Spatial motion patterns: action models from semi-dense trajectories

Thanh Phuong Nguyen, Antoine Manzanera, Matthieu Garrigues, Ngoc-Son Vu

International Journal of Pattern Recognition and Artificial Intelligence

Spatial Motion Patterns: Action Models from semi-dense Trajectories

Thanh Phuong Nguyen,* Antoine Manzanera, Matthieu Garrigues
ENSTA-ParisTech, 828, Boulevard des Maréchaux, 91762 Palaiseau, France
{*thanh-phuong.nguyen, antoine.manzanera, matthieu.garrigues*}*@ensta-paristech.fr*

Ngoc-Son Vu
ETIS-ENSEA, UCP, CNRS, 6 Avenue du Ponceau, 95014 Cergy, France
son.vu@ensea.fr

A new action model is proposed, by revisiting local binary patterns for dynamic texture models, applied on trajectory beams calculated on the video. The use of semi dense trajectory field allows to dramatically reduce the computation support to essential motion information, while maintaining a large amount of data to ensure robustness of statistical bag of features action models. A new binary pattern, called Spatial Motion Pattern (SMP) is proposed, which captures self similarity of velocity around each tracked point(particle), along its trajectory. This operator highlights the geometric shape of rigid parts of moving objects in a video sequence. SMPs are combined with basic velocity information to form the local action primitives. Then, a global representation of a space $\times$ time video block is provided by using hierarchical blockwise histograms, which allows to efficiently represent the action as a whole, while preserving a certain level of spatiotemporal relation between the action primitives. Inheriting from the efficiency and the invariance properties of both the semi dense tracker *Video extruder* and the LBP based representations, the method is designed for the fast computation of action descriptors in unconstrained videos. For improving both robustness and computation time in the case of high definition video, we also present an enhanced version of the semi dense tracker based on the so called *super particles*, which reduces the number of trajectories while improving their length, reliability and spatial distribution.

Keywords: action recognition, semi dense trajectory beam, local binary pattern, dynamic texture,...

1. Introduction

In the last decades, action recognition in videos has become a very active domain of computer vision research. To face the increasingly rapid proliferation of video contents, the design of a rapid and reliable method to automatically exploit videos is now a crucial challenge. The recognition of activities occurring in videos is a key problem in many applications, such as video surveillance, video annotation and retrieval, video summarization, human computer interaction and so on.

According to [1], the design of automated activity recognition systems faces three major problems. The first one is intra- and inter-class variations; for example

*Corresponding author

speed or stride length may vary much from one gait to the other, whereas running and jogging are different but similar actions. The second one comes from the huge variety of environments and recording settings, such as lighting conditions, viewpoints, backgrounds, camera motions, etc. The third one is the difficulty to obtain relevant training data and to label them. For these reasons, and in spite of many existing methods, designing a reliable action recognition system in real conditions is still an open problem.

One of the most critical parts of action recognition systems is the design and computation of the action model: what information should be extracted from the video, and how. Many methods have been proposed for action representation (see [2] for a comprehensive survey). One first way to classify them is according to the data used to calculate the features, which can be (1) Space-time appearance [3–7], (2) Apparent motion [8, 9], or (3) Body silhouette [10, 11].

Another way to classify the existing action models is to distinguish global *vs* local features. Global features are obtained using top-down strategy to encode the visual observation as a whole. Such approach generally represents an action by characterizing a region of interest. It often requires the detection of human body in videos. Due to this pre-processing step that is usually based on background subtraction or object tracking, these features are more sensitive to noise, occlusions or viewpoint change. If these factors are well controlled, the global features are a powerful representation because they encode most of the information. Therefore, they work well in controlled environments such as the KTH dataset [12] but are less effective in more realistic environments such as UCF Youtube [13] dataset. The global features are often derived from silhouettes, edges or optical flow. Bobick and Davis introduced MHI (Motion History Images) and MEI (Motion Energy Images) [10] for encoding evolution of human body movements in temporal and spatial dimensions. Blank et al. [11] described human actions as 3d shapes generated by the silhouettes in the spatio-temporal space. The actions are then modelled by Poisson equations. Efros et al. [8] represent action by global patterns produced by optical flow fields on a figure-centric spatio temporal volume for each person in a video.

On the other hand, the local representations rather follow a bottom-up strategy. They are made from a collection of local patterns, usually descriptors calculated on a set of spatio-temporal interest points. Then the collection of patterns is reduced using statistical techniques such as bag of features, to construct the action descriptor. The main advantage of these approaches is that they are more robust to noise and partial occlusions than the global representations. In addition, they don't require a segmentation step like background subtraction or human tracking. On the other hand, their results depend more strongly on the choice of input feature, and the performance of statistical classification depends on the amount of extracted local patterns. Let us give some important examples of local feature approaches.

Many local spatio-temporal features are designed by extending classical space detectors and descriptors, such as SIFT [14], SURF [15], or HOG [16] to 3d space-

time. Laptev [3] detected space-time interest points in videos extending Harris corner criteria from 2d images to 3d. Similarly, Dollár [4] extracted spatio temporal keypoints in the energy map referred to as cuboids by performing symmetric temporal Gabor filtering. It avoids the problem of sparse corner detection reported in [3]. A 3d extension of SURF descriptor is given in [5] using 3d Haar wavelets. Klaser [6] introduced HOG3D by using polyhedral structures for quantization of the 3d spatio-temporal edge orientations. Inspired from HOG, Histogram of Oriented Optical Flow (HOOF) [9] that was first proposed for human detection, uses the optical flow instead of the gradient as basis of the orientation distribution. Willems [7] performed an extension of the Hessian saliency measure to detect dense and scale-invariant spatio-temporal interest points. Chakraborty [17] proposed a surround suppression of detected interest points combined with spatial and temporal constraints to be robust with respect to camera motion and background cluster. An interesting approach is to consider the action as a texture pattern, and to apply dynamic or static texture based methods to action modelling and recognition. Thanks to the effective properties of Local Binary Patterns (LBP) for texture representation, several LBP-based methods have also been proposed for action recognition. The existing LBP-based methods will be reviewed in the next section.

In this paper, we present a new action model which can be seen as a hybrid solution between optical flow methods and dynamic texture based approaches. The motion is locally represented using a binary pattern, whose support is a space-time neighbourhood, located along a trajectory obtained by point tracking. We propose a new self-similarity operator to capture spatial relations in a trajectory beam, by representing the similarity of motion between the tracked point along its trajectory, and its neighbourhood. The semi-dense point tracker computes the displacement of many points in real time, then we apply self-similarity operator on appearance information to represent the motion information of a larger zone surrounding the trajectory.

The remainder is organized as follows. Section 2 presents the background knowledge on LBP based (dynamic) texture representations. Section 3 presents the algorithm used to extract the trajectories. Section 4 combines trajectories and LBPs to form the Spatial Motion Patterns (SMP), which are the descriptors used in action classification. Section 5 presents and discusses the evaluation of the SMP for action recognition on three classic datasets. The last section presents the conclusion and perspectives of this work.

2. LBP based representations

2.1. *Brief review of LBP*

Local Binary Patterns [18] were introduced by Ojala et al. Their idea is to capture the local structures of texture images using binary patterns obtained by comparing a pixel value with its surrounding neighbours. The LBP encoding of one pixel can

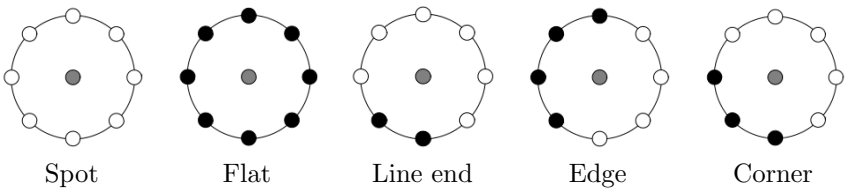

Fig. 1. Texture primitives corresponding to Uniform LBPs [18].

be defined as follows:

$$\mathrm{LBP}_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c).2^p, s(x) = \begin{cases} 1, x \geq 0 \\ 0, \text{otherwise} \end{cases}$$

where g_c is the value of the pixel $\mathbf{c}$ and $\{g_p\}_{0 \leq p < P}$ are the values of the P neighbours evenly located on a circle of radius R and centre $\mathbf{c}$. The values of neighbours can be obtained by direct sampling or estimated by interpolation.

Ojala et al. observed that most of the patterns in natural images have 0 or 2 bitwise transitions (i.e. 0-1 or 1-0 change occurring in a circular scan). Such patterns are called uniform (LBP^{u2}) and defined by $U(\mathrm{LBP}_{P,R}) \leq 2$, where:

$$U(\mathrm{LBP}_{P,R}) = \sum_{p=1}^{P} |s(g_p - g_c) - s(g_{p-1} - g_c)|,$$

with $g_P = g_0$.

The LBP operator has two important properties: it is invariant to monotonic gray scale changes, and its complexity is very low. Initially proposed for texture modelling and recognition, LBP-based approaches have proved suitable for many other applications, including face recognition, object modelling and action recognition. The uniform pattern coding ($\mathrm{LBP}^{u2}_{P,R}$, which corresponds to ignoring the non uniform patterns) is widely used in real applications because it reduces significantly the length of feature vectors while capturing important texture primitives: Fig. 1 displays the local geometry classification induced by $\mathrm{LBP}^{u2}_{P,R}$.

2.2. *Dynamic texture representation using LBP-based methods*

An intuitive extension of LBP is to represent dynamic texture in a 2d+t space, and applying the notion of self-similarity to the spatio-temporal domain. We recall hereafter two spatio-temporal LBP operators for image sequences.

2.2.1. *VLBP*

Volumetric LBP [19] is a direct extension of [18] for image sequence. Zhao and Pietikäinen defined dynamic texture at voxel $\mathbf{c}(x, y, t)$ considering $3P$ neighbours located on 3 circles of radius R, and centres $\mathbf{c}_1(x, y, t - \delta t)$, $\mathbf{c}(x, y, t)$ and $\mathbf{c}_2(x, y, t +$

δt), where δt is a time interval. Each circle provides a LBP binary code of length P. Similarly, comparing the values of the 3 centres $\mathbf{c}$, $\mathbf{c}_1$ and $\mathbf{c}_2$ leads to a 2-bit code. Finally, the VLBP, made by concatenating the 4 patterns, is a $(3P + 2)$-bit code.

2.2.2. *LBP-TOP*

VLBP produces very long codewords when P is large, and only takes into account one time interval δt. LBP-Three Orthogonal Patterns [20] were also introduced by Zhao and Pietikäinen to address this problem. LBP-TOP pattern at pixel $\mathbf{c}$ is made by the LBPs on 3 circles from the 3 orthogonal planes (x, y), (x, t) and (y, t) which intersect at $\mathbf{c}$. This approach provides 3 shorter codewords which, unlike VLBP, are gathered in separated histograms.

2.3. *LBP-based methods for action recognition*

Kellokumpu et al. [21] used dynamic texture operator (LBP-TOP) to represent human movements. They also presented another approach [22] using classical LBPs on temporal templates (MEI and MHI images [10], which are gray level images representing motion information). In the two methods the features were used as observations of a Hidden Markov Model which actually represented the action. Mattivi and Shao [23] presented a different method using LBP-TOP to describe cuboids detected by Dollár's feature detector. Nanni et al. [24] improved LBP-TOP using ternary units in the encoding step. Yeffet and Wolf proposed LTP (Local Trinary Patterns) [25] that combines the effective description of LBP with the adaptivity and appearance invariance of patch matching methods. They capture the motion effect on the local structure of self-similarities considering 3 neighbourhood circles at a spatial position and different instants. Kliper-Gross et al. developed this idea by capturing local changes in motion directions with Motion Interchange Patterns (MIP) [26].

3. Motion Representation from a Beam of Trajectories

Trajectories are compact and rich information source, and a natural support for the representation of actions. They have been used before for action recognition [27]. However, to obtain reliable trajectories, the spatial information is often dramatically reduced to a small number of keypoints, and then it may be hazardous to compute statistics on the set of trajectories. In this work we use *Video Extruder* [28], a semi dense point tracking method (see also Fig. 2) which is a trade-off between long term tracking and dense optical flow, and allows the tracking of a high number of weak keypoints in a video in real time, thanks to its high degree of parallelism.

To optimise the recognition, it is important to have a flexible compromise between the number of trajectories and their reliability. To this end, we introduce in this paper the new concept of super particles, which aggregate the information worn by a set of particles forming different trajectories. Super particles lead to a

Table 1. Performances of *Video extruder* semi dense tracker. # p is the number of particles, Mpix/s (resp. fps) is the computation frequency, in Megapixels (resp. in frames) per second, cpp is the number of cycles per particle.

Architecture	Resolution	# p	Mpix/s (fps)	cpp
GPU Geforce GTX 460 1.35GHz	640×480	8 500	50 (166)	957
CPU quad-core I5 2500k 3.3GHz	640×480	8 500	46 (152)	2 550
ARM dual-core STE U8500 1GHz	320×240	3 000	0.84 (11)	30 300
ARM single-core IMX.53 1GHz	720×288	2 000	2.07 (10)	50 000

better distribution of motion information, reduce the influence of noise and provide longer trajectories.

We give in this section a brief overview of the tracker, explain how the super particles are formed and tracked, and finally compare them with the original Video Extruder.

3.1. *Overview of Video Extruder*

To improve robustness to large motion (including motion of the camera itself), Video Extruder follows a coarse to fine pyramidal scheme: It starts by estimating the motion at the coarsest resolution level and iterates to the finest resolution level. For each level l of the pyramid, Video Extruder performs the following steps :

- **Particle detection:** A weakly salient point detector [28], designed to detect as many trackable points as possible, extracts a semi dense field of key points (particles).
- **Particle matching and tracking:** The matcher finds the new position of each particle in the current frame. Using velocities estimated at level $l + 1$ and the previous velocity of the particle, it predicts the position, and then minimises the distance between descriptors using a gradient descent (the descriptor is a vector of 16 pixel values sampled at two different scales).
- **Error filtering:** A filtering step finally remove particles that diverge from their neighbours.

Table 1 shows how the tracker performs on a GPU, a CPU, and two low-powered ARM processors.

3.2. *Super particle segmentation and tracking*

We present the algorithm to segment and track groups of analogous particles. Such groups are called super particles in the following. From the semi dense field of keypoints, a super particle is a set of particles that are close in both image and motion spaces.

Initialisation: The image is split in a grid of $D \times D$ pixel cells, and assume that

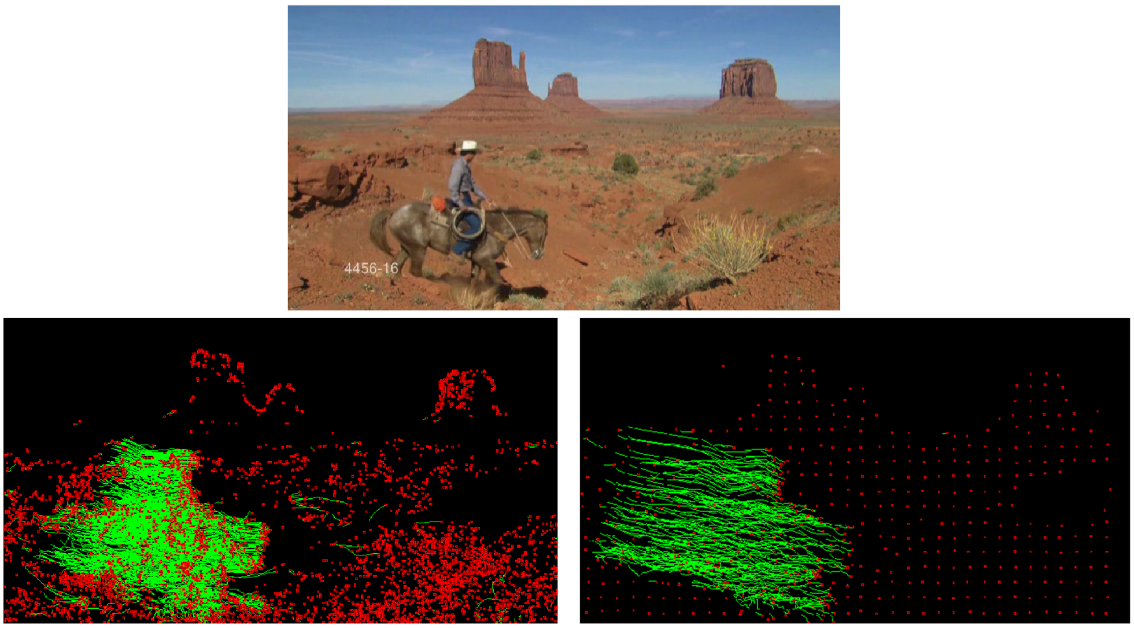

Fig. 2. Comparison of the particles and the super particles.

every particle of cell $\mathbf{x}$ belongs to the super particle $P^{\mathbf{x}}$. The position of $P^{\mathbf{x}}$ is initialised to the centre of its cell.

The super particles are updated as follows: Let m_x and m_y be the median horizontal (resp. vertical) displacement of the particles that belong to $P^{\mathbf{x}}$. We compute the new position of $P^{\mathbf{x}}$ as: $P^{\mathbf{x}}_t = P^{\mathbf{x}}_{t-1} + (m_x, m_y)^\intercal$.

To add the new particles and discard the divergent particles from the super particle, we apply the following: If the distance between an orphaned particle and $P^{\mathbf{x}}$ is smaller than D pixels, we add it to $P^{\mathbf{x}}$. If the distance between a particle and $P^{\mathbf{x}}$ is bigger that $2D$, it is removed from the super particle and set as orphaned. Finally, to enhance the robustness of super particles, they are discarded if they contain less than θ particles.

This algorithm can be seen as a simplified mobile object segmentation and tracking. It does not segment objects, but aggregates small groups of particles moving with similar motion.

Super particles have several advantages over simple particles. The trajectories of these groups are more robust and then track motion over a longer period of time. Furthermore, they are better distributed over the image domain, lowering the over-representation of highly textured areas in the statistics. Figure 2 and Table 2 compares the trajectories of the particles and the super particles on the same video. In this example $D = 10$ and $\theta = 5$. It is also the configuration chosen for experimentation in Section 5.4.

Table 2. Comparison of moving particles and super particles. An entity is classified as moving if it moved more than 100 pixels over its entire life. The average trajectory lifetime represents the average number of frames the particles were tracked.

	Particles	Super particles
Number of entities	2 289	178
Average trajectory lifetime (in frames)	30.68	48.53

4. Action Descriptor using Spatial Motion Patterns

This section details the construction of our action descriptor. The input data is the semi-dense trajectory beam described in Section 3, and no appearance information is explicitely used. A classic approach to action description using velocity information is to consider histogram of (orientation of) optical flow (HOOF). This method is simple and computationnally efficient, but suffers from limited discrimination capability, since it neglects the spatio-temporal relations between moving points. One partial solution is to compute the histograms in different sub-volumes defined by a spatio-temporal grid. The descriptor defined in this section aims at addressing more finely this problem. It is constructed in a way to exploit motion information at different context levels:

- *Point level:* The velocity vector is provided by the semi-dense tracker, at each frame and for every particle (or super particle).
- *Local spatio-temporal level:* Spatial Motion Pattern (SMP) is defined as a self-similarity operator for capturing the motion similarity between every (super) particle and its surrounding points.
- *Regional to global spatio-temporal level:* A hierarchical bag of feature (BoF) histogram vector is built to describe the action at different spatiotemporal scales.

These levels are detailed in the following subsections, then we discuss the properties of this action model, compared with other descriptors from related works.

4.1. *Point level*

The velocity of particles from frame to frame is provided by the semi-dense tracker. Let $\overrightarrow{p_t}$ be the 2d displacement of the particle between frames t and $t+\delta$. The first part of the encoding is simply a dartboard quantisation of vector $\overrightarrow{p_t}$ (see Fig. 3). In our implementation, we used intervals of $\pi/6$ for the angles and 2 pixels for the norm (the last interval being $[6, +\infty[$), resulting in 12 bins for direction angle, 4 bins for norm.

This code corresponds to motion information at the finest level, i.e. point context. However, its information range is much wider than the pixel, since a particle represent a certain spatial structure, and in the case of super particles, the range is even wider, since it depends on D, the space radius of super particles.

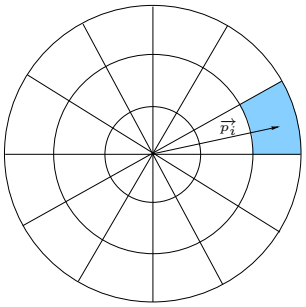

Fig. 3. Dartboard quantisation of the motion vector.

4.2. *Local spatio-temporal level*

At the local spatio-temporal level, we use an LBP-based representation to capture the relations between a point and its neighbours. The idea is to capture the inter-trajectory relations among a beam of trajectories. We combine the LBP-based self-similarity operator [18] and the appearance invariance of patch matching method inspired by [25]. This operator, called Spatial Motion Pattern (SMP), is presented below.

Consider a (super) particle $\mathbf{p}$ that moves from position P_1 at frame t to position P_2 at frame $t+\delta$, provided by the semi dense tracker. The similarity of motion between this particle and its surrounding neighbours is assessed by considering the $2 \times n$ patches sampled on the circles of radius r centred at P_1 and P_2 in their corresponding frames (see Fig. 4). Every index $i \in \{0, \ldots, n-1\}$ represents a neighbour, which is encoded by 0 if its motion is similar to the motion of the centre particle, and by 1 otherwise. Because semi-dense point tracking is applied instead of dense optical flow, the velocity information is not available everywhere. Following [25], SSD (sum of square difference) score is used as similarity measure to check the consistency of motion.

Let $\{\Delta(\mathbf{p},t)_i\}_{i=0}^{n-1}$ be the set of n patches surrounding particle $\mathbf{p}$ on the circle of radius r, at frame t. The corresponding $\text{SMP}_{n,r}$ codeword $(b_0, b_1, \ldots, b_{n-1})$ is calculated as follows:

$$b_i = \begin{cases} 1 & \text{If } SSD\big(\Delta(\mathbf{p},t)_i, \Delta(\mathbf{p},t+\delta)_i\big) \geq \tau \\ 0 & \text{otherwise} \end{cases},$$

where δ is the time interval between two frames, τ is the SSD threshold. Overall, there are 2^n different possible values for a SMP, and only $n(n-1)+3$ different values for SMP^{u2}, i.e. if non uniform patterns are discarded and grouped in a unique symbol [18].

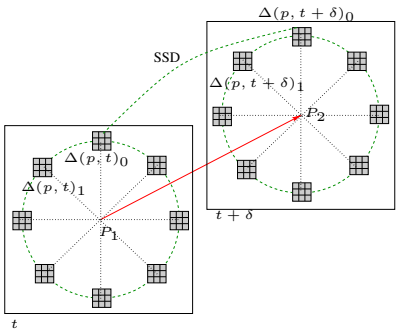

Fig. 4. The SMP descriptor is calculated for each tracked particle along its trajectory. The consistency of motion in every direction is checked by computing the SSD between the corresponding image patches.

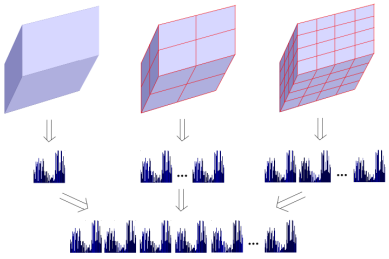

Fig. 5. Action modelling by SMP histogram concatenation.

4.3. *Regional to global spatio-temporal level*

To represent the action as a whole, while preserving the main trends of the spatio-temporal relations between its different components, a hierarchical bag of feature (BoF) model [29] is used.

An action is represented by histograms of codewords formed by the two previous primitives (motion code and spatial motion patterns) on spatio-temporal volumes. Note that unlike many BoF approaches, no vector quantisation is performed, and the number of codewords simply corresponds to the number of different motion code multiplied by the number of different SMP^{u2}, i.e. $48 \times (n(n-1)+3)$. In the hierarchical approach, the considered volumes can be the entire sequence, or a set of sub-sequences defined by a spatio-temporal grid. All histograms are concatenated into one vector that is then normalised to form the action descriptor. Fig. 5 shows an exemple of hierarchical BoF descriptor constructed using three different grids: $1 \times 1 \times 1$, $2 \times 2 \times 2$ and $4 \times 4 \times 4$.

4.4. *Properties of Spatial Motion Patterns*

Spatial Motion Patterns have attractive properties, most of which are inherited from [18, 25], :

- *Efficient computation.* They use SSD scores on small image patches, calculated on tracked keypoints only, thus avoiding many irrelevant calculations.
- *Appearance invariance.* This property is due to: (1) the LBP based encoding and (2) the input data itself, which only relates to the trajectory, not to the appearance.
- *Robustness against complex background.* Unlike many methods, SMP works better when the background is more complex. Indeed, in that case the SSD is more significant, and then the SMP will also better describe the local

Fig. 6. SMP^{u2} configurations allow to determine the shape of the rigid parts of the moving object around the keypoints (in red points). In the neighbouring circles, image patches in green (resp. blue) indicates that they belong to the same rigid part of the moving object as the keypoint (resp. another rigid part or the background).

motion information.

4.4.1. *SMP uniform patterns and primitive actions*

An important property of SMPs is the interpretation of the corresponding uniform patterns. In an analog manner as LBP^{u2} (see Fig. 1), SMP uniform patterns (SMP^{u2}) capture local action primitives. They characterise the motion between foreground objects and the background in videos, and more generally, between two rigid parts of a moving object. The interpretation of SMP^{u2} as action primitives can be done as follows (see Fig. 6, and refer to Fig. 1 for the name of primitives):

- **Spot:** Small foreground object moving on the background.
- **Flat:** Rigid part of an object (or background).
- **Line end:** End of a thin rigid part of a moving object.
- **Edge:** Frontier object/background, or between two parts of an object.
- **Corner:** Corner of a rigid part of a moving object.

4.4.2. *Comparison with Local Trinary Patterns*

Although inspired by them, the Spatial Motion Patterns (SMP) differ from Local Trinary Patterns (LTP) [25] in several aspects. Let us first recall the LTP principles. They use sum of squared differences (SSD) between patches centred at different space and time locations in order to capture the motion effect on the local structures. Let $\text{SSD}^{\Delta_{\mathbf{x}}}_{\Delta_t}$ be the SSD between the patch centred at pixel $\mathbf{x}$ at frame t and the patch centred at pixel $\mathbf{x}+\Delta_{\mathbf{x}}$ at frame $t+\Delta_t$. One ternary code $\{-1,0,1\}$ is obtained for each shift direction $\Delta_{\mathbf{x}}$, by comparing $\text{SSD}^{\Delta_{\mathbf{x}}}_{-\Delta_t}$ and $\text{SSD}^{\Delta_{\mathbf{x}}}_{+\Delta_t}$. Figure 7 illustrates this principle.

We list hereunder the main differences between SMP and LTP:

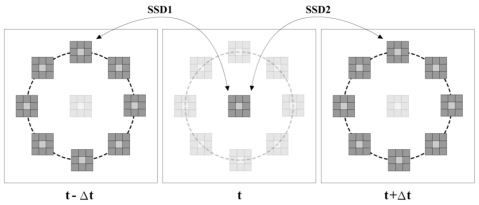

Fig. 7. Local Trinary Patterns [25].

- *Encoding process.* Unlike LTP, SMP use only 2 bits. The encoding of LTP is done by comparing SSD scores between neighbouring patches of past and future frames, and the centre patch of the middle frame. For SMP, SSD scores between two corresponding patches are calculated in two consecutive frames and the binary code is deduced by thresholding.
- *Neighbouring configuration.* LTP used three circles centred at the same position in 2D space. For SMP, the two neighbouring circles are centred at the tracked position of each particle. Then these circles are not always located at the same position.
- *Interpretation.* LTP aims to represent motion information at a given position, whereas for SMP, the motion information is already known, the SMP is interpreted as a local disparity map of velocities around each trajectory. It describes the relative similarity between the small patch in the middle frame and its neighbouring patches in the previous and next frames.

5. Experimentation on Human Action Classification

To perform action classification, we apply SVM classifier to action descriptors (made by concatenation of different SMP histograms). We choose the method of Vedaldi et al. [30] which approximates a large scale support vector machines using an explicit feature map for the additive class of kernels. Generally, it is much faster than classic kernel based SVMs and it can be used in large scale problems. We evaluate our descriptor on several well-known datasets. The first one (KTH) [12] is a classic dataset, used to evaluate many action recognition methods. The second and third ones are UCF Youtube [13] and UCF Sport [31], which are more realistic and challenging datasets.

5.1. *Parameter settings*

5.1.1. *SMP configuration*

There are several parameters concerning the construction of SMP. Like [25], we compute SSD score on image patch of size 3×3 with threshold $\tau = 1000$ that represents 0.17% of the maximal value of SSD. The time interval δ is set to 1.

In addition, because every tracked keypoint already represents a certain spatial

structure, the radius of SMP must be sufficiently large to better capture the geometric shape of rigid parts of moving object surrounding the keypoints. This differs from the typical LBP operator that uses small radius of spatial support (from 1 to 3) in order to capture "micro" structure of textured images. In our implementation, we consider 16 neighbours sampled on a circle of radius 9, as this configuration proved to obtain good results. To reduce the feature vector size, only uniform patterns ($\text{SMP}^{u2}_{16,9}$) were considered. Due to the intrinsic anisotropy of motion in real scenes, the rotation invariant uniform patterns ($riu2$) were not considered.

5.1.2. *BoF configuration*

Bag of Feature is a popular approach to build descriptor from local features in videos. Lazebnik et al. [29] showed that a hierarchical BoF still enhances the performance by capturing spatial relations in the distribution of codewords in subvolumes at different scales. They used 3 different grids (scales): $1 \times 1 \times 1$, $2 \times 2 \times 2$ and $4 \times 4 \times 4$. To reduce the dimensionality of feature vector, we used the 3 spatiotemporal grids: $1 \times 1 \times 1$, $2 \times 2 \times 2$ and $3 \times 3 \times 3$ to construct the histograms of codewords.

5.2. *Experiments on KTH dataset*

The dataset contains 25 people for 6 actions (running, walking, jogging, boxing, hand clapping and hand waving) in 4 different scenarios (indoors, outdoors, outdoors with scale change and outdoors with different clothes). It contains 599 [a] videos, of which 399 are used for training, and the rest for testing. As designed by [12], the test set contains the actions of 9 people, and the training set corresponds to the 16 remaining persons. Table 3 shows the confusion matrix obtained by our method on the KTH dataset. The ground truth is read by row. The average recognition rate is 93.33 % which is comparable to the state-of-the-art of LBP-based approaches (see Table 4). The main error factor comes from confusion between jogging and running, which is a typical problem in reported methods. We remark that unlike [21, 22] that work on segmented box, our results are obtained directly on unsegmented videos. Applying the same pre-processing step would probably improve our result.

[a]It should contain 600 videos but one is missing
[b]Ours 1: with particle trajectories; Ours 2: with super particle trajectories.
[c]Ours 1: with particle trajectories; Ours 2: with super particle trajectories.
[d]This result is obtained combining several local descriptors: TD, HOG, HOF, MBH. Using one descriptor, the result may vary from 58.2 % to 72.9% on KLT trajectories, from 67.2 % to 83.9 % on dense trajectories.

Table 3. Confusion matrix on KTH dataset.

	Box.	Clap.	Wav.	Jog.	Run.	Walk.
Box.	97.5	2.5	0	0	0	0
Clap.	2.5	97.5	0	0	0	0
Wav.	2.5	0	97.5	0	0	0
Jog.	0	0	0	95.0	0	5.0
Run.	0	0	0	12.5	82.5	5.0
Walk.	0	0	0	10.0	0	90.0

Table 4. Comparison on KTH dataset.

Method [b]	Result	Method	Result
Ours 1	93.33	[25]	90.17
Ours 2	92.08	[26]	93.0
[23]	88.38	[21]	93.8
[32]	82.36	[22]	90.8
[27]	94.2	[33]	97.1

Table 5. Comparison on UCF Youtube.

Method [c]	Result	Method	Result	Method	Result
Ours 1	72.07	[34]	64	[13]	71.2
Ours 2	70.85	[35]	64	[27]	84.2[d]

Table 6. Confusion matrix on UCF Youtube dataset.

	Walk.	Volley.	Tramp.	Tennis	Swing	Soccer.	Horse.	Golf.	Div.	Biking	Bask.
Walk.	38.64	0	6.82	2.27	6.82	4.55	18.18	4.55	0	18.18	0
Volley.	0	74.36	0	2.56	7.69	2.56	0	2.56	0	0	10.26
Tramp.	0	2.33	88.37	0	0	4.65	0	0	0	4.65	0
Tennis	1.59	0	1.59	79.37	0	0	4.76	3.17	3.17	4.76	1.59
Swing	5.36	0	8.93	0	78.57	0	0	0	0	7.14	0
Soccer.	5.56	3.7	1.85	16.67	0	55.56	1.85	7.41	0	0	7.41
Horse.	1.29	0	0	4.84	0	0	70.97	0	0	9.68	1.61
Golf.	1.85	0	0	0	0	0	0	92.59	3.7	0	1.85
Diving	5	1.67	1.67	1.67	0	0	0	1.67	88.33	0	0
Biking	0	0	6.25	0	0	12.5	6.25	2.08	0	72.92	0
Bask.	0	2.34	4.26	6.38	2.13	0	4.26	2.13	4.26	0	53.19

5.3. *Experiments on UCF Youtube dataset*

The UCF Youtube dataset records 11 categories (basketball shooting, cycling, diving, golf swinging, horse back riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking and walking with a dog), and contains 1 600 video sequences. Each category is divided into 25 groups sharing common appearance properties (actors, background, or other). It is much more challenging than KTH because of its large variability in terms of viewpoints, backgrounds and camera motions. Following the experimental protocol proposed by the authors [13], we used 9 groups out of the 25 as test and the 16 remaining groups as training data. Table 5 presents our results compared to other recent methods while Table 6 shows the confusion matrix of our method (Ours 1) on this dataset. The worst classification rate corresponds to confusions of "walking with dog" action with "biking" and "horse riding" actions. Our mean recognition rate on UCF Youtube dataset is 72.07 %, which is comparable to recent methods.

5.4. *Experiments on UCF Sport*

The UCF Sport dataset [31] contains a set of action clips from different sport videos, including 9 categories (diving, golf swing, kicking, lifting, riding, running, skate-

Table 7. Comparison on UCF Sport dataset.

Method	Result	Remark
Ours	81.3	Super particle trajectories
Ours	80.7	Particle trajectories
Rodriguez et al. [31]	69.2	
Hessian [7] + ESURF [36]	77.3	From [37]
Harris3D [3] + HOG/HOF [36]	78.1	From [37]
Yeffet and Wolf [25]	79.1	
Hessian [7] + HOG/HOF [36]	79.3	From [37]
Dense + HOF [36]	82.6	From [37]
Cuboids [4] + HOG3D [6]	82.9	From [37]
Wang et al. [27]	88.2 [e]	

boading, swing and walking). Testing is based on Leave One Out cross validation, selecting one video at a time for testing, and the others for training. Following [6], a horizontally flipped version of each video was added to increase the number of training samples. UCF Sport is a challenging dataset, because it contains a wide range of scenes and viewpoints. Compared with UCF Youtube, the resolution is higher and the actions are shot from closer. This results in a much larger number of trajectories, which implies higher computation time. Then, we chose this dataset to evaluate the super particle based enhanced trajectories.

Table 7 presents the results of our descriptors on this dataset using the two types of trajectories as input (particles or super particles), compared with existing methods. Working with super particles gives a comparable result as original particles. However the super particule strategy accelerates dramatically the computation time of the descriptor. Indeed, the tracking of particles or super particles takes approximately the same time, whereas the computation cost of the action descriptor essentially depends on the total length of tracked trajectories. As a typical example, it can be deduced from the figures of Tab. 2 that the super particle approach should imply more than 8 times fewer SMPs to compute.

5.5. *Discussions*

From the above experiments on three datasets, we can do the following remarks.

- For low resolution videos like KTH or UCF Youtube datasets, the super particles do not improve the performance. This is due to the loss of activity information when the number of trajectories is limited

[e]This result is obtained combining several local descriptors: trajectory, HOG, HOF, MBH. Using one descriptor, the result may vary from 72.7 % to 80.2 % on KLT trajectories and from 75.2 % to 84.8 % on dense trajectories.

- For high resolution videos like UCF Sport dataset, the super particles are a good way to accelerate the computation of descriptors without reducing the discrimination capability of the method.
- The proposed method outperforms LBP-based methods and is comparable to other recent methods.
- Two trajectory-based methods [27, 33] have better results than ours. However their computation time is much higher, due to their combination of a large number of different descriptors such as TD, MBH, HOF and HOG, and, in the case of [27], their use of dense trajectories, whose computation cost is higher than (semi-dense) VideoExtruder.

The comparison with [27, 33] leads to different remarks. The performance of trajectory-based methods depends on the type of trajectories. For example, changing KLT for dense trajectories improves for more than 6% on UCF Sport. Also, a combination of different descriptors can improve the performance. So experimenting on different kinds of trajectories and combining with other descriptors can be considered in future works.

Due to the difficulty in re-implementing existing methods, we did not quantitatively compare the computation time for the different methods. However, the efficiency of the proposed method can be justified by its components:

- The extraction of trajectories is very fast thanks to VideoExtruder
- The computation of SMP is simple like other LBP-based variants. Moreover, SMP code is not calculated at each voxel of the video. It is only calculated along the tracked trajectories, whose number can be adjusted using the super particle framework.

6. Conclusions

We have presented a new method for action recognition based on semi-dense trajectory beams and a LBP based local motion pattern, which captures spatial relations of moving parts, along their trajectories. It inherits good properties of invariance and computational efficiency from *Video Extruder* and LBPs, and it is designed to work in unconstrained videos with complex background. To enhance the robustness of the trajectories, and to reduce the computation time of the action descriptors, the new concept of super particles was introduced, leading to fewer trajectories, but longer, more stable and better space distributed. In the future, we are interested in several perspectives related to this method, such as multi-scale SMPs, and improving the results in the case of moving backgrounds.

Acknowledgment

A preliminary version of this article was presented in [38]. This work was supported by the French Ministry of Economy (DGCIS), as part of a European ITEA2 project.

We thank the anonymous referees for their constructive comments and suggestions.

References

1. Poppe, R.: A survey on vision-based human action recognition. Image Vision Comput. **28** (2010) 976–990
2. Aggarwal, J., Ryoo, M.: Human activity analysis: A review. ACM Comput. Surv. **43** (2011) 16:1–16:43
3. Laptev, I.: On space-time interest points. IJCV **64** (2005) 107–123
4. Dollár, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: VS-PETS. (2005) 65–72
5. Scovanner, P., Ali, S., Shah, M.: A 3-dimensional sift descriptor and its application to action recognition. In: ACM Multimedia. (2007) 357–360
6. Klaser, A., Marszalek, M., Schmid, C.: A Spatio-Temporal Descriptor Based on 3D-Gradients. In Everingham, M., Needham, C., Fraile, R., eds.: BMVC. (2008) 275:1–10
7. Willem, G., Tuytelaars, T., Gool, L.V.: An efficient dense and scale-invariant spatio-temporal interest point detector. In Forsyth, D.A., Torr, P.H.S., Zisserman, A., eds.: ECCV(2). Volume 5303 of LNCS. (2008) 650–663
8. Efros, A.A., Berg, A.C., Mori, G., Malik, J.: Recognizing action at a distance. In: ICCV. Volume 2. (2003) 726–733
9. Dalal, N., Triggs, B., Schmid, C.: Human detection using oriented histograms of flow and appearance. In Leonardis, A., Bischof, H., Pinz, A., eds.: ECCV(2). LNCS (2006) 428–441
10. Bobick, A.F., Davis, J.W.: The recognition of human movement using temporal templates. PAMI **23** (2001) 257–267
11. Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. In: ICCV. (2005) 1395–1402
12. Schuldt, C., Laptev, I., Caputo, B.: Recognizing Human Actions: A Local SVM Approach. In: ICPR. (2004) 32–36
13. J. Liu, J.L., Shah, M.: Recognizing realistic actions from video "in the wild". In: CVPR. (2009) 1996–2003
14. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision **60** (2004) 91–110
15. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (surf). Comput. Vis. Image Underst. **110** (2008) 346–359
16. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (1). (2005) 886–893
17. Chakraborty, B., Holte, M.B., Moeslund, T.B., Gonzalez, J., Xavier Roca, F.: A selective spatio-temporal interest point detector for human action recognition in complex scenes. In: ICCV. (2011) 1776–1783
18. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. PAMI **24** (2002) 971–987
19. Zhao, G., Pietikäinen, M.: Dynamic texture recognition using volume local binary patterns. In: Dynamical Vision. Volume 4358 of LNCS. (2007) 165–177
20. Zhao, G., Pietikainen, M.: Dynamic texture recognition using local binary patterns with an application to facial expressions. PAMI **29** (2007) 915–928
21. Kellokumpu, V., Zhao, G., Pietikäinen, M.: Human activity recognition using a dynamic texture based method. In: BMVC. (2008) 1–10
22. Kellokumpu, V., Zhao, G., Pietikäinen, M.: Texture based description of movements

for activity analysis. In: VISAPP (2). (2008) 206–213
23. Mattivi, R., Shao, L.: Human action recognition using LBP-TOP as sparse spatio-temporal feature descriptor. In Jiang, X., Petkov, N., eds.: CAIP. Volume 5702 of LNCS. (2009) 740–747
24. Nanni, L., Brahnam, S., Lumini, A.: Local ternary patterns from three orthogonal planes for human action classification. Expert Syst. Appl. **38** (2011) 5125–5128
25. Yeffet, L., Wolf, L.: Local trinary patterns for human action recognition. In: ICCV. (2009) 492–497
26. O. Klipper-Gross, Y. Gurovich, T.H., Wolf, L.: Motion interchange patterns for action recognition in unconstrained videos. In Fitzgibbon, A.W., Lazebnik, S., Perona, P., Sato, Y., Schmid, C., eds.: ECCV. Volume 7577 of LNCS. (2012) 256–269
27. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Action recognition by dense trajectories. In: CVPR. (2011) 3169–3176
28. Garrigues, M., Manzanera, A., Bernard, T.: Video extruder: a semi-dense point tracker for extracting beams of trajectories in real time. Journal of Real-Time Image Processing (2014) 1–14
29. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR (2). (2006) 2169–2178
30. Vedaldi, A., Zisserman, A.: Efficient additive kernels via explicit feature maps. PAMI **34** (2012) 480–492
31. Rodriguez, M.D., Ahmed, J., Shah, M.: Action mach: a spatio-temporal maximum average correlation height filter for action recognition. In: CVPR. (2008) 1–8
32. Tabia, H., Gouiffès, M., Lacassagne, L.: Motion histogram quantification for human action recognition. In: ICPR. (2012) 2404–2407
33. Jargalsaikhan, I., Little, S., Direkoglu, C., O'Connor, N.: Action recognition based on sparse motion trajectories. In: ICIP. (2013) 3982–3985
34. Lu, Z., Peng, Y., Ip, H.H.S.: Spectral learning of latent semantics for action recognition. In: ICCV. (2011) 1503–1510
35. Bregonzio, M., Li, J., Gong, S., Xiang, T.: Discriminative topics modelling for action feature selection and recognition. In: BMVC. (2010) 1–11
36. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR. (2008) 1–8
37. Wang, H., Ullah, M.M., Kläser, A., Laptev, I., Schmid, C.: Evaluation of local spatio-temporal features for action recognition. In: BMVC. (2009) 124.1–124.11
38. Nguyen, T.P., Manzanera, A., Vu, N.S., Garrigues, M.: Revisiting lbp-based texture models for human action recognition. In Ruiz-Shulcloper, J., di Baja, G.S., eds.: CIARP (2). Volume 8259 of LNCS. (2013) 286–293