Selective spatio-temporal interest points

Bhaskar Chakraborty[a], Michael B. Holte[b], Thomas B. Moeslund[b], Jordi Gonzàlez[a]

[a]*Computer Vision Center (CVC) & Department of Computer Science (UAB), Edifici O, Campus UAB, 08193 Bellaterra, Spain*
[b]*Department of Architecture, Design and Media Technology, Aalborg Universtity, DK-9220 Aalborg, Denmark*

Abstract

Recent progress in the field of human action recognition points towards the use of Spatio-Temporal Interest Points (STIPs) for local descriptor-based recognition strategies. In this paper, we present a novel approach for robust and selective STIP detection, by applying surround suppression combined with local and temporal constraints. This new method is significantly different from existing STIP detection techniques and improves the performance by detecting more repeatable, stable and distinctive STIPs for human actors, while suppressing unwanted background STIPs. For action representation we use a bag-of-video words (BoV) model of local N-jet features to build a vocabulary of visual-words. To this end, we introduce a novel vocabulary building strategy by combining spatial pyramid and vocabulary compression techniques, resulting in improved performance and efficiency. Action class specific Support Vector Machine (SVM) classifiers are trained for categorization of human actions. A comprehensive set of experiments on popular benchmark datasets (KTH and Weizmann), more challenging datasets of complex scenes with background clutter and camera motion (CVC and CMU), movie and YouTube video clips (Hollywood 2 and YouTube), and complex scenes with multiple actors (MSR I and Multi-KTH), validates our approach and show state-of-the-art performance. Due to the unavailability of ground truth action annotation data for the Multi-KTH dataset, we introduce an actor specific spatio-temporal clustering of STIPs to address the problem of automatic action annotation of multiple simultaneous actors. Additionally, we perform cross-data action recognition by training on source datasets (KTH and Weizmann) and testing on completely different and more challenging target datasets (CVC, CMU, MSR I and Multi-KTH). This documents the robustness of our proposed approach in the realistic scenario, using separate training and test datasets, which in general has been a shortcoming in the performance evaluation of human action recognition techniques.

Keywords: action recognition, complex scenes, multiple actors, spatio-temporal interest points, local descriptors, bag-of-words, support vector machines

1. Introduction

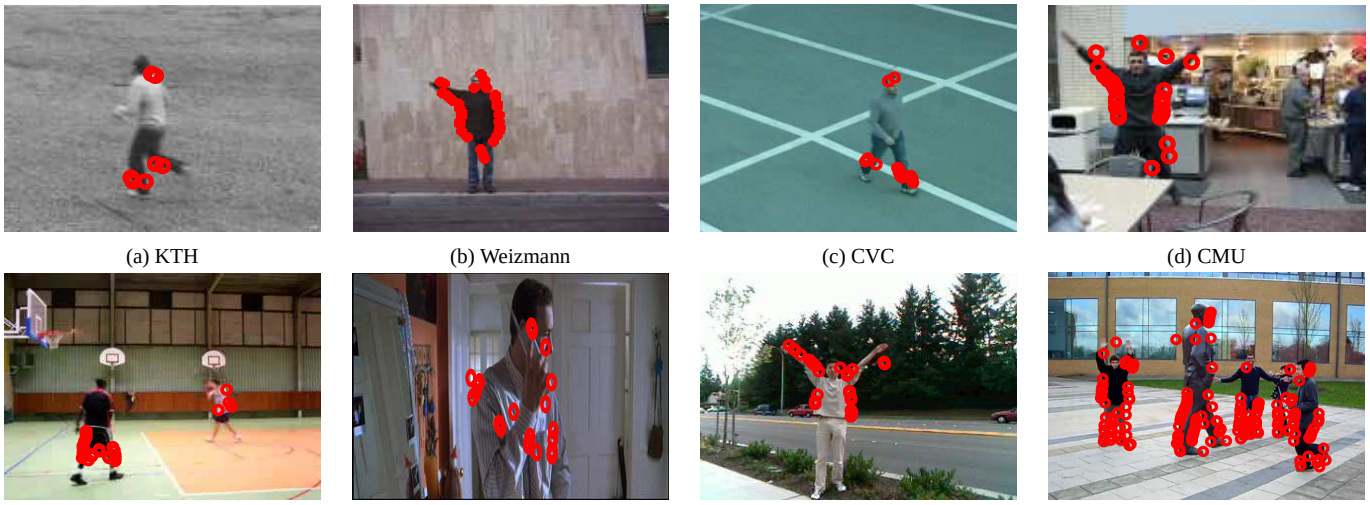
1.1. Human action recognition

In this paper, we address the task of human action recognition in complex scenes in diverse and realistic settings (background clutter, camera motion, occlusions and illumination variations). During the last decade action recognition has been an important topic in the "looking at people" domain [1–3]. A large number of methods for human action recognition have been proposed, stretching from human model and trajectory-based methods towards holistic and local descriptor-based methods.

Most of these previous approaches for human action recognition are constrained to well-controlled environments. Among the proposed action recognition techniques, one type of approach uses motion trajectories to represent actions and it requires target tracking [4, 5]. However, due to the difficulty in building robust object tracker only limited success has been achieved. Another type of approach uses sequences of silhouettes or body contours to model actions [1, 6] and it requires background subtraction. Boiman and Irani [7] extract densely sampled local video patches for detecting irregular actions in videos with simple background. Rodriguez et al. [8] designed a novel method to analyze the filtering responses of different actions. This approach has difficulties in aligning non-repetitive actions in complex scenes. Moreover, some researchers model the configuration of the human body and its evolution in the time domain [9, 10], and others solely perform action recognition from still images by computing pose primitives [11, 12].

The research trend in the field of action recognition has, recently, led to more robust techniques [13–22], which to some extent are applicable for action recognition in complex scenes. Action recognition in complex scenes is an extremely difficult task, due to several challenges, like background clutter, camera motion, occlusions and illumination variations. To address these challenges, several methods, like tree-based template matching [14], tensor canonical correlation [15], prototype based action matching [16], a hierarchical approach [18], incremental discriminant analysis of canonical correlation [20], latent pose estimation [21] and generalized Hough transform [22] have been proposed. Most of these methods are very complex and require preprocessing, like segmentation, tree data structure building, target tracking, background subtraction or a human

Email addresses: bhaskar@cvc.uab.es (Bhaskar Chakraborty), mbh@create.aau.dk (Michael B. Holte), tbm@create.aau.dk (Thomas B. Moeslund), jordi.gonzalez@cvc.uab.cat (Jordi Gonzàlez)



(a) KTH (b) Weizmann (c) CVC (d) CMU

(a) YouTube (b) Hollywood 2 (c) MSR I (d) Multi-KTH

Figure 1: Example images with superimposed STIPs from the eight action datasets applied for evaluation of our approach: KTH, Weizmann, CVC, CMU, YouTube, Hollywood 2, MSR I and Multi-KTH. The examples give an indication of the described challenges and differences in the datasets: simple scenes (KTH and Weizmann), semi-complex (CVC), and scenes of high complexity (CMU, YouTube, Hollywood 2, MSR I and Multi-KTH).

body model. Other methods [23–44] for action recognition in complex scenes, which demand less or no preprocessing, apply STIP detectors and local descriptors to characterize and encode the video data, and thereby perform action classification.

1.2. Spatio-temporal interest points

The extraction of appropriate features is critical to action recognition. Ideally, visual features are able to handle the following challenges for robust performance: (i) scale, rotation and viewpoint variations of the camera, (ii) performance speed variations for different people, (iii) different anthropometry of the actors and their movement style variations, and (iv) cluttered backgrounds and camera motion. The ultimate goal is to be able to perform reliable action recognition applicable for video indexing and search, intelligent human computer interaction, video surveillance, automatic activity analysis and behavior understanding. Recently, the use of STIPs has received increasing interest for local descriptor-based action recognition strategies. STIP-based methods avoid the temporal alignment problem, are exceptionally invariant to geometric transformations, and therefore distorted less by changes in scale, rotation and viewpoint than image data. Features are locally detected, thus inherently robust to occlusion and do not suffer from conventional figure-ground segmentation problems (imprecise segmentation, object splitting and merging etc.). Additionally, partial robustness to illumination variations and background clutter are incorporated.

Laptev and Lindeberg first proposed STIPs for action recognition [45], by introducing a space-time extension of the popular Harris detector [46]. They detect regions having high intensity variation in both space and time as spatio-temporal corners. The STIP detector of [45] usually suffers from sparse STIP detection. Later several other methods for detecting STIPs have been reported [47–51]. Dollar et al. [47] improved the sparse STIP detector by applying temporal Gabor filters and select regions of high responses. Dense and scale-invariant spatio-temporal interest points were proposed by Willems et al. [50], as a spatio-temporal extension of the Hessian saliency measure, previously applied for object detection [52, 53]. Instead of applying local information for STIP detection Wong et al. [51] propose a global information-based approach. They use global structural information of moving points and select STIPs according to their probability of belonging to the relevant motion. Although promising results have been reported, these methods are quite vulnerable to camera motion and cluttered background, since they detect interest points directly in a spatio-temporal space.

Hence, STIP-based methods have some shortcomings. First of all, (i) STIPs focus on local spatio-temporal information instead of global motion, thus the detection of STIPs on human actors in complex scenes might fall on cluttered backgrounds, especially if the camera is not fixed. Secondly, (ii) the stability of STIPs varies due to the local properties of the detector, and therefore some STIPs can be unstable and imprecise, as a result they have low repeatability or the local descriptors can become ambiguous. Thirdly, (iii) redundancy can occur in the local descriptors extracted from the surrounding image region of two adjacent STIPs. According to Schmid et al. [54] robust interest points should have high repeatability (geometric stability) and information content (distinctiveness of features). Furthermore, Turcot and Lowe [55] investigate and report that it is better to select a small subset of useful features for recognition problems, than a larger set of unreliable features which represent irrelevant clutter. We address these three shortcomings, by first (i) detecting Spatial Interest Points (SIPs), then (ii) suppressing unwanted background points, and finally (iii) imposeing local and (iv) temporal constraints, achieving a set of selective STIPs which are more robust to these challenges.

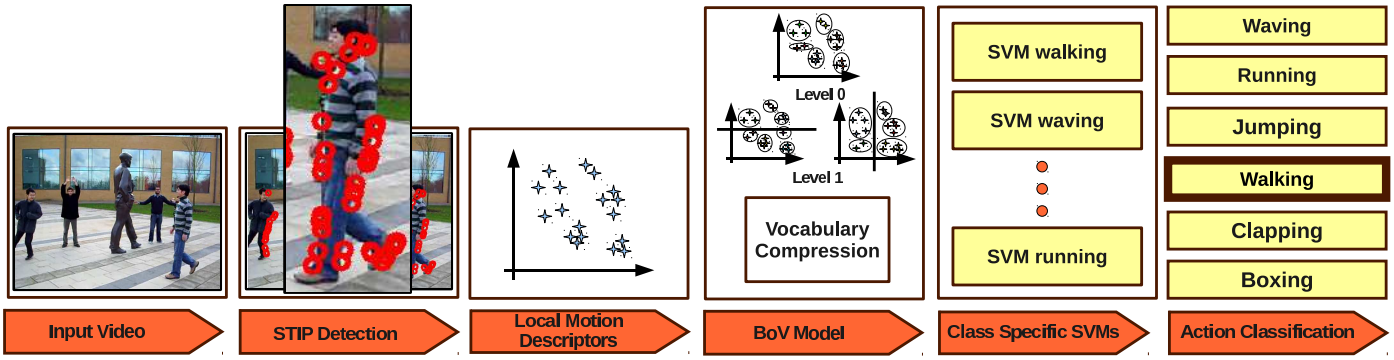


Figure 2: A schematic overview of the system structure and data flow pipeline of our approach.

1.3. Local descriptors

Several local descriptors have been proposed in the past few years [30, 47, 50, 56–59]. Local feature descriptors extract shape and motion in the neighborhoods of selected STIPs using image measurements, such as spatial or spatio-temporal image gradients or optical flow. Laptev et al. [30] introduced a combined descriptor to characterize local motion and appearance by computing histograms of spatial gradient (HOG) and optic flow (HOF) accumulated in space-time neighborhoods of detected interest points. Willems et al. [50] proposed the Extended SURF (ESURF) descriptor, which extends the image SURF descriptor [60] to videos. The authors divide 3D patches into cells, where each cell is represented by a vector of weighted sums of uniformly sampled responses of the Haar-wavelets along the three axes. Dollar et al. [47] proposed a descriptor along with their detector. The authors concatenate the gradients computed for each pixel in the neighborhood into a single vector and apply Principal Component Analysis (PCA) to project the feature vector onto a low dimensional space. Compared to the HOG-HOF descriptor proposed by Laptev et al. [30], it does not distinguish the appearance and motion features. The 3D-SIFT descriptor was developed by Scovanner et al. [59]. This descriptor is similar to the Scale Invariant Feature Transformation (SIFT) descriptor [61], except that it is extended to video sequences by computing the gradient direction for each pixel spatio-temporally in three-dimensions. Another extension of the popular SIFT descriptor was proposed by Kläser et al. [56]. It is based on histograms of 3D gradient orientations, where gradients are computed using an integral video representation. Another popular descriptor is the N-jets [57, 62]. An N-jet is the set of partial derivatives of a function up to order N, and is usually computed from a scale-space representation. The N-jets is an inherently strong local motion descriptor, where the two first levels implicitly represent velocity and acceleration.

1.4. Vocabulary building strategies

Bag-of-video words (BoV) models have become popular for generic action recognition [32, 33, 45, 47, 51, 63], whereas other techniques based on co-occurrence of STIP based motion features are also used [64]. The basic BoV model computes and quantizes the feature vectors, extracted at the detected STIPs in the video, into video-words. Finally, the entire video sequence is represented by a statistical distribution of those video-words. For classification, discriminative learning models such as SVM [47] and generative models, e.g. pLSA [51], have achieved excellent performance for action recognition. Since the BoV model does not provide a spatio-temporal distribution of features, the spatial correlogram and spatio-temporal pyramid matching are applied [33, 34] to capture the spatio-temporal relationship between local features. Additionally, vocabulary compression techniques are used to reduce the final feature space [32, 33]. We introduce a novel vocabulary building strategy by first applying a spatial pyramid and then compress the vocabulary at each pyramid level, achieving a compact and efficient pyramid representation of actions. This is different from [33], where first a vocabulary is computed, then it is compressed, and finally a spatial correlogram and a spatio-temporal pyramid are applied.

1.5. Complex scenes

While reliable human action recognition in simple scenes (KTH [65] and Weizmann [66]) has been achieved [15, 16, 20, 23, 26, 28], the task remains unsolved for complex scenes. These datasets have been recorded in well-controlled environments with clean or simple background, controlled lighting conditions, and no camera motion nor occlusions. In contrast, Real world human actions are often recorded in scenes of high complexity, with cluttered background, illumination variations, camera motion and occluded bodies. Hence, these datasets do not correspond very well to real world scenarios. The mentioned properties make action recognition in complex scenes much more challenging. New datasets for the purpose of evaluation of action recognition algorithms in complex and semi-complex scenes have therefore been produced (CMU [67], CVC [68], YouTube [32], Hollywood 2 [34], MSR I [63] and Multi-KTH [41]). We utilize all these datasets for evaluation of our approach (see Figure 1).

1.6. Cross-data evaluation

Conventional approaches usually build a classifier from labeled examples and assume the test samples are generated from the same distribution, which is rarely the case in realistic scenarios. In contrast, cross-data evaluation is highly necessary for

commercial systems, where the classifier is trained on a specific dataset during a learning phase and then set up for operation in the field. Additionally, it also prevents the algorithm to benefit from the internal data correlation during the evaluation. Cross-data evaluation is more challenging, since the two dataset have usually been recorded in two different occasions. Only a few authors have recently reported cross-data evaluation [23, 26, 41]. The problem is related to transfer learning known from machine learning, which attempts to develop methods to transfer knowledge learned in one or more source tasks and use it to improve learning in a related target task [69, 70]. We conduct a comprehensive set of cross-data experiments to carry out a more realistic evaluation of our approach.

1.7. Our approach and contributions

In this work we follow the recent progress and employ a STIP and local descriptor-based recognition strategy. A schematic overview of our approach is outlined in Figure 2. (1) We introduce a novel approach for selective STIP detection, by applying surround suppression combined with local and temporal constraints, achieving robustness to camera motion and background clutter. For action representation we use a BoV model of local N-jet features, extracted at the detected STIPs, to build a vocabulary of visual-words. (2) To this end, we introduce a novel vocabulary building strategy by combining (i) a pyramid structure to capture spatial information, and (ii) vocabulary compression to reduce the dimensionality of the feature space, resulting in improved performance and efficiency. Action class-specific SVM classifiers are trained and applied for categorization of natural human actions. (3) We evaluate our approach on both popular benchmark datasets (KTH and Weizmann), more challenging datasets (CVC, CMU), movie and YouTube video clips (Hollywood 2 and YouTube) and perform an exhaustive cross-data evaluation, trained on source dataset (KTH and Weizmann) and tested on more challenging target datasets (CVC, CMU, MSR I and Multi-KTH). Due to the unavailability of ground truth action annotation data for the Multi-KTH dataset, we introduce an actor specific spatio-temporal clustering of STIPs to address the problem of automatic action annotation of multiple simultaneous actors. To observe the performance our automatic STIP clustering-based annotation, we manually annotate the ground truth actions and compare the action recognition accuracies. Finally, we compare our approach to the most popular action recognition techniques and show beyond state-of-the-art performance.

1.8. Paper structure

The remainder of the paper is organized as follows. We describe our STIP detector and local descriptor-based action representation in section 2. Section 3 outlines our vocabulary building strategy and narrates the applied classifier for action categorization. Experimental results and comparisons, along with our technique for spatio-temporal clustering of STIPs for automatic action annotation of Multi-KTH, are reported in section 4, followed up by concluding remarks in section 5.

2. Selective spatio-temporal interest points

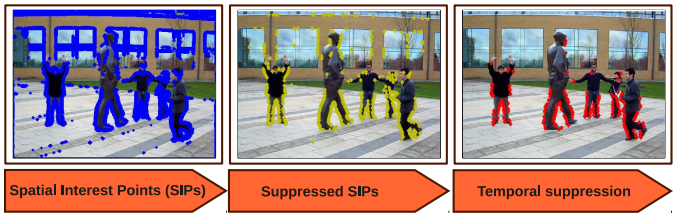


Figure 3: A schematic overview of the spatio-temporal interest point detection module and the associated data flow pipeline.

2.1. Detection of spatial interest points.

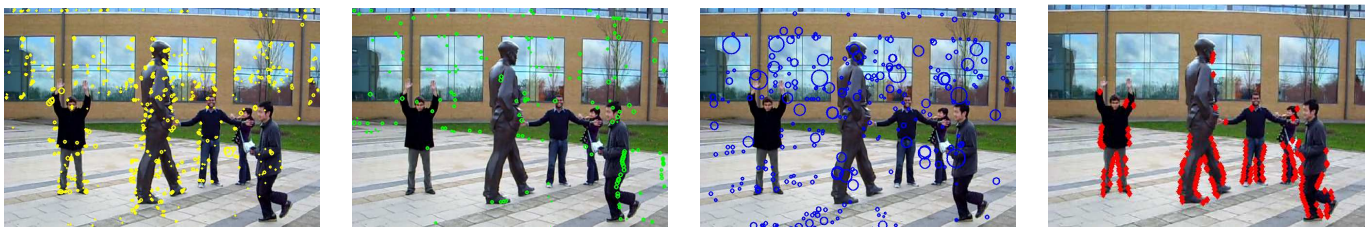
Existing STIP detectors [45, 47, 48, 50, 51] are vulnerable to camera motion and moving background in videos, and therefore detect unwanted STIPs in the background (see Figure 4). Cao et al. [23] have recently reported, that of all the STIPs detected by Laptev's STIP detector [45], only about 18% correspond to the three actions performed by the actors in the MSR I dataset [63], while the rest of the STIPs (82%) belong to the background. To overcome this problem, we first detect the spatial interest points (SIPs), then perform background suppression and impose local and temporal constraints (see Figure 3). We apply the basic Harris corner detector [46] and compute the first set of interest points with corner strength C_σ, where σ is the spatial scale. Apart from the detected SIPs on the human actors, the obtained spatial corners C_σ contain a significant amount of unwanted background SIPs (see Figure 3).

2.2. Suppressing background interest points

The main idea of our spatial interest point suppression originates in the fact that most corner points detected in the background texture or on non-human objects follow some particular geometric pattern, while those on humans do not have this property. For suppression we use a surround suppression mask (SSM) for each interest point, taking the current point under evaluation as the center of the mask. We then estimate the influence of all surrounding points of the mask on the central point, and accordingly, a suppression decision is taken. The idea is motivated by [71], where surround suppression is used for texture edges to improve object contour and boundary detection in natural scenes. The similar concept of surround suppression based on center surround saliency measure is been adopted in tracking [72], spatio-temporal saliency algorithm [73] and detection of suspicious coincidences in visual recognition [74]. We implement surround suppression by computing an inhibition term for each point of C_σ. For this purpose we introduce a gradient weighting factor $\triangle_{\Theta,\sigma}(x, y, x-u, y-v)$, which is defined as:

$$\triangle_{\Theta,\sigma}(x, y, x-u, y-v) = \\ |\cos(\Theta_\sigma(x, y) - \Theta_\sigma(x-u, y-v))| \quad (1)$$

where $\Theta_\sigma(x, y)$ and $\Theta_\sigma(x-u, y-v)$ are the gradients at point (x, y) and $(x-u, y-v)$, respectively; u and v define the horizontal and vertical range of the SSM. If the gradient orientations at



(a) (b) (c) (d)

Figure 4: STIP detection results for the Multi-KTH dataset. (a) Laptev et al. [45], (b) Dollar et al. [47] (c) Willems et al. [50] and (d) Our approach. Due to background clutter and camera motion (a), (b) and (c) detect quite a large number of STIPs in the background compared to our approach.

point (x, y) and $(x-u, y-v)$ are identical, the weighting factor attains its maximum ($\triangle_{\Theta,\sigma} = 1$), while the value of the factor decreases with the angle difference and reaches a minimum ($\triangle_{\Theta,\sigma} = 0$), when the two gradient orientations are orthogonal. Hence, the surrounding interest points which have the same orientation, as that of (x, y), will have a maximal inhibitory effect.

For each interest point $C_\sigma(x, y)$, we define a suppression term $t_\sigma(x, y)$ as the weighted sum of gradient weights in the suppression surround of that point:

$$t_\sigma(x,y) = \int\int_{\Omega} C_\sigma(x-u, y-v) \times \triangle_{\Theta,\sigma}(x, y, x-u, y-u) du dv \quad (2)$$

where Ω is the image coordinate domain. We now introduce an operator $C_{\alpha,\sigma}(x, y)$, which takes its inputs: the corner magnitude $C_\sigma(x, y)$ and the suppression term $t_\sigma(x, y)$:

$$C_{\alpha,\sigma}(x,y) = H(C_\sigma(x,y) - \alpha t_\sigma(x,y)) \quad (3)$$

where $H(z) = z$ when $z \geq 0$ and *zero* for negative z values. The factor α controls the strength of the surround suppression. If no interest points have been detected in the surrounding texture of a given point, the response of the operator retains the original corner magnitude $C_\sigma(x, y)$. However, if a large number of interest points are detected in the surrounding background texture, the suppression term $t_\sigma(x, y)$ will be higher, resulting in a suppression of the current interest point under evaluation.

2.3. Imposing local constraints

We select a final set of interest points from the surround suppression responses $C_{\alpha,\sigma}$ (Equation 3) by applying non-maxima suppression, similar to Grigorescu et al.'s method for suppressing gradients [71]. Non-maxima suppression thins the areas in which $C_{\alpha,\sigma}$ is non-zero to one-pixel wide candidate contours as follows: for each position (x, y), the two responses $C_{\alpha,\sigma}(x', y')$ and $C_{\alpha,\sigma}(x'', y'')$ in adjacent positions (x', y') and (x'', y''), which are intersection points of a line passing through (x, y) with orientation $\Theta_\sigma(x, y)$ and a square defined by the diagonal points of an 8-neighbourhood, are computed by linear interpolation (see Figure 5). A point is kept, if the response $C_{\alpha,\sigma}(x, y)$ is greater than that of the two adjacent points, i.e., it is a local maximum of the neighbourhood. Otherwise its value is set to zero. Figure 6 shows an example of the performance of our inhibitive SIP detector. As can be seen in Figure 6.b some background SIPs might remain in $C_{\alpha,\sigma}$. However, these static SIPs can be removed by imposing temporal constraints.

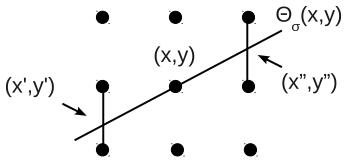


Figure 5: Responses at position (x', y') and (x'', y'') along the line passing through (x, y) [71]. Non-maxima suppression retains the value in the central position (x, y), if it is greater than the values at (x', y') and (x'', y'').

2.4. Scale adaptive SIPs

Scale selection plays an important role in the detection of spatial interest points. Automatic scale selection can be achieved based on the maximization of normalized derivatives expressed over scale, or by the behavior of entropy or error measures evaluated over scale [53, 75]. Instead of applying an automatic scale selection, as in [76], we apply a multi-scale approach [30] and compute suppressed SIPs in *five* different scales $S_\sigma = \{\frac{\sigma}{4}, \frac{\sigma}{2}, \sigma, 2\sigma, 4\sigma\}$. We follow the idea of scale selection presented by Lindeberg [53] to keep the best set of SIPs obtained for each scale. The best scales are selected by maximizing the normalized differential invariant,

$$\tilde{\kappa}_{norm} = \sigma_0^{2\gamma} L_y L_{xx}. \quad (4)$$

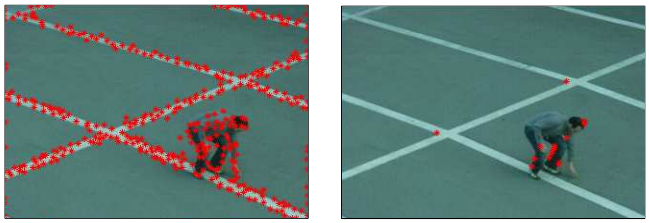
where $L = g(\cdot; \sigma_0, \tau_0) \otimes I$, i.e. the image I is convoluted with the Gaussian kernel g; L_y is the first order y derivative and L_{xx} is the second order x derivative of L. Lindeberg [53] report that $\gamma = \frac{7}{8}$ performs well in practice to achieve the maximum value of $(\tilde{\kappa}_{norm})^2$ for spatial interest point detected at multiple scales. After computing the suppressed SIPs in the scale-space in S_σ, we apply this scale selection procedure based on the normalized differential invariant (Equation 4), and keep the n best SIPs as our final set of suppressed SIPs.

2.5. Imposing temporal constraints

After obtaining the final set of spatial interest points we impose temporal constraints to neglect static SIPs. We consider two consecutive frames at a time and remove the common interest points, since static interest points do not contribute any motion information:

$$\mathcal{P}^T_{\alpha,\sigma} = C^T_{\alpha,\sigma} \backslash \{C^T_{\alpha,\sigma} \cap C^{T-1}_{\alpha,\sigma}\} \quad (5)$$

where $C^T_{\alpha,\sigma}$ is the set of interest points in the T^{th} frame. To avoid the camera motion we have used an interest point matching algorithm along with a temporal Gabor filter response to



(a) (b)

Figure 6: Performance of our SIP detector with $\alpha = 1.5$. Detected SIPs (a) before suppression and (b) after suppression.

remove the static interest points (Equation 5). The remaining points are the final set of detected STIPs, which are used to extract local features. The pseudo code for the full STIP detection is described in Algorithm 1. Parallelization can be adopted for speed optimization by parallel computation of the **for** loops in each algorithm (Algorithm 1,3,2,4 and 5).

Algorithm 1 STIP detection from an image stack.

Require: An image stack (H × W × N): iS;
(contains all the video frames)
Array containing spatial scales: sA;
Alpha: α;
Mask: m;
Ensure: Detected STIPs: $stip$

$sip = \{\}$; $stip = \{\}$;
$N = size(iS, 3)$; (Total no. of frames)
for $i = 1 \rightarrow N$ **do**
  for $j = 1 \rightarrow size(sA)$ **do**
    $sip \leftarrow sip \cup \{SCD(iS(:,:,i), sA(j), \alpha, m), sA(j)\}$;
  end for
  $stip \leftarrow stip \cup blobDetector(iS(:,:,i), sip)$;
end for
$stip = temporalConstraint(iS, stip)$;
Return($stip$);

Algorithm 2 SCD: Selective STIP detection.

Require: An image (H × W): $image$;
Spatial scale: σ;
Alpha: α;
Mask: $mask$;
Ensure: Detected selective spatial interest points: sip

$cp = harrisCorner(image, \sigma)$;
$cornerPoints = find(cp > 0)$;
$cp = cp(cornerPoints)$;
$\Theta = gradient(image)$;
$sip = \{\}$;
for Each point $(x, y, \sigma) \in cornerPoints$ **do**
  $\triangle_{\Theta_{mask}} = |\cos(\Theta_{mask} - \Theta_{mask_{(x,y)}})|$;
  $t(x, y) = cp_{mask} \otimes \triangle_{\Theta_{mask}}$;
  $cp(x, y) = H(cp_{(x,y)} - \alpha t_{(x,y)})$;
  $(x', y') = round(line(x, x + 1, y, \Theta(x, y)))$;
  $(x'', y'') = round(line(x, x - 1, y, \Theta(x, y)))$;
  if $(cp(x, y) > cp(x', y')) \wedge (cp(x, y) > cp(x'', y''))$ **then**
    $sip \leftarrow sip \cup (x, y, \sigma)$;
  end if
end for
Return(sip);

Algorithm 3 blobDetector: Corner strength detection using Gaussian blob.

Require: An image (H × W): im;
Corner points: $corners$;
Ensure: Detected selective spatial interest points based on Gaussian blob strength: $cornerPoints$

$cornerPoints = \{\}$;
for Each point $(X, Y, \sigma) \in corners$ **do**
  $bS = \sigma^{1.75} * L_{y,im}(X, Y) * L_{xx,im}(X, Y)$;
  if $(bS > \tau)$ **then**
    $cornerPoints \leftarrow cornerPoints \cup (X, Y, \sigma)$;
  end if
end for
Return($cornerPoints$);

2.6. Local feature descriptors

We use local N-jet features [57] extracted at the detected STIPs. We extract N-jet features of order-2 in five different temporal scales. Consequently, we end up with a 10-dimensional feature vector,

$$\mathcal{F}_{norm}(g(\cdot; \sigma_0, \tau_0) \cdot I) = \{L, \sigma L_x, \sigma L_y, \ldots, \tau^2 L_{tt}\} \quad (6)$$

at locally adopted scale level (σ_0, τ_0) for the image sequence I; where $g(\cdot; \sigma_0, \tau_0)$ is the Gaussian kernel at spatio-temporal scale (σ_0, τ_0) and σ_0 is identical to the scale of the STIP detector; $L = g(\cdot; \sigma_0, \tau_0) \otimes I$, i.e. the image I is convoluted with the Gaussian kernel g; L_x is the first order x derivative and L_{xx} is the second order x derivative of L etc.

These features are computed with a fixed spatial scale σ_0 but with five different temporal scales $(\frac{\tau}{4}, \frac{\tau}{2}, \tau, 2\tau, 4\tau)$. We do not increase the order of N-jet, like Laptev et al. [62], since the two first levels represent velocity L_{xt} and acceleration L_{tt} information, while higher order spatial or temporal derivatives are sensitive to noise and do not bring significant additional motion information. The experimental results reported in section 4 document our feature selection by showing state-of-the-art performance.

3. Vocabulary building and classification

We apply a BoV model to learn the visual vocabularies of the extracted local motion features. We extend the idea of [32] by introducing pyramid levels in the feature space, but instead of applying a pyramid at feature level, as in [33], we apply it at STIP level. This makes the problem of grouping the local features much simpler yet robust, since our STIPs are detected in a selective and robust manner. Finally, we apply vocabulary compression, at each pyramid level, to reduce the dimensionality of the feature space (see Figure 7).

Algorithm 4 temporalConstraint: Imposed temporal constraint on the selected spatial corner points

Require: An image stack (H × W × N): iS;
Spatial corner points: cp;
Ensure: Detected STIPs: $stip$

for $i = 1 \rightarrow H$ **do**
  for $j = 1 \rightarrow W$ **do**
    $gabor(i, j, :) = gaborFiler1D(iS(i, j, :))$;
  end for
end for
for $i = N \rightarrow 2$ **do**
  $f_1 = iS(:, :, i)$; $f_2 = iS(:, :, i-1)$;
  $g_1 = gabor(:, :, i)$; $g_2 = gabor(:, :, i-1)$;
  $im_1 = iS(:, :, i)$; $im_2 = iS(:, :, i-1)$;
  $cp_{f_1} \leftarrow cp_{f_1} \backslash pointMatch(cp_{f_1}, cp_{f_2}, g_1, g_2, im_1, im_2)$;
end for
Return(cp)

Algorithm 5 pointMatch: Detect the set of matching corner points in two consecutive frames.

Require: Image frames: im_1, im_2;
Corner strengths: cp_1, cp_2;
Gabor strength: g_1, g_2;
Ensure: Detected matching STIPs: mS

$mP = \{\}$;
$cornerPoints_1 = find(cp_1 > 0)$;
$cornerPoints_2 = find(cp_2 > 0)$;
for Each point $(x_1, y_1, \sigma_1) \in cornerPoints_1$ **do**
  $H = \sigma_1$;
  for Each point $(x_2, y_2, \sigma_2) \in cornerPoints_2$ **do**
    $similarity = \frac{min(cp_1(x_1,y_1), cp_2(x_2,y_2))}{min(cp_1(x_1,y_1), cp_2(x_2,y_2))}$;
    $W = \sigma_2$;
    if $similarity > \tau_{sim}$ **then**
      $a_1 = cropRect(im_1, x_1, y_1, H, W)$;
      $a_2 = cropRect(im_2, x_2, y_2, H, W)$;
      $sC = crossCorrelation(a_1, a_2)$;
      if $(sC > \tau_{corr}) \wedge (g_1(x_1, y_1) > \tau_{gabor})$ **then**
        $mP \leftarrow mP \cup (x_1, y_1, \sigma_1)$;
      end if
    end if
  end for
end for
Return(mS);

3.1. Pyramid structure

Let I_T be the T^{th} frame of the image sequence I and $P^T_{\alpha,\sigma}$ (Equation 5) the set of detected STIPs in this frame. We then quantize this set of STIPs into q levels, $\mathcal{S} = \{s_0, s_1, \ldots, s_{q-1}\}$ [34]. For each of these levels, the STIPs are divided based on center of mass information. Accordingly, we group the motion features into different levels of the pyramid. The structure of our 2-level pyramid is illustrated in Figure 8. The horizontal division helps to capture the distinguishing characteristics of arm and leg-based actions, whereas the vertical division distinguishes the actions within each of these arm and leg-based action classes.

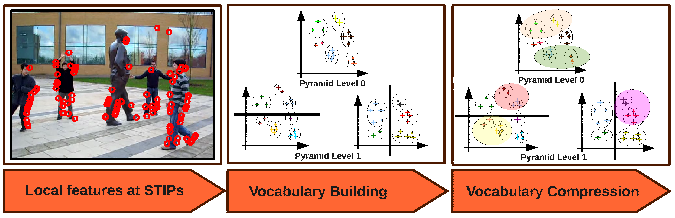


Figure 7: A schematic overview of the vocabulary building module and the associated data flow pipeline.

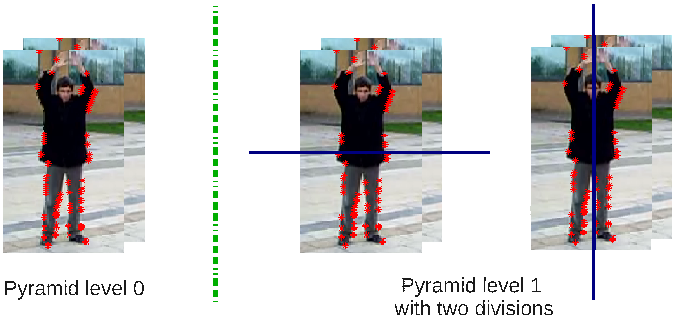


Figure 8: Spatial pyramid of level 2.

3.2. Vocabulary compression

After dividing the motion features into the described pyramid levels, we create initial vocabularies of a relatively large size (about 400 words). To reduce the final feature dimensionality, we use vocabulary compression, as in [32], but at each level of the pyramid to achieve a compact yet discriminative visual-word representation of actions.

Let A be a discrete random variable which takes the value of a set of action classes $A = \{a_1, a_2, \ldots, a_n\}$, and W_s be a random variable which range over the set of video-words $W_s = \{w_1, w_2, \ldots, w_m\}$ at pyramid level s. Then the information about A captured by W_s can be expressed by the Mutual Information (MI), $I(A, W_s)$. Now, let $\widehat{W_s} = \{\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_k\}$ for $k < m$, be the compressed video-word cluster of W_s. We can measure the loss of quality of the resulting compressed vocabulary $\widehat{W_s}$, as the loss of MI:

$$Q(\widehat{W_s}) = I(A, W_s) - I(A, \widehat{W_s}) \tag{7}$$

To find the optimal compression $\widehat{W_s}$ we use an Agglomerative Information Bottleneck (AIB) approach.

3.3. AIB compression

AIB [77] iteratively compresses the vocabulary W_s by merging the visual-words w_i and w_j which cause the smallest decrease in MI, $I(A, W_s)$. The algorithm can be summarized as follows:

- Initiate $\widehat{W_s} \equiv W_s$, i.e., by taking each video-word of W_s as a singleton cluster.

- Pair-wise distance computation: for every $\{w_i, w_j\} \in \widehat{W_s}$, $i < j$, the distance d_{ij} (which is a measure of MI) is computed:

$$d_{ij} = (p(w_i) + p(w_j)) \cdot JS_{\Pi}[p(a|w_i), p(a|w_j)] \quad (8)$$

where $JS_{\Pi}[p(a|w_i), p(a|w_j)]$ is the Jensen-Shannon divergence for a M class distribution, $p_i(x)$, each with a prior π_i, and is defined as:

$$JS_{\Pi}[p_1, p_2, \ldots, p_M] \equiv H[\sum_{i=1}^{M} \pi_i p_i(x)] - \sum_{i=1}^{M} \pi_i H[p_i(x)] \quad (9)$$

where $H[p(x)]$ is Shannon's entropy:

$$H[p(x)] = -\sum_{x} p(x) \log p(x) \quad (10)$$

- Merging: select the pair of video-words $\{w_\alpha, w_\beta\}$ for which the distance $d_{\alpha\beta}$ is minimum and merge them. Hence, we merge the video-words which result in the minimum MI loss by optimizing the global criterion in Equation 7.

AIB is a greedy algorithm in nature and optimizes the merging of only two word clusters at every step (local optimization). Hence, it optimizes the global criteria defined in Equation 7. We use the described vocabulary compression at each level of the pyramid per class, and obtain a final class-specific compact pyramid representation of video-words.

We use AIB for the vocabulary compression instead of Principal Component Analysis (PCA) based dimensionality reduction, since PCA is a linear model, whereas the relationship among the video words are highly non-linear in nature. Besides, PCA based dimensionality reduction will work on the first level cluster (k-means) of the bag-of-words model to reduce the final bag-of-words histogram dimensionality. Hence, it will not take inter and intra cluster similarities into account. Unlike PCA, the agglomerative information bottleneck (AIB) method presented in the article, is non-linear and it yields a set of compressed clusters from the first level clusters, such that the set of resulting compressed clusters maximally preserves the original information among them. Additionally, AIB based compression explores the mutual information present among video words and apply compression based on this information. Hence, in this case, AIB based compression is analytically more appropriate than PCA.

To empirically support our selection of AIB based compression, we have conducted experiments on the Weizmann dataset using PCA based dimensionality reduction. The obtained average accuracy is quite low (40% in the range of 30% – 70% compression) compared to the recognition rate of AIB (99% in the same range of compression), which documents that AIB is a far better choice.

3.4. Action classification

After compression of the video-words at each pyramid level we compute a histograms of the video-words, using the extracted local motion features, and concatenate them to a final feature set for SVM learning. We design a class specific χ-square kernel-based SVM, $\text{SVM}_{a_i}(k, h^{a_i}_{W_{a_i}})$ [78], where a_i is the i^{th} action class A, k is the SVM kernel and $h^{a_i}_{W_{a_i}}$ is the histogram of action class a_i, computed using the class-specific video-words W_{a_i}. For a test set a_{Test} we detect its action class:

$$i^*_{a_{Test}} = argmax_j \text{SVM}_{a_j}(k, h^{a_{Test}}_{W_{a_j}}), \forall a_j \in A \quad (11)$$

We conduct experiments using different SVM kernels, and observe that the χ-square and intersection kernel are the best perfoming SVM kernels for all the datasets. Hence, we apply the χ-square kernel for all our experiments on human action recognition in section 4. Table 1 shows the average recognition accuracy for the Weizmann dataset using a number of different SVM kernels.

Table 1: Average recognition accuracy for the Weizmann dataset using different SVM kernels. We have used a Polynomial kernel of degree 3.

SVM Kernel	Recognition rate (%)
χ-square	**99.50**
Intersection	97.78
Radial basis function	87.77
Polynomial	78.67
Linear	58.89

4. Experimental results

4.1. Human action datasets

To test our proposed approach for action recognition we conduct a comprehensive set of experiments using a number of publicly available human action datasets (see Figure 1), which are categorized as follows.

4.1.1. Single actor benchmark

To conduct benchmark testing we choose the two most popular human action datasets: KTH [65] and Weizmann [66]. Both of these datasets contain single actors and clean backgrounds. The KTH dataset consists of 6 different actions: *walking*, *jogging*, *running*, *boxing*, *clapping* and *waving*. These actions are performed in 4 different but well-controlled environments by 25 different actors, resulting in a total of 600 action instances. The Weizmann dataset contains 90 videos separated into 10 actions performed by 9 persons. The actions are: *bend*, *jumping-jacks*, *jump*, *jump-in-place*, *run*, *gallop-sideways*, *skip*, *walk*, *one-hand-waving* and *two-hands-waving*.

4.1.2. Single actor with complex background

In this category we choose the CVC action dataset [68] and the CMU action dataset [67]. The CVC dataset consists of 5 actors performing 7 actions: *walking*, *jogging*, *running* (with horizontal and vertical two-way paths), *hand-waving*, *two-hands-waving*, *jump-in-place* and *bending*. The dataset is rated "semi-complex" and is interesting, since it has a textured background. The CMU dataset is composed of 48 video sequences of five action classes: *jumping-jacks*, *pick-up*, *push-button*, *one-hand-waving* and *two-hands-waving*. The test data contains 110

videos (events) which are down-scaled to 160×120 in resolution. This dataset has been recorded by a hand-held camera with moving people and vehicles in the background, and is known to be very challenging.

4.1.3. Movie and YouTube video clips

To evaluate our approach in different challenging stettings, we conduct experiments on movie and YouTube video clips. Concretely, we use the Hollywood 2 human actions and scenes dataset [34] and the YouTube action dataset [32]. The Hollywood 2 dataset is composed of video clips extracted from 69 Hollywood movies, and contains 12 classes of human actions: *AnswerPhone*, *DriveCar*, *Eat*, *FightPerson*, *GetOutCar*, *HandShake*, *HugPerson*, *Kiss*, *Run*, *SitDown*, *SitUp* and *StandUp*. In total, there are 1707 action samples divided into a training set (823 sequences) and a test set (884 sequences), where train and test sequences are obtained from different movies. The dataset intends to provide a comprehensive benchmark for human action recognition in realistic and challenging settings. The YouTube dataset is a collection of 1168 complex and challenging YouTube videos of 11 human actions categories: *basketball shooting*, *volleyball spiking*, *trampoline jumping*, *soccer juggling*, *horseback riding*, *cycling*, *diving*, *swinging*, *golf swinging*, *tennis swinging* and *walking (with a dog)*. The dataset has the following properties: a mix of steady cameras and shaky cameras, cluttered background, low resolution, and variation in object scale, viewpoint and illumination. The first four actions are easily confused with jumping, the next two may have similar camera motion, and all the swing actions share some common motions. Some actions are also performed with objects such as a horse, bike or dog.

4.1.4. Multiple actors with complex background

We use two multiple actor datasets: the Microsoft research action dataset I (MSR I) [63] and the Multi-KTH dataset [41]. MSR I consists of 16 video sequences and a total of 63 actions: 14 *hand-clapping*, 24 *hand-waving* and 25 *boxing*, performed by 10 subjects. The sequences contain multiple types of action recorded in indoor and outdoor scenes with cluttered and moving backgrounds. Some sequences contain multiple actions performed by different people. Each video is of low resolution 320×240 with a frame rate of 15 frames per second, and their lengths are between 32 to 76 seconds. The Multi-KTH dataset is a more challenging version of the KTH dataset. It contains 5 (except *running*) of the 6 KTH-actions, which have been recorded by a hand-held camera, with multiple simultaneous actors, a significant amount of camera motion, scale changes and a more realistic cluttered background.

4.2. Automatic action annotation for Multi-KTH

When multiple actors appear simultaneously in a scene, it is necessary to group the detected STIPs into actor-specific clusters. An excellent example is the Multi-KTH dataset, where five actors are present in the scene. Based on this dataset we introduce a spatio-temporal clustering technique for actor-specific STIP grouping and evaluate its performance in section 4.8. This

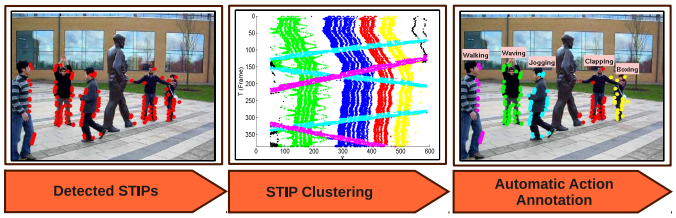


Figure 9: A schematic overview of the spatio-temporal clustering module and the associated data flow pipeline.

spatio-temporal clustering is only a part of Multi-KTH dataset for automatic annotation.

4.2.1. Actor-specific STIP clustering

The actions present in the Multi-KTH dataset can be divided into two main groups: the actions with moving actors, like *walking* and *jogging*, and the actions with static actors, like *boxing*, *waving* and *clapping*. These two different nature of actions can be analyzed in the 2D spatio-temporal XT-space (see Figure 10.b). The actor-specific STIP clustering exploits the 2D spatio-temporal XT-space and consist of two main steps:

i) detection of lines in the XT-space and cluster STIPs accordingly,
ii) after the first set of STIP clusters have been estimated, the associated STIPs are excluded and the resulting subset is clustered using morphological operations and a spatio-temporal distance measurement.

The surround suppression effect of our STIP detector, resulting in a low detection rate of unwanted background STIPs, facilitates STIP clustering in the XT-space. This will simply not be possible with a high number of background STIPs. Figure 9 illustrates the concept of the spatio-temporal clustering.

4.2.2. The spatio-temporal XT-space

A plot of the detected STIPs in 3D spatio-temporal XYT-space for the Multi-KTH sequence is shown in Figure 10.a. As can be seen, actor-specific clustering of the STIPs is non-trivial due to camera motion and occlusions. Hence, successful clustering cannot be accomplished by commonly used methods, e.g., *k*-means or Mean Shift clustering. Instead, we project the 3D spatio-temporal STIPs onto a 2D spatio-temporal XT-space, as shown in Figure 10.b, which reveals some interesting and useful patterns. The XT-space can be seen as the top-down view of the 3D spatio-temporal XYT-space (Figure 10.a), with the horizontal and vertical axes representing the X-position and the time T, respectively. Hence, the T-axis demonstrates the evolution of STIPs in time.

4.2.3. Detection of lines in XT-space

Actions like *walking*, *jogging* or *running* create lines in the XT-space. Hence, we detect line segments in XT-space to cluster STIPs detected for the actors. This is valid, since actors with a certain target destination move in a linear pattern for those actions. Hough transform [79] is applied for the detection of these linear patterns (i.e., line segments) and the candidates

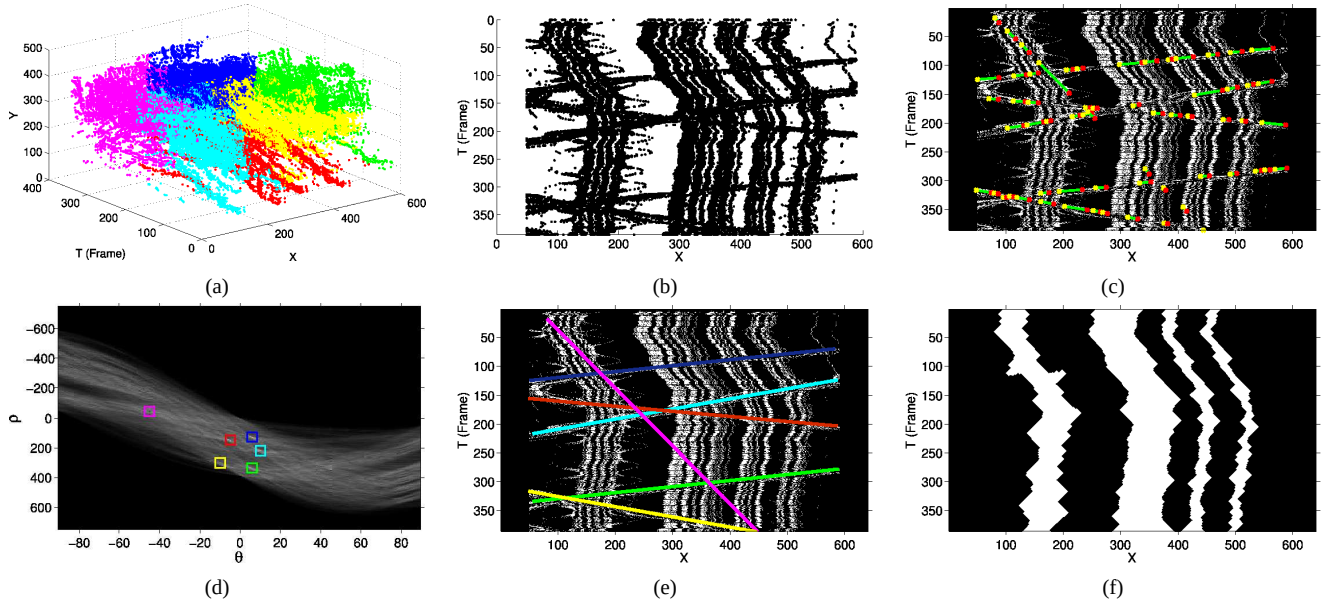


Figure 10: Plots of the detected STIPs for the Multi-KTH dataset, and detection of linear patterns in the XT-space. (a) *k*-means clustered STIPs in the 3*D* spatio-temporal XYT-space and (b) ungrouped STIPs in the 2*D* spatio-temporal XT-space; (c) line segments in XT-space caused by actions like walking, jogging or running; (d) candidates with high responses in the Hough space; (e) detected line segment using the Hough transfrom and (f) *blobs* obtained by morphological operations.

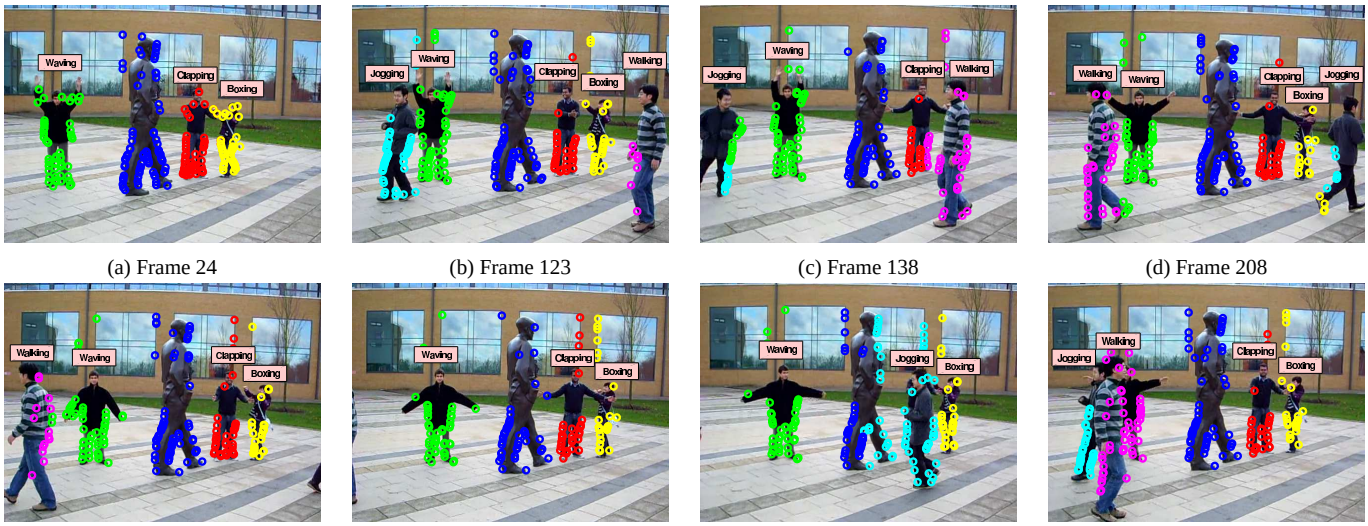


Figure 12: Automatic annotation of STIPs detected for multiple simultaneously actors for a number of frames from the Multi KTH dataset.

with high response in the Hough Space are kept. Furthermore, a post candidate approval is applied based on the slope of the lines. Figure 10 shows this process and the intermediate results. As can been seen, the erroneously detected (magenta colored) line can be discarded according to its steep slope. Furthermore, Line segments for the crossing actors are detected but due to a high amount of camera motion, it is not possible to detect good candidates for the other actors performing upper body acations, like *boxing*, *clapping* and *waving*.

4.2.4. STIP clustering in XT-space

We use the detected lines to cluster the STIPs by applying a point-line distance measure $d(x,t)$, and threshold according to a maximum distance d_{max} for each line segment:

$$d(x,t) = \frac{|(\mathbf{p}-\mathbf{q_1})\times(\mathbf{p}-\mathbf{q_2})|}{|\mathbf{q_2}-\mathbf{q_1}|} < d_{max} \qquad (12)$$

where $\mathbf{p}$ is the current STIP under evaluation, and $\mathbf{q_1}$ and $\mathbf{q_2}$ are two points lying on a detected line. The maximum distance d_{max} is set according to the size of the actors appearing in the dataset. After clustering the first set of STIPs, we exclude them and use the remaining STIPs for further clustering. We merge

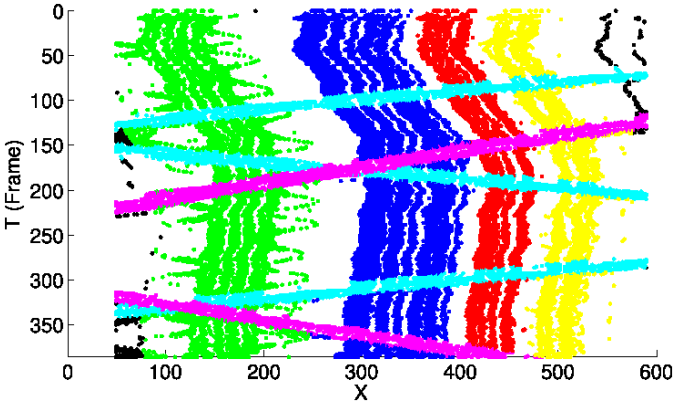


Figure 11: Actor-specific STIP clustering in the XT-space.

Table 2: STIP detection ratios (%): the number of STIPs detected on the actors with respect to the total number of detected STIPs, estimated for the MSR I and Multi-KTH datasets using our approach and state-of-the-art methods.

Method	MSR I	Multi-KTH
Our approach	**76.21**	**90.34**
Laptev el al. [45]	18.73	48.16
Dollár et al. [47]	21.36	16.03
Willems et al. [50]	24.02	20.24

the new subset of STIPs by morphological operations (see Figure 10.f) and use the resulting *blobs* to cluster the STIPs, by considering the spatio-temporal distance between a STIP and the contours. Figure 11 shows the resulting actor-specific STIP clustering in the XT-space, and in figure 12 the grouped STIPs are superimposed on a number of frames from the Multi-KTH dataset.

4.3. Evaluation of STIP detector

We evaluate our STIP detector by estimating a score for the number of detected STIPs for the actors in comparison to those detected in the background. Cao et al. [23] have recently reported that of all the STIPs detected by Laptev's STIP detector [45], only 18.73% correspond to the three actions performed by the actors in the MSR I, while the rest of the STIPs (81.27%) belong to the background. Ground truth bounding boxes are used to determine if a STIP belongs to an action instance. We evaluate our STIP detector on MSR I in a similar way, and detect **76**.**21**% STIPs for the actors. We observe that our detector tends to detect more points in the background, when applied to the sequences of MSR I with several moving people in the background. Our STIP detector is designed to detect interest point for people, hence it will also consider moving people in the background as candidates. We also conduct this experiment for the Multi-KTH dataset by manually annotating ground truth bounding boxes, and find that **89**.**35**% STIPs belong to the actors (see Figure 4). This is consistent with the concept of our STIP detector, and documents the effectiveness of our incorporated surround suppression followed up by imposing local and temporal constraints. Table 2 shows STIP detection ratios of the state-of-the-art methods, and clearly documents the superior performance of our STIP detector.

Table 3: State-of-the-art recognition accuracies (%) for the KTH, Weizmann and YouTube datasets. *Liu et al. [31] test on 8 out of the 11 YouTube actions.

Method	KTH	Weiz.	YouTube
Our approach	**96.35**	99.50	**86.98**
Lui et al. [17]	96.00	-	-
Yu et al. [44]	95.67	-	-
Kim et al. [15]	95.33	-	-
Wu et al. [20]	95.10	98.90	-
Cao et al. [23]	95.02	-	-
Kaâniche et al. [28]	94.67	-	-
Kovashka et al. [29]	94.53	-	-
Gilbert et al. [26]	94.50	-	-
Sadek et al. [37]	94.30	-	-
Liu & Shah [33]	94.16	-	-
Sun et al. [40]	94.00	97.80	-
Saghafi et al. [38]	93.94	-	-
Shao et al. [39]	93.89	-	-
Liu et al. [32]	93.80	-	71.20
Uemura et al. [41]	93.70	-	-
Lin et al. [16]	93.43	**100.00**	-
Yuan et al. [63]	93.30	-	-
Liu et al. [31]	92.30	-	76.10*
Yao et al. [22]	93.00	92.20	-
Schindler et al. [19]	92.70	**100.00**	-
Laptev et al. [62]	91.80	-	-
Jhuang et al. [48]	91.70	98.80	-
kläser et al. [56]	91.40	84.30	-
Yang et al. [12]	87.30	99.40	-
Wong et al. [51]	86.62	-	-
Willems et al. [50]	84.26	-	-
Niebles et al. [35]	81.50	-	-
Dollár et al. [47]	81.17	-	-
Schüldt et al. [65]	71.72	-	-
Gorelick et al. [66]	-	99.64	-
Thurau et al. [11]	-	94.40	-
Ali et al. [4]	-	92.60	-
bregonzio et al. [13]	-	-	64.00

The time complexity of our STIP computation highly depends on the size of the input video. For a video of size ($160 \times 120 \times 550$), the STIP computation, executed on a standard dual core Desktop PC (Intel(R) Core(TM)2 CPU 6400@2.13GHz 6GB RAM) using MATLAB R2010, takes approximately 10 mins.

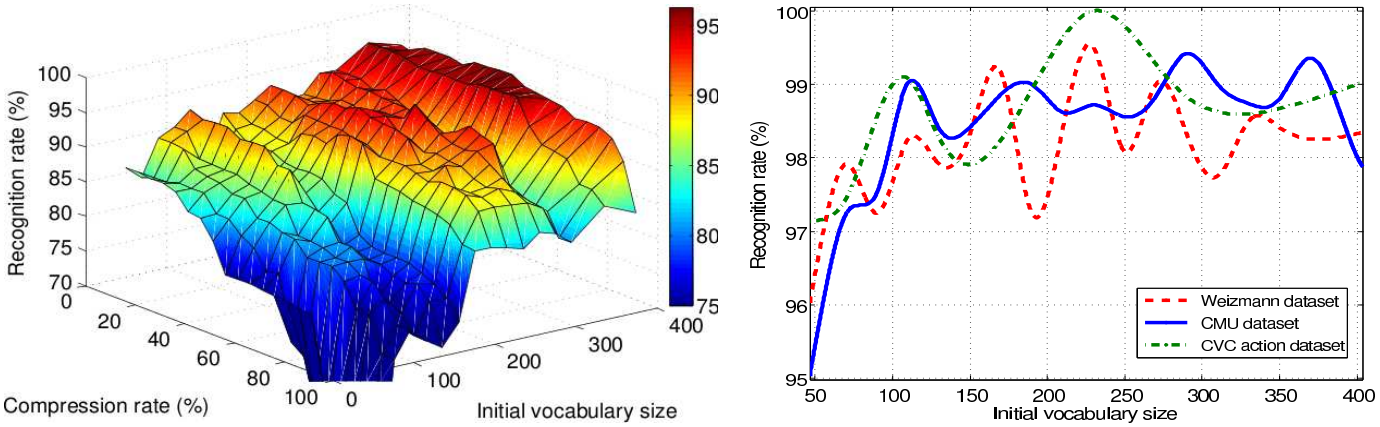
Figure 13 shows the perfomance of the STIP detector in complex scenarios. Despite of the camera movement, the STIP detector performs well (Figure 13(a) and (b)). However, in some cases, due to the combination of complex backgorund, low resolution and large background motion, the STIP detector loses focus and detects a larger number of background STIPs (Figure 13(c)) or an insufficient number of actor STIPs (Figure 13(d)).

4.4. Vocabulary building

The purpose of this experiment is to reveal the optimal initial vocabulary size and compression rate for our vocabulary build-

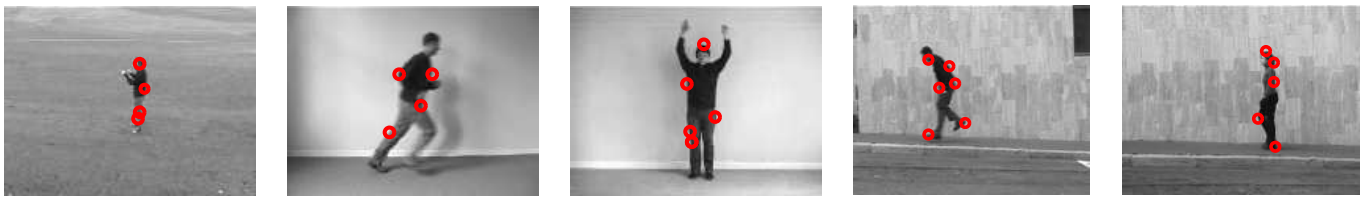
(a) (b) (c) (d)

Figure 13: Performance of the STIP detector in sequences with complex scenarios. Successful STIP detection is shown for frames of the (a) YouTube and (b) Hollywood 2 dataset, respectively. Additionally, the failure frames of (c) YouTube and (d) Hollywood 2 are also shown. In (a) and (b) our STIP detector successfully handles camera motion and the STIPs are detected only in the motion of interest. On the contrary, in the frames of (c) and (d), due to high background motion and difference in scene resolution the STIP detector loses the focus on the motion of the human actors.



(a) (b)

Figure 14: Revealing the influence of the vocabulary size and compression on the average action recognition rates. (a) A 3D Plot of the recognition rate, as a function of the initial vocabulary size and the compression rate, for the KTH dataset. (b) Recognition rates, as a function of the initial vocabulary size, for the three single actor datasets: CMU, CVC and Weizmann. The compression rate is fixed to 65%, i.e., 35% of the initial vocabulary size is used.



(a) KTH *boxing* (b) KTH *running* (c) KTH *waving* (d) Weizmann *skip* (e) Weizmann *walking*

Figure 15: Error-frames of the videos that are miss-classified for the KTH and Weizmann datasets. The first three frames depict miss-classified *boxing*, *running* and *waving* actions from the KTH dataset, respectively. The last two error-frames are *skip* and *walking* actions from the Weizmann dataset. These frames show cases which result in miss-classification. Due to low resolution only a limited number of STIPs are detected for the important body parts (arms and legs), which are taking major part in these actions.

ing strategy. We divide each dataset into 50% training, 20% validation and 30% testing partitions. The final training of the SVMs uses both the training and validation sets. The recognition rates are computed by averaging over 50 random instances of these sets. We conduct experiments using a similar vocabulary size range as Liu et al. [32], with an initial vocabulary size of 50 video-words and incrementing it up to 400. We weight the initial vocabulary size according to the pyramid level using a weight factor 2^{-s}, where s is the pyramid level. The vocabulary size is weighted to avoid the empty/singleton cluster creation in finer levels of the pyramid. We reduce the dimensionality of the final feature vectors for the SVM classifiers by apply-

ing vocabulary compression at each pyramid level. To choose the optimal vocabulary size and compression rate, we vary the initial vocabulary size range [50-400] with an increment of 20, and for each of these vocabularies we vary the compression rate from 0% to 95% with an increment of 5%. Figure 14.a shows the resulting 3D plot of the recognition rate as a function of the initial vocabulary size and the compression rate, for the KTH dataset. The maximum recognition rate indicates the optimal vocabulary size and compression rate. We observe that the best result is obtained at a compression rate upto **65**%, and the performance starts to degrade rapidly above **80**%. In Figure 14.b the recognition rate, as a function of the initial vocabulary size for the three other single actor datasets: CMU, CVC and Weizmann, is shown. We obtain approximately 100% recognition rate in the initial vocabulary size range [230-300] for the Weizmann, CMU and CVC datasets, which is similar to the the middle peak in Figure 14.a for KTH.

4.5. Benchmark testing

We use the KTH and Weizmann datasets for benchmark testing, and achieve an accuracy of **96**.**35**% for KTH and **99**.**50**% for Weizmann. Table 3 shows a comparison of the recognition rates of our approach and several other state-of-the-art methods for these two datasets. It should be noted that we achieve state-of-the-art recognition rate for KTH. We obtained this recognition with an initial vocabulary size of 350 and a 60% compression rate. The main reasons for this improvement are the selective STIP detection and the spatial pyramids, which capture the local characteristics of actions, and thereby reduce interclass confusion. The accuracy for Weizmann is approximately 100%, which is comparable to the state-of-the-art. Lin et al. [16] report a clear 100% recognition rate for Weizmann. However, this work applies a template matching technique, using holistic features extracted from global boundary box-based interest regions. Furthermore, it requires background subtraction and target tracking. In contrast, our approach uses local features and does not require any preprocessing. Since, Weizmann is a simple datasets without any further challenges, it favors global and holistic methods. In contrast, our approach is applicable for all types of scenes, including very challenging scenes of high complexity, which we will validate in the following.

We analyze the error-frames of the 0.50% videos of the Weizmann dataset, which are miss-classified. Similarly, we analyse the miss-classified frames from the confusion matrix for KTH. Figure 15 shows some example error-frames. Due to low resolution only a limited number of STIPs are detected for the important body parts (arms and legs), which are taking major part in actions like *boxing* and *running*. In these few cases this results in miss-classification.

4.6. Evaluation on complex scene

The main objective of this evaluation is to test the capability of our method to handle background clutter. For this purpose we choose the CMU action dataset and the CVC Action dataset with textured background. Despite the presence of strong background texture and clutter, we achieve a **100**.**0**% accuracy rate for CVC and 99.42% for CMU (see Table 5). The high performance for both of these dataset is consistence with the theoretical foundation of our proposed STIP detector. The detector's selective behavior, achieved by incorporating surround suppression and imposing local and temporal constraints, results in robustness to background texture and clutter.

4.7. Action recognition in movie and YouTube video clips

Next, we conduct experiments on movie and YouTube video clips, using the YouTube and Hollywood 2 action datasets. We achieve 99.13% recognition rate for the YouTube actions. Table 3 shows the comparison with other state-of-the-art method for this dataset. Our approach is far superior compared to the other reported methods, due to our STIP detector's capability to handle complex and challenging scenes with camera motion, cluttered background, and variation in scale, viewpoint and illumination.

For the Hollywood 2 dataset, the performance is evaluated as suggested in [34], i.e., by computing the average precision (AP) for each of the action classes and reporting the mean AP over all classes (MAP). Table 4 shows the AP for the actions in comparison to other state-of-the-art methods. The Hollywood 2 dataset contains very complex scenes from movies with no ground truth information available, and moreover the different instances of an action are sometimes viewed from different camera angles.

Notes: "*Answerphone* and *Handshake* are quite small, and therefore need a very complex set of compound features in order to classify the action over the background noise. In contrast, *FightPerson* and *DriveCar* use more global contextual features and therefore they work with lower level features."

4.8. Cross-data experiments

We perform exhaustive cross-data evaluation to test our proposed method in more realistic scenarios and use the KTH and Weizmann datasets for training data. We observe that the Weizmann dataset is not appropriate for training, and results in a poor 40% and 45% recognition rate for CVC and CMU, respectively. This is due to inadequate training data since Weizmann contains a very limited number of action instances per category compared to KTH. Table 5 shows the accuracy rates obtained using KTH as training. These cross-data results validate that our approach is applicable for more practical scenarios, where training and test data are coming from different sources.

The KTH dataset has only one common action, *two-hands-waving*, with the CMU action dataset. We use the KTH *running* sequence as negative data and obtain a **91**.**94**% recognition rate. It is noticeable, that the accuracy is actually higher for Weizmann (**100**%) and CMU (**99**.**42**%), than when training and testing on the same dataset, due to the sufficient action instances for training. Additionally, for CMU we only recognize one action, *two-hands-waving*, compared to five actions when both training and testing on CMU. On the contrary, the accuracy decreases by 3% for CVC, due to its lower inter-dataset correlation with KTH. For the Multi-KTH dataset we manually annotate the action labels as ground truth, using bounding boxes, and obtain **98**.**40**% accuracy. We perform another test using our automatic

Table 4: The average precision (%) and mean average precision (MAP) for the actions of Hollywood 2, using our apporach in comparison to the state-of-the-art.

Action	Marszalek [34]	Han [27]	Wang [43]	Gilbert [26]	Ullah [42]	**Our approach**
AnswerPhone	13.10	15.57	-	40.20	26.30	**41.60**
DriveCar	81.00	87.01	-	75.00	86.50	**88.49**
Eat	30.60	50.93	-	51.50	**59.20**	56.50
FightPerson	62.50	73.08	-	77.10	76.20	**78.20**
GetOutCar	8.60	27.19	-	45.60	45.70	**47.37**
HandShake	19.10	17.17	-	28.90	49.70	**52.50**
HugPerson	17.00	27.22	-	49.40	45.40	**50.30**
Kiss	57.60	42.91	-	56.60	**59.00**	57.35
Run	55.50	66.94	-	47.50	72.00	**76.73**
SitDown	30.00	41.61	-	62.00	62.40	**62.50**
SitUp	17.80	7.19	-	26.80	27.50	**30.00**
StandUp	33.50	48.61	-	50.70	58.80	**60.00**
MAP results	35.50	42.12	47.70	50.90	55.70	**58.46**

Table 5: Recognition accuracies (%) for cross-data evaluation trained on KTH and tested on other datasets: Weizmann, CVC, CMU, MSR I and Multi-KTH. The first row presents results when training and testing on the same dataset for Weizmann, CVC and CMU.

Method	Weizmann	CVC	CMU	MSR I	Multi-KTH
Our approach (without cross-data)	**99.50**	**100.00**	**99.42**	-	-
Our approach	**100.0**	**96.95**	**91.94**	**84.77**	**98.40**
Yuan et al. [63]	-	-	70.00	-	-
Cao et al. [23]	-	-	-	60.00	-
Gilbert et al. [25]	-	-	-	-	75.20
Gilbert et al. [26]	-	-	-	-	68.80
Uemura et al. [41]	-	-	-	-	65.40

action annotation described in section 4.2, and obtain a 94.20% recognition rate, which is comparable to the results of the manual annotation. For the MSR I dataset we achieve **84.77**% accuracy. The difficult part of MSR I is that some sequences contain moving people in the background depicted by the bounding box of the agent performing the action, which result in unwanted STIP in the background, and thereby a lower recognition rate compared to the other datasets. In conclusion, these results outperform the state-of-the-art significantly (see Table 5) and hereby validate the robustness of our method in more realistic action recognition scenarios. Although these datasets are very complex and contain several practical challenges: cluttered and moving backgrounds (including people and vehicles), camera motion and multiple actors, our approach performs robustly.

5. Conclusion

In this paper we have presented a novel approach for human action recognition in complex scenes. Our approach is based on selective STIPs which are detected by suppressing background SIPs and imposing local and temporal constraints, resulting in more robust STIPs for actors and less unwanted background STIPs. We apply a BoV model of local N-jet descriptors extracted at the detected STIPs and introduce a novel vocabulary building strategy by combining a spatial pyramid and vocabulary compression. Action class-specific SVM classifiers are trained to finally identify human actions.

The strong aspect of our proposed STIP detection method is, it can detect dense STIPs at the motion region without affected by the complex background. This is an important property to detect actions in complex scenarios. Regarding the weak aspect, our method suffers in the presence of other motion (presence of multiple actors) together with the region of action. In this scenario we detect several STIPs from different motion region results in poor classification.

In the current system, we use greedy approach for vocabulary compression. Sometimes, the time complexity is higher with this approach. A non-greedy method for vocabulary compression might be an interesting inclusion for the future work. Our automatic action annotation using STIP clustering works well for the multi-KTH dataset, yet it is not generalized for other multi-actor action datasets. The automatic action annotation for multi-actor datasets is a very difficult and challenging task. We could include more complex shape matching algorithm along with a human model in the XT-space to minimize the overlap in the STIP clusters of the moving and non-moving actors.

We have reported superior action recognition results in comparison to the state-of-the-art, when testing on benchmark datasets of simple scenes (96.35% accuracy for KTH and 99.50% for Weizmann), and similar performance for complex scenes (CVC and CMU). Additionally, we have shown state-of-the-art performance and proven the applicability of our approach for action recognition in movie and YouTube video clips by significantly outperforming other methods evaluated on the YouTube action dataset, and showing the highest mean average precision for the Hollywood 2 dataset. A comprehensive cross-data evaluation has been performed by separating the training

(KTH) and test datasets (CVC, CMU, MSR I and Multi-KTH). To our best knowledge we are the first to report exhaustive cross-data evaluation. Compared to state-of-the-art we have reported superior results by raising the recognition rates from approximately 60-75% to 85-100%.

Acknowledgements

This work has been supported by the Spanish Research Programs Consolider-Ingenio 2010:MIPRCV (CSD200700018); Avanza I+D ViCoMo (TSI-020400-2009-133); the Spanish project TIN2009-14501-C02-02; and the Danish National Research Councils - FTP under the research project "Big Brother *is* watching you!".

References

[1] T. Moeslund, A. Hilton, V. Krüger, A survey of advances in vision-based human motion capture and analysis, CVIU 104 (2-3) (2006) 90–126.
[2] R. Poppe, A survey on vision-based human action recognition, IVC 28 (6) (2010) 976–990.
[3] D. Weinland, R. Ronfard, E. Boyer, A survey of vision-based methods for action representation, segmentation and recognition, INRIA Report RR-7212 (2010) 54–111.
[4] S. Ali, A. Basharat, M. Shah, Chaotic invariants for human action recognition, in: ICCV, 2007.
[5] N. Nguyen, D. Phung, S. Venkatesh, H. Bui, Learning and detecting activities from movement trajectories using the hierarchical hidden markov model, in: CVPR, 2005.
[6] A. Galata, N. Johnson, D. Hogg, Learning variable-length markov models of behavior, CVIU 81 (3) (2001) 398–413.
[7] O. Boiman, M. Irani, Detecting irregularities in images and in video, in: ICCV, 2005.
[8] M. Rodriguez, J. Ahmed, M. Shah, Action mach: A spatio-temporal maximum average correlation height filter for action recognition, in: CVPR, 2008.
[9] N. Ikizler, D. Forsyth, Searching video for complex activities with finite state models, in: CVPR, 2007.
[10] L. Zelnik-Manor, M. Irani, Statistical analysis of dynamic actions, PAMI 28 (9) (2006) 1530–1535.
[11] C. Thurau, V. Hlavac, Pose primitive based human action recognition in videos or still images, in: CVPR, 2008.
[12] W. Yang, Y. Wang, G. Mori, Recognizing human actions from still images with latent poses, in: CVPR, 2010.
[13] M. Bregonzio, J. Li, S. Gong, T. Xiang, Discriminative topics modelling for action feature selection and recognition, in: BMVC, 2010.
[14] Z. Jiang, Z. Lin, L. Davis, A tree-based approach to integrated action localization, recognition and segmentation, in: ECCV Workshops, 2010.
[15] T. Kim, S. Wong, R. Cipolla, Tensor canonical correlation analysis for action classification, in: CVPR, 2007.
[16] Z. Lin, Z. Jiang, L. Davis, Recognizing actions by shape-motion prototype trees, in: ICCV, 2009.
[17] Y. M. Lui, J. R. Beveridge, M. Kirby, Action classification on product manifolds, in: CVPR, 2010.
[18] F. Nater, H. Grabner, L. V. Gool, Exploiting simple hierarchies for unsupervised human behavior analysis, in: CVPR, 2010.
[19] K. Schindler, L. van Gool, Action snippets: How many frames does human action recognition require, in: CVPR, 2008.
[20] X. Wu, W. Liang, Y. Jia, Incremental discriminative-analysis of canonical correlations for action recognition, in: ICCV, 2009.
[21] M. Yang, F. Lv, W. Xu, K. Yu, Y. Gong, Human action detection by boosting efficient motion features, in: ICCV, 2009.
[22] A. Yao, J. Gall, L. V. Gool, A hough transform-based voting framework for action recognition, in: CVPR, 2010.
[23] L. Cao, Z. Liu, T. Huang, Cross-dataset action detection, in: CVPR, 2010.
[24] O. Duchenne, I. Laptev, J. Sivic, F. Bach, J. Ponce, Automatic annotation of human actions in video, in: ICCV, 2009.
[25] A. Gilbert, J. Illingworth, R. Bowden, Action recognition using mined hierarchical compound features, PAMI 99 (PrePrints).
[26] A. Gilbert, J. Illingworth, R. Bowden, Fast realistic multi-action recognition using mined dense spatio-temporal features, in: ICCV, 2009.
[27] D. Han, L. Bo, C. Sminchisescu, Selection and context for action recognition, in: ICCV, 2009.
[28] M. B. Kaâniche, F. Brémond, Gesture recognition by learning local motion signatures, in: CVPR, 2010.
[29] A. Kovashka, K. Grauman, Learning a hierarchy of discriminative space-time neighborhood features for human action recognition, in: CVPR, 2010.
[30] I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, Learning realistic human actions from movies, in: CVPR, 2008.
[31] J. Liu, Y. Yang, M. Shah, Learning semantic visual vocabularies using diffusion distance, in: CVPR, 2009.
[32] J. Liu, J. Luo, M. Shah, Recognizing realistic actions from videos "in the wild", in: CVPR, 2009, the YouTube dataset is available at http://www.cs.ucf.edu/~liujg/YouTube_Action_dataset.html.
[33] J. Liu, M. Shah, Learning human actions via information maximization, in: CVPR, 2008.
[34] M. Marszalek, I. Laptev, C. Schmid, Actions in context, in: CVPR, 2009, the Hollywood 2 dataset is available at http://www.irisa.fr/vista/actions/hollywood2.
[35] J. Niebles, H. Wang, L. Fei-Fei, Unsupervised learning of human action categories using spatial-temporal words, IJCV 79 (3) (2008) 299–318.
[36] K. Prabhakar, S. Oh, P. Wang, G. Abowd, J. Rehg, Temporal causality for the analysis of visual events, in: CVPR, 2010.
[37] S. Sadek, A. Al-Hamadi, B. Michaelis, U. Sayed, Toward robust action retrieval in video, in: BMVC, 2010.
[38] B. Saghafi, E. Farahzadeh, D. Rajan, A. Sluzek, Embedding visual words into concept space for action and scene recognition, in: BMVC, 2010.
[39] L. Shao, R. Gao, A wavelet based local descriptor for human action recognition, in: BMVC, 2010.
[40] X. Sun, M. Chen, A. Hauptmann, Action recognition via local descriptors and holistic features, in: CVPR, 2009.
[41] H. Uemura, S. Ishikawa, K. Mikolajczyk, Feature tracking and motion compensation for action recognition, in: BMVC, 2008, the Multi-KTH dataset is available at http://www.openvisor.org/video_details.asp?idvideo=303.
[42] M. Ullah, S. Parizi, I. Laptev, Improving bag-of-features action recognition with non-local cues, in: BMVC, 2010.
[43] H. Wang, M. Ullah, A. Kläser, I. Laptev, C. Schmid, Evaluation of local spatio-temporal features for action recognition, in: BMVC, 2009.
[44] T. Yu, T. Kim, R. Cipolla, Real-time action recognition by spatiotemporal semantic and structural forests, in: BMVC, 2010.
[45] I. Laptev, T. Lindeberg, Space-time interest points, in: ICCV, 2003.
[46] C. Harris, M. Stephens, A combined corner and edge detector, in: Alvey Vision Conference, 1988.
[47] P. Dollár, V. Rabaud, G. Cottrell, S. Belongie, Behavior recognition via sparse spatio-temporal features, in: VS-PETS, 2005.
[48] H. Jhuang, T. Serre, L. Wolf, T. Poggio, A biologically inspired system for action recognition, in: ICCV, 2007.
[49] A. Oikonomopoulos, I. Patras, M. Pantic, Spatiotemporal salient points for visual recognition of human actions, SMC-B 36 (3) (2006) 710–719.
[50] G. Willems, T. Tuytelaars, L. V. Gool, An efficient dense and scale-invariant spatio-temporal interest point detector, in: ECCV, 2008.
[51] S. Wong, R. Cipolla, Extracting spatiotemporal interest points using global information, in: ICCV, 2007.
[52] P. Beaudet, Rotationally invariant image operators, in: ICPR, 1978.
[53] T. Lindeberg, Feature detection with automatic scale selection, IJCV 30 (2) (1998) 79–116.
[54] C. Schmid, R. Mohr, C. Bauckhage, Evaluation of interest point detectors, IJCV 37 (2) (2000) 151–172.
[55] P. Turcot, D. G. Lowe, Better matching with fewer features: The selection of useful features in large database recognition problems, in: ICCV Workshops, 2009.
[56] A. Kläser, M. Marszalek, C. Schmid, A spatio-temporal descriptor based on 3d-gradients, in: BMVC, 2008.
[57] J. Koenderink, A. V. Doorn, Representation of local geometry in the visual system, Biological Cybernetics 55 (1987) 367–375.
[58] I. Laptev, T. Lindeberg, Local descriptors for spatio-temporal recognition,

in: First International Workshop on Spatial Coherence for Visual Motion Analysis, 2004.
[59] P. Scovanner, S. Ali, M. Shah, A 3-dimensional sift descriptor and its application to action recognition, in: ACM International Conference on Multimedia, 2007.
[60] H. Bay, T. Tuytelaars, L. V. Gool, Surf: Speeded up robust features, in: ECCV, 2006.
[61] D. Lowe, Distinctive image features from scale-invariant keypoints, IJCV 60 (2) (2004) 91–110.
[62] I. Laptev, B. Caputo, C. Schüldt, T. Lindeberg, Local velocity-adapted motion events for spatio-temporal recognition, IJCV 108 (3) (2007) 207–229.
[63] J. Yuan, Z. Liu, Y. Wu, Discriminative subvolume search for efficient action detection, in: CVPR, 2009, the MSR dataset is available at http://research.microsoft.com/en-us/um/people/zliu/ActionRecoRsrc.
[64] P. Matikainen, M. Hebert, R. Sukthankar, Representing pairwise spatial and temporal relations for action recognition, in: ECCV, 2010, pp. 508–521.
[65] C. Schüldt, I. Laptev, B. Caputo, Recognizing human actions: A local svm approach, in: ICPR, 2004, the KTH dataset is available at http://www.nada.kth.se/cvap/actions.
[66] L. Gorelick, M. Blank, E. Shechtman, M. Irani, R. Basri, Actions as space-time shapes, PAMI 29 (12) (2007) 2247–2253, the Weizmann dataset is available at http://www.wisdom.weizmann.ac.il/~vision/SpaceTime Actions.html.
[67] Y. Ke, R. Sukthankar, M. Hebert, Event detection in crowded videos, in: ICCV, 2007, the CMU dataset instructions are available at http://www.cs.cmu.edu/~yke/video/#Dataset.
[68] The CVC dataset is available at http://iselab.cvc.uab.es/files/Tools/Cvc-ActionDataSet/index.htm.
[69] E. Soria, J. Martin, R. Magdalena, M. Martinez, A. Serrano, Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques, IGI Global, 2009, Ch. 11, pp. 242–264.
[70] C. B. Do, A. Y. Ng, Transfer learning for text classification, Journal of Advances in Neural Information Processing Systems 18 (2006) 299–306.
[71] C. Grigorescu, N. Petkov, M. A. Westenberg, Contour and boundary detection improved by surround suppression of texture edges, IVC 22 (8) (2004) 609–622.
[72] J. Fan, Y. Wu, S. Dai, Discriminative spatial attention for robust tracking, in: ECCV, 2010, pp. 480–493.
[73] V. Mahadevan, N. Vasconcelos, Spatiotemporal saliency in dynamic scenes, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (2010) 171–177.
[74] D. Gao, S. Han, N. Vasconcelos, Discriminant saliency, the detection of suspicious coincidences, and applications to visual recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2009) 989–1005.
[75] L. Bretzner, T. Lindeberg, Feature tracking with automatic selection of spatial scales, CVIU 71 (3) (1998) 385–392.
[76] I. Laptev, On space-time interest points, IJCV 64 (2/3) (2005) 107–123.
[77] N. Slonim, N. Tishby, Agglomerative information bottleneck, in: NIPS, 1999.
[78] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, http://www.csie.ntu.edu.tw/~cjlin/libsvm (2001).
[79] R. O. Duda, P. E. Hart, Use of the hough transformation to detect lines and curves in pictures, Communications of the ACM 15 (1) (1972) 11–15.