An Efficient Dense and Scale-Invariant Spatio-Temporal Interest Point Detector

Geert Willems[1], Tinne Tuytelaars[1], and Luc Van Gool[1,2]

[1] ESAT-PSI, K.U. Leuven, Belgium,
{gwillems,tuytelaa,vangool}@esat.kuleuven.be
[2] ETH, Zürich, Switzerland

Abstract. Over the years, several spatio-temporal interest point detectors have been proposed. While some detectors can only extract a sparse set of scale-invariant features, others allow for the detection of a larger amount of features at user-defined scales. This paper presents for the first time spatio-temporal interest points that are at the same time scale-invariant (both spatially and temporally) and densely cover the video content. Moreover, as opposed to earlier work, the features can be computed efficiently. Applying scale-space theory, we show that this can be achieved by using the determinant of the Hessian as the saliency measure. Computations are speeded-up further through the use of approximative box-filter operations on an integral video structure. A quantitative evaluation and experimental results on action recognition show the strengths of the proposed detector in terms of repeatability, accuracy and speed, in comparison with previously proposed detectors.

1 Introduction

As video becomes a ubiquitous source of information, video analysis (*e.g.* [1]) and action recognition (*e.g.* [2, 3]) have received a lot of attention lately. In this context, local viewpoint invariant features, so successful in the field of object recognition and image matching, have been extended to the spatio-temporal domain [4–8]. These extensions take the 3D nature of video data into account and localize features not only spatially but also over time.

Laptev and Lindeberg [5] were the first to propose such a spatio-temporal extension, building on the Harris-Laplace detector proposed by Mikolajczyk and Schmid [9]. They typically detect only a sparse set of features as a time-consuming iterative procedure has to be repeated for each feature candidate separately. Furthermore, the iterative procedure often diverges. As a result, detecting a low number of features is a necessity to keep the computation time under control.

Dollár *et al.* [6], on the other hand, claim that direct 3D counterparts to 2D interest point detectors are inadequate for the detection of spatio-temporal feature points, since true spatio-temporal corners are quite rare. They propose to select local maxima over space and time of a response function based on a spatial Gaussian convolved with a quadrature pair of 1D Gabor-filters along the time axis. However, their features are not scale-invariant. The size of these *cuboids* is determined by the user.

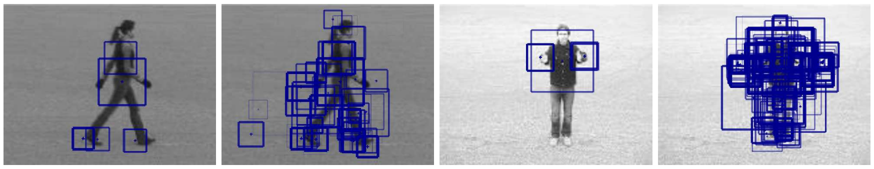


Fig. 1. The proposed scale-invariant spatio-temporal interest points (Hes-STIP). The density of features can be varied from very sparse (first and third image) to very dense (second and fourth image), simply by changing the threshold and with minimal effect on the computation time.

Oikonomopoulos *et al*. [8] have proposed a spatio-temporal extension of the salient region detector proposed by Kadir and Brady [10]. The features are scale-invariant yet sparse, as was also the case for the original spatial detector.

Recently, Wong and Cipolla [11] have developed a novel method for extracting spatio-temporal features using global information. Using their method based on the extraction of dynamic textures, only a sparse set of features is needed for action recognition. However, all input videos need to be preprocessed into samples containing one iteration of the action each.

Also related is the work of Ke *et al*. [7] on visual event detection. They build on the concept of integral video to achieve realtime processing of video data. However, rather than relying on interest points, they use dense spatio-temporal Haar-wavelets computed on the optical flow. Discriminative features are then selected during a training stage. This results in application dependent features which are, again, not scale-invariant.

Table 1 summarizes the most important properties of the previously mentioned detectors. The currently available spatio-temporal interest point (STIP) detectors [5, 6, 8] are computationally expensive and are therefore restricted to the processing of short or low resolution videos. The existing scale-invariant feature detectors [5, 8] only yield a sparse set of features.

In this paper, we present a novel spatio-temporal feature detector which is the first to obtain a dense set of scale-invariant features (fig 1) in an efficient way. Our main contributions can be summarized as follows. First, we show that features can be localized both in the spatio-temporal domain and over both scales simultaneously when using the determinant of the Hessian as saliency measure. We thus remove the need for the iterative scheme in the work by Laptev and Lindeberg [5]. Second, building on the

detector	scale selection	feature set	efficient	app. independent
Laptev [5]	yes, iterative	sparse (rare)	no	yes
Dollár [6]	no	dense	no	yes
Ke [7]	no	dense	yes (box filters)	no
Oikonomopoulos [8]	yes	sparse (rare)	no	yes
proposed method	yes	dense	yes (box filters)	yes

Table 1. Comparison between spatio-temporal interest point detectors.

work of Bay *et al.* [12] and Ke *et al.* [7], we create an efficient implementation of the detector by approximating all $3D$ convolutions using box-filters. Finally, we compare the repeatability of our detector with the two best known state-of-the-art detectors and show experimental results on action recognition and video synchronization.

2 Spatio-temporal interest point detection

In this section, we first briefly recapitulate some basic scale space terminology as well as the Harris-Laplace-based space-time interest points (HL-STIP) of Laptev and Lindeberg [5]. Next, we propose our Hessian-based spatio-temporal interest point (Hes-STIP) detector and discuss its advantage w.r.t. localisation and scale selection.

2.1 Some scale space principles

Starting from a spatio-temporal signal $f(\cdot)$, a spatio-temporal scale space representation is obtained by convolving $f(\cdot)$ with a Gaussian kernel [13]

$$L(\cdot;\sigma^2,\tau^2) = g(\cdot;\sigma^2,\tau^2) * f(\cdot) \tag{1}$$

with σ and τ the spatial and temporal scales respectively. Building blocks of virtually any method working in scale space are the Gaussian derivatives

$$L_{x^k y^l t^m}(\cdot;\sigma^2,\tau^2) = \partial_{x^k y^l t^m} g(\cdot;\sigma^2,\tau^2) * f(\cdot) \tag{2}$$

The amplitude of these spatio-temporal derivatives decreases with scale. To obtain scale invariance, *scale-normalized derivatives* should be used, defined as

$$L^{norm}_{x^k y^l t^m}(\cdot;\sigma^2,\tau^2) = \sigma^{k+l}\tau^m L_{x^k y^l t^m}(\cdot;\sigma^2,\tau^2) \tag{3}$$

Working with scale-normalized derivatives ensures that the same values are obtained irrespective of the scale.

Scale selection refers to the process of selecting a characteristic scale [13]. This can be achieved by searching local extrema of some saliency measure. In this context, normalization factors $\sigma^{\gamma(k+l)}$ or $\tau^{\lambda m}$ are often used. This is known as γ*-normalization*. By playing with different values for γ and λ one can ensure that, at least for prototypical patterns such as a perfect Gaussian blob, the saliency measure reaches a local maximum and that the spatio-temporal extent of the detected features corresponds to some meaningful entity (*e.g.* the correct scale of the Gaussian blob).

2.2 Introduction to space-time interest points

Since our approach has some similarity with the space-time interest points proposed by Laptev and Lindeberg [5], we shortly describe their method that extends the 2D scale-invariant Harris-Laplace corner detector [14] into the spatio-temporal domain. To this end, a 3 × 3 spatio-temporal second-moment matrix

$$\mu = g(\cdot;\sigma_i^2,\tau_i^2) \star \begin{pmatrix} L_x^2 & L_xL_y & L_xL_t \\ L_xL_y & L_y^2 & L_yL_t \\ L_xL_t & L_yL_t & L_t^2 \end{pmatrix} \tag{4}$$

is defined, with σ_i and τ_i the spatial and temporal integration scales. The strength of each interest point at a certain scale is then computed by the (extended) Harris corner function

$$S = det(\mu) - k' trace^3(\mu) \tag{5}$$

with a typical value for $k' = 0.001$. To recover the spatio-temporal extent of $f(\cdot)$, scale selection is applied based on the γ-normalized Laplacian, defined as

$$(\nabla^2 L)^{\gamma norm} = \sigma^{2a}\tau^{2b}L_{xx} + \sigma^{2a}\tau^{2b}L_{yy} + \sigma^{2c}\tau^{2d}L_{tt} \tag{6}$$

In order to achieve an extremum at the correct scales for a prototype Gaussian blob, the normalizing parameters are set to $a = 1, b = 1/4, c = 1/2, d = 3/4$ [5].

To find points in scale-space that are both maxima of the Harris corner function (5) in space/time and extrema of the normalized Laplacian (6) over both scales, an iterative scheme must be used. First, interest points are detected for a sparsely distributed set of scales. Then each point is iteratively updated until convergence by alternating between scale optimization and re-detection of the position given the novel scales.

2.3 Hessian-based localization and scale selection

In this paper, we propose the use of the Hessian matrix for spatio-temporal feature detection:

$$H(\cdot; \sigma^2, \tau^2) = \begin{pmatrix} L_{xx} & L_{xy} & L_{xt} \\ L_{yx} & L_{yy} & L_{yt} \\ L_{tx} & L_{ty} & L_{tt} \end{pmatrix} \tag{7}$$

The strength of each interest point at a certain scale is then computed by

$$S = |det(H)| \tag{8}$$

This can be considered as a spatio-temporal extension of the saliency measure proposed by Beaudet for blob detection [15]. However, unlike the 2D case, a positive value of S does not guarantee all eigenvalues of $H(., \sigma^2, \tau^2)$ having the same sign. As a result, apart from blobs, also saddle points can give rise to local extrema. For most applications, the nature of the interest points does not really matter – as long as they can be detected reliably, i.e. with high repeatability. If for whatever reason only blobs should be found, one can easily check the sign of the eigenvalues and reject the saddle points in a postprocessing step.

Using the Hessian matrix, scale selection can be realized in various ways. First, we show how to use γ-normalization, similar to the work of Laptev and Lindeberg. Next, a different strategy is proposed, leading to a more efficient solution.

Scale selection through γ-normalization Using γ-normalization, we alter the saliency measure so as to ensure that the 'correct' scales σ_0 and τ_0 are found for a perfect Gaussian blob $g(x, y, t; \sigma_0^2, \tau_0^2)$. At the center of this blob, the determinant is solely determined by the first term $L_{xx}L_{yy}L_{tt}$ as all other terms vanish. The γ-normalized determinant at the center can thus be written as

$$L_{xx}^{\gamma norm} L_{yy}^{\gamma norm} L_{tt}^{\gamma norm} = \sigma^{2p}\tau^{2q} L_{xx}L_{yy}L_{tt} \tag{9}$$

To find the extrema, we differentiate $det(H)^{\gamma norm}$ with respect to the spatial and temporal scale parameters σ^2 and τ^2, and set these derivatives equal to zero. Again, all terms but the first vanish at the center of the Gaussian blob. From this analysis, it follows that the local extrema $\tilde{\sigma}$ and $\tilde{\tau}$ coincide with the correct scales σ_0 and τ_0 if we set $p = 5/2$ and $q = 5/4$.

Note that these values are related to the values a, b, c, and d for the normalization of the Laplacian, namely $p = 2a + c$ and $q = 2b + d$. This reflects the fact that the determinant at the center of the Gaussian blob reduces to the product of two spatial second-order derivates and one temporal second-order derivative.

This way, a γ-normalized operator is obtained with $\gamma \neq 1$, as was also the case with the γ-normalized Laplacian (6). This implies, however, that the measure used for scale selection is *not* truely scale invariant, and as a result cannot be used to find local maxima over scales. With two different criteria to optimize, we are again bound to use an iterative method.

Simultaneous localization and scale selection In contrast with the normalized Laplacian, scale invariance *and* good scale selection can be achieved simultaneously with the scale-normalized determinant of the Hessian. Indeed, using $p = 2$ and $q = 1$ (*i.e.* $\gamma = 1$) in equation 9, we obtain the following relationship between the local extrema $(\tilde{\sigma}, \tilde{\tau})$ and the correct scales (σ_0, τ_0) for a Gaussian blob:

$$\tilde{\sigma}^2 = \frac{2}{3}\sigma_0^2 \quad \tilde{\tau}^2 = \frac{2}{3}\tau_0^2 \tag{10}$$

In general, it can be shown that, in D dimensions, the determinant of the scale-normalized Hessian, with $\gamma = 1$, reaches an extremum at the center of a Gaussian blob $g(\mathbf{x}; \boldsymbol{\sigma_0})$ with $\boldsymbol{\sigma_0} = [\sigma_{0,1}, \ldots, \sigma_{0,D}]$, for scales

$$\tilde{\boldsymbol{\sigma}} = \sqrt{\frac{2}{D}}\boldsymbol{\sigma_0} \tag{11}$$

Even though the detected scales $\tilde{\sigma}, \tilde{\tau}$ do not coincide with the correct scales σ_0, τ_0, they are related by a fixed scale factor. This makes it trivial to obtain the latter [3].

Since we now have a single, scale-invariant measure that can be used both for the localization as well as for the selection of the spatial and temporal scale, a non-iterative method can be used. To this end, we select local extrema over the $5D$ space defined by (x, y, t, σ, τ). We then multiply the scales of each interest point found with the factor $\sqrt{3/2}$ to obtain the real scales of the underlying signal. This brings a clear speed advantage over the iterative procedure of [5], avoids problems with convergence and allows for the extraction of any number of features simply by changing the threshold of the saliency measure.

[3] Note that the use of the determinant of the Hessian is crucial for the above method to work. Using a scale-normalized version of the Laplacian, no extrema are found unless for $\tilde{\boldsymbol{\sigma}} = 0$.

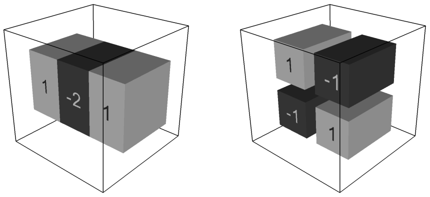


Fig. 2. The two types of box filter approximations for the $2+1$D Gaussian second order partial derivatives in one direction (left) and in two directions (right).

2.4 Implementation details

Integral video In the previous section, we showed that the use of the determinant of the Hessian as a saliency measure allows for the direct localisation of spatio-temporal interest points in a $5D$ space. Nevertheless, computing the determinant of the Hessian at many positions and many scales can become computationally prohibitive.

In [7], Ke *et al.* combine box-filters in order to obtain volumetric features. By using an *integral video* structure - the spatio-temporal generalization of integral images - these box-filters can be computed very efficiently and even allow for realtime action detection.

We also build on integral videos in order to make this part of the problem tractable. In a first step, a video containing F frames of dimension $W \times H$ is converted into an integral video structure where an entry at location $\mathbf{x} = (x, y, t)$ holds the sum of all pixels in the rectangular region spanned by $(0,0) - (x,y)$, summed over all frames $[0,t]$. Using integral videos, the sum of values within any rectangular volume can be approximated with 8 additions, independent of the volume's size.

We approximate all Gaussian second-order derivatives very roughly with box-filter equivalents as was done in 2D by Bay *et al.* [12]. In total there are 6 different second order derivatives in the spatio-temporal domain: D_{xx}, D_{yy}, D_{tt}, D_{xy}, D_{tx} and D_{ty}, which can be computed using rotated versions of the two box-filters shown in figure 2.

Spatio-temporal search space Thanks to the use of the integral video structure and the box-filters, the scale spaces do not have to be computed hierarchically but can be efficiently implemented by upscaling the box-filters. Each octave is divided into 5 scales, with a ratio between subsequent scales in the range $1.2 - 1.5$ for the inner 3 scales.

The determinant of the Hessian is computed over several octaves of both the spatial and temporal scale. For a combination of octaves o_σ and o_τ, each pair of scales (σ_i, τ_i) results in a cube structure filled with Hessian-based strengths. Once all cubes have been filled, we use a non-maximum suppression algorithm to obtain all extrema within the obtained 5 dimensional search-space (x, y, t, σ, τ). Upon the detection of an extremum, a gradient descent is started to obtain *sub-pixel* accuracy in all 5 values.

Note that, although it is possible to search for local extrema in the 5 dimensional search-space, this is not always required by the application. Indeed, depending on the application, the temporal or spatial scale can be fixed, reducing the search space to 3 or 4 dimensions. Another strategy, similar to what is done in [16], is to consider only a small band around a chosen temporal scale.

Descriptor To describe the interest points, we implemented an extended version of the SURF descriptor [12]. Around each interest point with spatial scale σ and temporal scale τ, we define a rectangular volume with dimensions $s\sigma \times s\sigma \times s\tau$ with s a user-defined magnification factor (typically 3). The volume is subsequently divided into $M \times M \times N$ bins, where M and N are the number of bins in the spatial and temporal direction respectively. The bins are filled by a weighted sum of uniformly sampled responses of the 3 axis-aligned Haar-wavelets d_x, d_y, d_t. For each bin, we store the vector $\boldsymbol{v} = (\sum d_x, \sum d_y, \sum d_t)$. We do not include the sums over the absolute values, as done in [12], as they proved to be of no significant benefit in our experiments while doubling the descriptor size.

If invariance to (spatial) rotation is required, we compute the dominant orientation as proposed by [12] except that, for the spatio-temporal case, all Haar-wavelets used in this step stretch out over the full length of the temporal scale of the interest point. Due to space limitations, we refer the reader to the technical report [17] for more in-depth information regarding the implementation issues of both the detector and the descriptor.

3 Quantitative evaluation

3.1 Methodology

To quantitatively evaluate our detector, we measure the *repeatability*, i.e. how often does it extract the same spatio-temporal features in spite of various geometric and photometric transformations? To this end, we define an *overlap criterion* between two corresponding features defined by ellipsoids E_a and E_b, similar to the 2D overlap criterion for elliptical regions used in [9]. We define the *overlap error* ϵ_O as

$$\epsilon_O = 1 - \frac{V_{E_a} \cap V_{T.E_b}}{V_{E_a} \cup V_{T.E_b}} \tag{12}$$

where V_E represents the volume of an ellipsoid, and T stands for the geometric transformation between the two videos. Two spatio-temporal features are said to *correspond* if the overlap error ϵ_O is below a predefined threshold. The *repeatability* of a detector for a transformation T is then computed as the ratio between the number of corresponding features found and the minimum number of features detected in the common part of the two videos. As in [9], we rescale all features to a fixed size before computing the overlap, so as to prevent a bias towards larger regions. For our experiments, we set the overlap error threshold to 55%. This is equivalent to the threshold of 40% used in [9] for 2D detectors. Figure 3.1 shows what this means in practice. To put this threshold in perspective, we randomly create several sets of features inside a $3D$ volume where the number of features per set, the dimension of the volume and the scale ranges are chosen to be similar to the obtained data used in section 3.2. Using the specified threshold, repeatability between random sets lies around 6%.

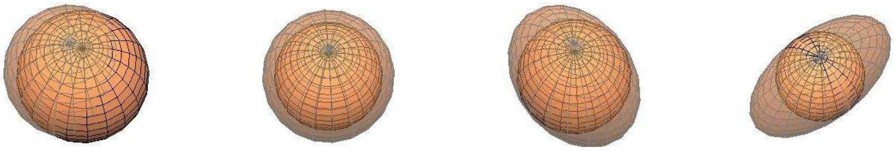


Fig. 3. Some examples of pairs of spatio-temporal ellipsoids with an overlap error equal to our threshold value of 55%. This corresponds to (from left to right) a position error with respect to the diameter of 12%, a uniform (spatial + temporal) scale change of 22%, a 2D spatial scaling of 35%, a 1D temporal scaling of 88%.

3.2 Discussion

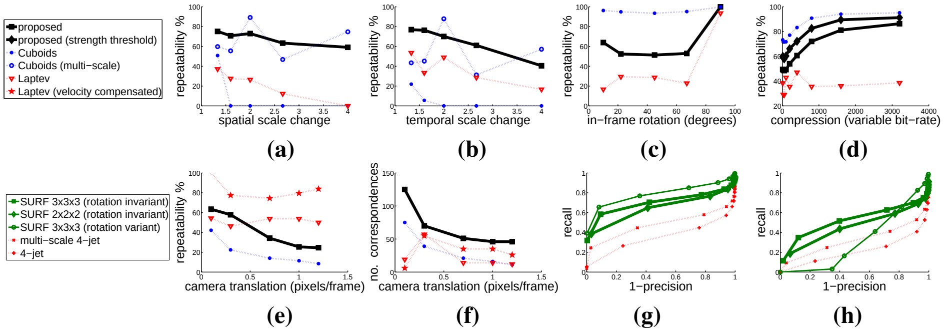
We compare our Hes-STIP detector with the HL-STIP detector of [5] at multiple scales and cuboids [6] extracted both at a single scale and at multiple scales. For all of these, we have used executables made available by the respective authors with default parameters. The multi-scale versions were run at scales of 2, 4, 8 and 16, resulting in 16 scale combinations for space and time. Figure 4 shows the results. These were obtained based on artificial transformations of several randomly selected clips from the TRECVID 2006 dataset [4].

Scale changes The scale invariance of our detector stands out when we test the effect of spatial and temporal scale changes (fig. 4(a,b)). The repeatability degrades only slightly in spite of significant scale changes. The single-scale cuboids can only cope with minor scale changes - in fact, only as long as it stays within the error bounds of our overlap criterion. Extracting features at multiple scales only partially overcomes this problem: good repeatability scores are then obtained when the scale factor equals a power of 2, i.e. when the rescaling corresponds exactly to the multiscale resolution. However, for other scale factors, the results again drop significantly. The HL-STIPs only give moderate results.

In-plane rotation From figure 4(c), one can conclude that the Hes-STIPs are relatively sensitive to in-plane rotations (50% repeatability for rotations of 45 degrees), although not as much as the HL-STIPs (30% repeatability). The non-scale-invariant cuboids are more robust in this respect. This is in line with findings on 2D scale-invariant interest points: the higher complexity needed to increase the level of invariance comes at a price and reduces the robustness.

Compression The same also holds for the results under increased levels of compression (fig. 4(d)). However, we detect many more features than the other detectors. If we increase the threshold for our detector such that the number of extracted features is similar, the repeatability improves and comes closer to that of the non-scale-invariant cuboids (red line in figure 4(b)). HL-STIPs again perform quite poorly.

[4] Due to lack of space, we cannot show the results for all the clips. However, the overall trends were the same. For each transformation, the behaviour of one clip is shown.



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

(e) **(f)** **(g)** **(h)**

Fig. 4. Quantitative evaluation: repeatability scores for spatial scale changes (a), temporal scale changes (b), in-plane rotations (c), compression (d), and horizontal camera translation (e); total number of corresponding features found for camera translation (f); precision vs. recall of different descriptors for a spatial scale change of 1.5 (g) and a rotation of 22 degrees (h).

Camera motion Finally, we show results for a moving camera (figure 4(e)). This is simulated by gradually translating the image over time. Laptev developed a velocity compensated version of his detector to deal with this case [16], and this seems to give high repeatability scores indeed. Our detector performs only moderately, as could be expected since this type of transformation was not taken into account during the design of the detector. Still, it performs significantly better than cuboids. This is the only type of transformation where HL-STIPs outperform the other detectors - even without motion compensation. Nevertheless, since our method extracts a dense set of features the total number of correspondences found by our method is still higher than what is obtained by the other methods. This is illustrated in figure 4(f), where we plot the absolute number of corresponding features found.

3.3 Evaluation of the descriptor

We also evaluate the quality of our descriptor and compare it with 4-jet [18] (both single- and multiple-scale). To this end, we compute $precision$-$recall$ curves, i.e. for a varying threshold, how many of the matched features are actually correct ($\epsilon_O < 55\%$) (*precision*), and how many of the corresponding features have actually been matched (*recall*). Matching is performed based on the nearest neighbour ratio threshold measure. Here we show results for a spatial scale change of 1.5 (fig. 4(g)) and for a rotation of 22 degrees (fig. 4(h)).

Our descriptor clearly outperforms the jet-descriptors. $3 \times 3 \times 3$ subdivisions are more discriminative than $2 \times 2 \times 2$, albeit at the cost of a higher dimensional descriptor. Even larger descriptors do not result in significant further improvements anymore. Rotation invariance is valuable when rotations are to be expected. However, when no rotations are present, it has a negative impact on the results due to the increased complexity and reduced discriminativity.

	drink	eat	explore	groom	sleep
drink	83.3	0.0	0.0	16.7	0.0
eat	1.9	88.9	9.3	0.0	0.0
explore	0.0	1.4	83.6	12.3	2.7
groom	0.0	6.7	6.7	86.7	0.0
sleep	0.0	0.0	0.0	0.0	100.0

(a)

	walking	running	jogging	waving	clapping	boxing
walking	91.7	8.3	0.0	0.0	0.0	0.0
running	5.6	72.2	22.2	0.0	0.0	0.0
jogging	5.6	16.7	77.8	0.0	0.0	0.0
waving	0.0	0.0	0.0	100.0	0.0	0.0
clapping	0.0	0.0	0.0	11.1	86.1	2.8
boxing	0.0	0.0	0.0	8.3	13.9	77.8

(b)

Fig. 5. Action recognition results. (a) Confusion matrix for the mouse behaviour dataset [6]. The overall recognition rate is 87.12%. (b) Confusion matrix for the KTH human action dataset [2] using all 4 scenarios for training and testing. The accuracy is 84.26%.

4 Applications

4.1 Action classification

Next, we test our detector and descriptor in the context of action classification. To this end, we extract spatio-temporal features over 5 octaves. A visual vocabulary is built based on the feature descriptors contained in the training set videos using k-means clustering. Then, a bag-of-words is computed for each video using a *weighted* approach, similar to the one proposed in [19], where each visual word t_k of the visual vocabulary $T = [t_1, \ldots, t_k, \ldots, t_K]$ is given the weight

$$t_k = \sum_{i=1}^{N} \sum_{j=1}^{M_i} \frac{1}{2^{i-1}} \tag{13}$$

with M_i the number of interest points that have visual word k as their i^{th} nearest neighbour. In practice, we use the $N = 4$ nearest neighbours. Finally, the histogram is normalized using the $L1$-norm. An SVM classifier [20] is trained with a χ^2-RBF-kernel using 5-fold cross-validation, followed by a single run over the joined training and validation sets. The results below show the classification performance on the test sets.

The mouse behaviour dataset [6] contains clips taken from seven 15-minute videos of the same mouse filmed in a vivarium while eating, drinking, exploring, grooming and sleeping. As the number of videos differs significantly between the 5 behaviours, weights inversely proportional to their occurrence are included while clustering the features of the training set. Roughly 37% of the clips of each action of the dataset have been designated as the test set, while the other clips are divided between training and validation set. The resulting confusion matrix, shown in figure 5(a), has an overall accuracy of 87.1% on the test set, compared to the 72% reported by Dollár *et al.* [6].

The KTH human action dataset [2] contains six types of human actions performed by 25 people in 4 different scenarios. Some detected interest points for two actions are shown in figure 1. We divide the dataset in training, validation and test set as given

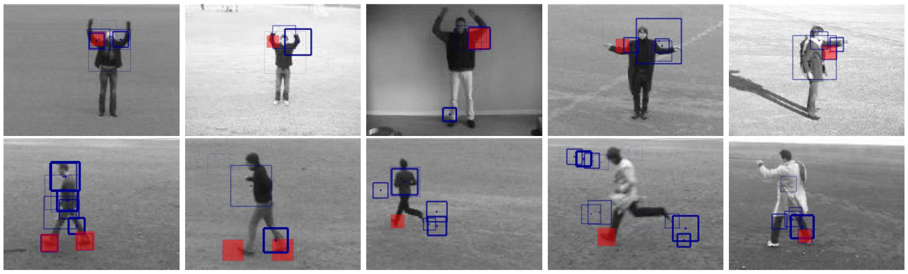


Fig. 6. In each row, features corresponding to a selected visual word are highlighted (in red). The same basic motions are recognized despite temporal and spatial variations. On the top row, the temporal invariance of our feature allows to cope with variable speeds of hand waving, yet also introduces some confusion with some instances of clapping and boxing. On the bottom row, the average temporal scale of the features varies from 8.9 for walking to 5.2 and 3.6 for jogging and running respectively. The same feature was also detected on the foot work by the boxing figure. (For visibility, only a very small number of detected features is shown.)

by [2]. As can be seen in the confusion matrix in figure 5(b), we obtain an accuracy of 84.26%, which is in line with state-of-the-art results reported by Schüldt *et al.* [2] (71.72%), Dollár *et al.* [6] (81.17%), Ke *et al.* [7] (62.96%), Niebles *et al.* [3] (81.50%) and Nowozin *et al.* [21] (87.04%). We outperform all but [21] who employed an extensive grid search in order to find the best model parameters. Further finetuning over all parameters could probably also improve our results further.

As expected, running and jogging turn out to be the most difficult to distinguish, followed by boxing and clapping. This may to some extent be explained by the temporal scale invariance of our features. On one hand, this temporal invariance brings robustness to actions performed at varying speed (as observed for instance in the waving action, where we get a 100% recognition accuracy). On the other hand, this increases the confusion between different actions sharing the same basic motions but performed at different speeds, such as the touch-down of the foot in walking, jogging and running (see also figure 6). By including the selected scales as part of the descriptor, this confusion can probably be reduced.

4.2 Synchronization between an original video and a mashup

Next, we demonstrate the use of our features in the context of synchronizing a video mashup with the original source material. *Video mashups* are video clips that have been created by combining material from different sources. Here, we focus on a 3 minute video clip ”Robocop vs. Neo” [5] which combines scenes from the movies ”Robocop” and ”The Matrix”. As original source material, we downloaded the corresponding scene of the movie ”The Matrix” [6]. Both video clips have different resolutions, are heavily compressed and contain aliasing artefacts due to ripping and recoding. Moreover,

[5] http://www.youtube.com/watch?v=UFou895WluU by AMDS Films

[6] http://www.youtube.com/v/T8fDJpid7gg

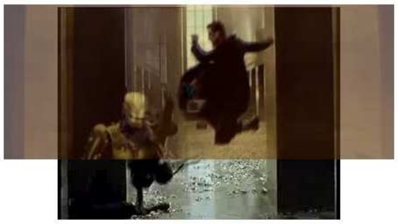


Fig. 7. Synchronization of a scene from the movie ”The Matrix” and a scene from a mashup between ”The Matrix” and ”Robocop”.

the mashup does not merely consist of concatenations of shots from both movies, but includes color changes, gamma corrections, scale changes, time-stretching, flipped & reversed shots as well as novel shots where elements from both movies are combined into new scenes (see fig. 7). As a result, global methods such as the correlation-based approach of [22] are bound to fail.

First, the interest point detector is run on the full videos. Then, shot cut detection is applied to split up the videos, after which approximate nearest neighbour matches between the features of each shot are computed. Finally, we check geometric consistency based on the random sampling scheme RANSAC, using a simple, linear model with 5 parameters (spatial and temporal translations and scale factors, i.e. d_x, d_y, d_t, $s_x = s_y$ and s_t).

An example of synchronized shots is shown in figure 7. Note the significant change in color and the different spatial resolution. Furthermore, additional elements have been added into the scene, while other details have been removed.

4.3 Computation time

Table 2 gives an overview of some of the video sequences we processed and the needed computation time. For low-resolution videos, such as the KTH human action dataset, detection and description can be done in realtime. Moreover, the needed computation time for feature detection is independent of the number of features. This is in contrast to HL-STIPs where the most time consuming step is the iterative procedure, which takes a time linear in the number of features.

Depending on the application, the number of temporal octaves to process can be adapted, and this further reduces the needed computation time. This is illustrated in the bottom two rows of table 2, where we give the difference in processing time and in the number of detected interest points between processing just one or five temporal octaves. Note that even with one octave we are still scale-invariant, albeit within a smaller range of scale changes.

datasets	#octaves temporal/spatial	spatial resolution	#frames	#interest points	detection & description time	fps
mouse behaviour dataset [6] (70min)	5/5	(avg.) 244 × 180	105650	784303	2h43min	10.8
human action dataset[2] (3h14min)	5/5	160 × 120	291756	715279	3h15min	24.9
”Robocop vs Neo” (2,7min)	5/5	125 × 313	4084	39237	6min	11.3
scenes from ”The Matrix” (1,7min)	5/5	240 × 320	2536	93970	25min	1.6
scenes from ”The Matrix” (1,7min)	1/5	240 × 320	2536	54448	5min	7.9

Table 2. General information on all video sequences used in this paper together with their processing times. A quad CPU Opteron 275 with 6GB of memory was used for processing. The buildup of the integral videos is included in the timings. In the second column, the numbers $'N/M'$ denote that the search space extended over N temporal and M spatial octaves. The minimum strength threshold for detection was set to 0.001 (with 1.0 the maximum response at a perfect spatio-temporal Gaussian blob).

5 Conclusion

In this paper, we have proposed a novel spatio-temporal interest point detector. First, we have shown that by using the determinant of the 3D Hessian matrix, it is possible to combine point localization and scale-selection in a direct way, therefore removing the need for an iterative scheme. Further, we have developed an implementation scheme using integral video, that allows for an efficient computation of scale-invariant spatio-temporal features. Our detector scores well in terms of repeatability and is on par with currently used spatio-temporal interest points. Finally, we have demonstrated its potential in the domain of action classification and video synchronization. Future work will aim at extending the current batch-mode approach towards a sliding-window framework, similar to Ke *et al.* [7]. Executables are available [7] to the community.

Acknowledgements This work is supported by the European IST Programme DIRAC Project FP6-0027787, the European IST Programme CLASS Project FP6-0027978, the Research Fund K.U.Leuven and the Flemish Fund for Scientific Research (FWO).

References

1. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: ICCV. Volume 2. (October 2003) 1470–1477
2. Schüldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local SVM approach. In: ICPR. (2004)
3. Niebles, J.C., Wang, H., Fei-Fei, L.: Unsupervised learning of human action categories using spatial-temporal words. In: BMVC, Edinburgh, U.-K (2006)
4. Laptev, I., Lindeberg, T.: Space-time interest points. In: ICCV, Nice, France (October 2003)
5. Laptev, I.: On space-time interest points. IJCV **64**(2) (2005) 107–123
6. Dollár, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. Visual Surveillance and Performance Evaluation of Tracking and Surveillance (2005) 65–72

[7] http://homes.esat.kuleuven.be/~gwillems/research/Hes-STIP

7. Ke, Y., Sukthankar, R., Hebert, M.: Efficient visual event detection using volumetric features. In: ICCV. Volume I. (2005) 166–173
8. Oikonomopoulos, A., Patras, I., Pantic, M.: Spatiotemporal salient points for visual recognition of human actions. IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics **36**(3) (2006) 710–719
9. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Gool, L.V.: A comparison of affine region detectors. IJCV **65**(1-2) (2005) 43–72
10. Kadir, T., Brady, M.: Scale, saliency and image description. IJCV **45**(2) (2001) 83–105
11. Wong, S.F., Cipolla, R.: Extracting spatiotemporal interest points using global information. In: ICCV, Rio de Janeiro, Brazil (2007) 1–8
12. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded-up robust features. In: ECCV, Graz, Austria (2006)
13. Lindeberg, T.: Feature detection with automatic scale selection. IJCV **30**(2) (1998) 77–116
14. Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. IJCV **60**(1) (2004) 63–86
15. Beaudet, P.: Rotationally invariant image operators. In: International Joint Conference on Pattern Recognition. (1978) 579–583
16. Laptev, I., Lindeberg, T.: Velocity adaptation of space-time interest points. In: ICPR, Cambridge, U.K (2004)
17. Willems, G., Tuytelaars, T., Van Gool, L.: An efficient dense and scale-invariant spatio-temporal interest point detector. Technical Report KUL/ESAT/PSI/0802, K.U. Leuven (2008)
18. Laptev, I., Lindeberg, T.: Local descriptors for spatio-temporal recognition. In: Int. Workshop on Spatial Coherence for Visual Motion Analysis
19. Jiang, Y.G., Ngo, C.W., Yang, J.: Towards optimal bag-of-features for object categorization and semantic video retrieval. In: CIVR. (2007) 494–501
20. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. (2001) Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.
21. Nowozin, S., Bakir, G., Tsuda, K.: Discriminative subsequence mining for action classification. (2007) 1919–1923
22. Yan, J., Pollefeys, M.: Video synchronization via space-time interest point distribution. In: Advanced Concepts for Intelligent Vision Systems, ACIVS '04. (2004)