. Because of this the over all performance of this action suffers. In Table 3 the average precision values obtained from only generalized Hough transformation and by using verification SVM are shown. We obtain significant improvement of the precision value when a verification is SVM is used. In particular, we gain $\sim 14\%$ and $\sim 12\%$ precision values in hand waving and boxing actions respectively, where as in hand clapping the gain is $\sim 1\%$.

Figure 10 shows true detection of different actions in MSR along with some failure cases. Detection bounding boxes are presented for boxing, clapping and waving actions as a true detections (Figure 10.a-c). In the Figure 10.d & e two failure cases are shown where only one of the two actions are detected. This is due the the closeness of the two actions where the STIPs contributing more to one actions gets higher detection score.



(a) Boxing

(b) Hand clapping



(c) Hand waving

Figure 9: ROC curves of the three actions, (a) boxing, (b) hand clapping and (c) hand waving of MSR dataset.

6. Conclusion

In this paper we present a novel approach for event detection in large scale activity dataset using max-margin Hough transformation framework. We tackle the large search space by applying a region extraction algorithm which is based on motion segmentation and region clustering. This algorithm is simple, fast and obtains better recall compared to tracking based approaches. For activity detection, generalized Hough transformation technique is applied which is popular in the field of object recognition. We apply a max-margin framework for learning the weights of the visual vocabularies. Finally, a verification action classifier is used to obtain the overall score of the detected event hypothesis obtained from Hough transformation framework. The proposed algorithm avoids the need of exhaustive search which is infeasible for activity detection problem. It works directly on the computed features and generates activity hypothesis. A significant gain in AP values is obtained by using the final verification action classifier.

To evaluate our approach, large scale activity detection dataset, VIART is

Table 4: Comparison of average precision (AP) values of the three actions of MSR dataset with other state-of-the-art approaches. Note that both Yu et al. [7] and Cao et al. [49] use cross-data action detection approach with full set of MSR, instead we train on first 16 videos of MSR and test on the rest 38 videos to comply with our proposed algorithm setup. Note that the detection score of [49] is obtained after groundtruth adaptation.

AP	Boxing	Hand clapping	Hand waving
Our approach	**0.4571**	0.2327	**0.4938**
Yu et al. [7]	0.3029	**0.3155**	0.4923
Cao et al. [49]	0.1748	0.1316	0.3671

(a) Boxing (b) Hand clapping (c) Hand waving

(d) Detection of waving (e) Detection of boxing

Figure 10: Activity detection in MSR dataset. First row shows the true detection of (a) boxing, (b) clapping and (c) waving action in MSR. Next row presents some failure cases. (d) Only waving action is detected although boxing is present. Similarly, in (e) only boxing action is detected.

used. We obtain so far the best result on this dataset by achieving $\sim 16\%$ higher detection score compared to the state-of-the-art. To show the effectiveness of our method, similar test on small scale benchmark dataset is also performed with state-of-the-art result. We gain $\sim 10\%$ more in detection score than the current state-of-the-art. More number of tests on the large scale videos would be one of the next steps. Usage of more motion features and improving the vitrification action classifier by adding pyramid level and boosting would also be an interesting area to explore.

References

[1] J. Aggarwal, M. Ryoo, Human activity analysis: A review, ACM Computing Surveys 43 (3) (2011) 1–43.

[2] T. Moeslund, A. Hilton, V. Krüger, A survey of advances in vision-based

human motion capture and analysis, Computer Vision and Image Understanding 8 (3) (2006) 231–268.

[3] R. Poppe, A survey on vision-based human action recognition, Image Vision Computing 28 (6) (2010) 976–990.

[4] P. Turaga, R. Chellappa, V. Subrahmanian, O. Udrea, Machine recognition of human activities: A survey, Circuits and Systems for Video Technology, IEEE Transactions on 18 (11) (2008) 1473–1488.

[5] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.-C. Chen, J. T. Lee, S. Mukherjee, J. K. Aggarwal, H. Lee, L. Davis, E. Swears, X. Wang, Q. Ji, K. Reddy, M. Shah, C. Vondrick, H. Pirsiavash, D. Ramanan, J. Yuen, A. Torralba, B. Song, A. Fong, A. Roy-Chowdhury, M. Desai, A large-scale benchmark dataset for event recognition in surveillance video, in: CVPR'11: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2011.

[6] J. Yuan, Z. Liu, Y. Wu, Discriminative subvolume search for efficient action detection, in: CVPR'09: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009.

[7] G. Yu, J. Yuan, Z. Liu, Unsupervised random forest indexing for fast action search, in: CVPR'11: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 865–872.

[8] C. Stauffer, W. E. L. Grimson, Adaptive background mixture models for real-time tracking, in: CVPR'99: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1999, pp. 2246–2252.

[9] B. Chakraborty, M. H. and. T.B. Moeslund, J. Gonzáles, A selective spatio-temporal interest point detector for human action recognition in complex scenes, in: ICCV'11: Proceedings of the Intrnatinal Conference on Computer Vision, 2011.

[10] P. Dollár, V. Rabaud, G. Cottrell, S. Belongie, Behavior recognition via sparse spatio-temporal features, in: VSPETS'05: Proceedings of the 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005, pp. 65–72.

[11] I. Laptev, On space-time interest points, International Journal of Computer Vision 64 (2/3) (2005) 107–123.

[12] J. Liu, J. Luo, M. Shah, Recognizing realistic actions from videos "in the wild", in: CVPR, 2009.

[13] C. Schuldt, I. Laptev, B. Caputo, Recognizing human actions: a local svm approach, in: ICPR'04: Proceedings of the International Conference on Pattern Recognition, Cambridge, UK, 2004, pp. 32–36.

[14] R. Chaudhry, A. Ravichandran, G. D. Hager, R. Vidal, Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions, in: CVPR'09: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1932–1939.

[15] N. Buch, J. Orwell, S. A. Velastin, 3d extended histogram of oriented gradients (3dhog) for classification of road users in urban scenes, in: BMVC'09: Proceedings of the British Machine Vision Conference, 2009.

[16] G. Willems, T. Tuytelaars, L. Gool, An efficient dense and scale-invariant spatio-temporal interest point detector, in: ECCV'08: Proceedings of the 10th European Conference on Computer Vision: Part II, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 650–663.

[17] J. Sivic, A. Zisserman, Video google: A text retrieval approach to object matching in videos, in: ICCV'03: Proceedings of the International Conference on Computer Vision, 2003, pp. 1470–1477.

[18] J. Niebles, H. Wang, L. Fei-Fei, Unsupervised learning of human actions categories using spatial-temporal words, International Journal of Computer Vision 79 (3) (2008) 299–318.

[19] G. Csurka, C. Dance, L. Fan, J. Willamowski, C. Bary, Visual categorization with bags of keypoints, in: Workshop on Statistical Learning in Computer Vision, 2004, pp. 1–22.

[20] R. Duda, P. Hurt, Use of hough transformation to detect lines and curves in pictures, Communications of the ACM 15 (1) (1972) 11–15.

[21] D. Ballard, Generalizing the hough transform to detect arbitrary shapes, Patteren Recognition 13 (2) (1981) 111–122.

[22] J. Gall, A. Yao, N. Razavi, L. J. V. Gool, V. S. Lempitsky, Hough forests for object detection, tracking, and action recognition, IEEE Transaction on Pattern Analysis and Machine Intellegence 33 (11) (2011) 2188–2202.

[23] B. Leibe, A. Leonardis, B. Schiele, Robust object detection with interleaved categorization and segmentation, International Journal of Computer Vision 77 (1-3) (2008) 259–289.

[24] J. Leiblt, C. Schimd, K. Schertler, View-point independent object class detection using 3d feature maps, in: CVPR'08: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2008.

[25] S. Maji, J. Malik, Object detection using a max-margin hough transform, in: CVPR'09: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009.

[26] B. Ommer, J. Malik, Multi-scale object detection by clustering lines, in: ICCV'09: Proceedings of the International Conference on Computer Vision, 2011.

[27] A. Opelt, A. Pinz, A. Zisserman, Learning an alphabet of shape and appearence for multi-class object detection, International Jounral of Computer Vision 80 (1) (2009) 16–44.

[28] S. Ali, A. Basharat, M. Shah, Chaotic invariants for human action recognition, in: ICCV'07: Proceedings of the IEEE 11th International Conference on Computer Vision, 2007, pp. 1–8.

[29] N. Nguyen, D. Phung, S. Venkatesh, H. Bui, Learning and detecting activities from movement trajectories using the hierarchical hidden markov model, in: CVPR'05: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2005.

[30] A. Galata, N. Johnson, D. Hogg, Learning variable-length markov models of behavior, Computer Vision and Image Undersanding 81 (3) (2001) 398–413.

[31] P. Scovanner, S. Ali, M. Shah, A 3-dimensional sift descriptor and its application to action recognition, in: ICM'07: Proceedings of the International Conference on Multimedia, ACM, New York, NY, USA, 2007, pp. 357–360.

[32] M. Bregonzio, J. Li, S. Gong, T. Xiang, Discriminative topics modelling for action feature selection and recognition, in: BMVC'10: Proceedings of the British Machine Vision Conference, 2010.

[33] A. Kläser, M. Marszałek, C. Schmid, A spatio-temporal descriptor based on 3d-gradients, in: BMVC'08: Proceeding of the British Machine Vision Conference, 2008, pp. 995–1004.

[34] P. Natarajan, R. Nevatia, View and scale invariant action recognition using multi-view shape-flow models, in: CVPR'08: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2008.

[35] S. Vitaladevuni, V. Kellokumpu, L. Davis, Action recognition using ballistic dynamics, in: CVPR'08: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2008.

[36] A. Yilmaz, M. Shah, Actions as objects: a novel action representation, in: CVPR'05: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2005.

[37] J. K. Aggarwal, N. Nandhakumar, On the computation of motion from sequences of images - a review, Proceedings of the IEEE 76 (8) (1988) 917–935.

[38] L. Zappella, X. Lladó, J. Salvi, Motion segmentation: a review, in: CCIA'08: Proceedings of the 11th International Conference of the Catalan Association for Artificial Intelligence, 2008, pp. 398–407.

[39] H. Bilen, V. Namboodiri, L. Gool, Action recognition: A region based approach, in: WACV'11: Proceedings of the IEEE Workshop on Applications of Computer Vision, 2011, pp. 294–300.

[40] M. Ullah, S. Parizi, I. Laptev, Improving bag-of-features action recognition with non-local cues, in: BMVC'10: Proceedings of the British Machine Vision Conference, 2010.

[41] A. Efros, A. Berg, G. Mori, J. Malik, Recognizing action at a distance, in: ICCV'03: Proceedings of the Ninth IEEE International Conference on Computer Vision, IEEE Computer Society, Washington, DC, USA, 2003.

[42] Y. Ke, R. Sukthankar, M. Hebert, Event detection in crowed videos, in: ICCV'05: Proceedings of the IEEE International Conference on Computer Vision, 2007.

[43] M. Rodriguez, J. Ahmed, M. Shah, Action mach: A spatio-temporal maximum average correlation height filter for action recognition, in: CVPR'08: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2008.

[44] E. Schechtman, M. Irani, Space-time behaviour based correlation, in: CVPR'05: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2005.

[45] J. Davis, A. Bobick, The representation and recognition of action using temporal templates, in: CVPR '97: Proceedings of the IEEE Computer Vision and Pattern Recognition, 1997, pp. 928–934.

[46] L. Gorelick, M. Blank, E. Shechtman, M. Irani, R. Basri, Actions as space-time shapes, Pattern Analysis and Machine Intelligence, IEEE Transactions 29 (2007) 2247–2253.

[47] Y. Hu, L. Cao, F. Lv, S. Yan, T. Huang, Action detection in complex scenes with spatial and temporal ambiguities, in: ICCV'09: Proceedings of the IEEE International Conference on Computer Vision, 2009.

[48] C. Lampert, M. Blaschko, T. Hofmann, Beyond sliding windows: Object localization by efficient sub-window search, in: CVPR'08: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2008.

[49] L. Cao, Z. Liu, T. Huang, Cross-dataset action detection, in: CVPR'10: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, 2010.

[50] J. Gall, V. Lempitsky, Class-specific hough forests for object detection, in: CVPR'09: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009.

[51] A. Yao, J. Gall, L. V. Gool, A hough transform-based voting framework for action recognition, in: CVPR'10: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2010.

[52] M. Dikmen, Surveillance event detection, in: In TRECVID Video Evaluation Workshop, 2008.

[53] H. Samet, M. Tamminen, Efficient component labeling of images of arbitrary dimension represented by linear bin-trees, IEEE Transactions on Pattern Analysis and Machine Intelligence 10 (4) (1988) 579–586.

[54] C. Cortes, V. Vapnik, Support -vector networks, Machine Learning 20 (3) (1995) 273–297.

[55] M. Grant, S. Boyd, cvx: Matlab software for disciplined convex programming (2008).

[56] D. Comaniciu, P. Meer, Mean shift: A robust approach towards feature space analysis, Pattern Analysis and Machine Intelligence, IEEE Transactions 24 (2002) 603–619.

[57] N. Slonim, N. Tishby, Agglomerative information bottleneck, in: NIPS'99: Proceedings of the Neural Information Processing Systems Foundation, 1999.