

Detection of complex video events through visual rhythm

Berthin S. Torres¹ · Helio Pedrini¹

Published online: 19 October 2016
© Springer-Verlag Berlin Heidelberg 2016

Abstract The recognition of complex events in videos has currently several important applications, particularly due to the wide availability of digital cameras in environments such as airports, train and bus stations, shopping centers, stadiums, hospitals, schools, buildings, roads, among others. Advances in digital technology have enhanced the capabilities for detection of video events through the development of devices with high resolution, small physical size, and high sampling rates. This work presents and evaluates the use of feature descriptors extracted from visual rhythms of video sequences in three computer vision problems: abnormal event detection, human action classification, and gesture recognition. Experiments conducted on well-known public datasets demonstrate that the method produces promising results.

Keywords Visual rhythm · Spatio-temporal features · Abnormal event detection · Human action classification

1 Introduction

Technological advances and their influence in our society have promoted the generation of new trends and innovations in surveillance systems. The use of video security cameras in public areas (parks, underground stations, airports), private areas (gambling houses, hotels, banks) and restricted areas (high reactive chemical labs, electrical instal-

lations, radioactive areas) represents just an example of implementing surveillance solutions to record events for further analysis.

In this process of innovation, several powerful resources have been developed and are now currently available to address the automation process. Full access to high definition, infrared vision and even biometric identification systems show some of the advances that the community did to address security issues through the application of computer systems; however crucial concerns happen because these systems need to be reviewed and constantly controlled. For instance, in large closed circuit television (CCTV) installations with hundreds of cameras, only a small portion of them is usually watched in real time [23]. Eventually, the use of human operators to monitor surveillance cameras can represent a serious problem since looking at monitors for several hours is inherently a difficult task that may have health impacts or even generate controversy [52, 77, 89].

Many proposals have been done to address these drawbacks, in special to reduce the need for monitoring surveillance systems entirely by human operators. One of these proposals is focused on the development and improvement of new techniques in the computer vision field. Although video analysis can be used to figure out the solution to many problems, some of them (but not limited to) include human activity classification [1], abnormal event detection [20], gesture recognition [47], action similarity [43], motion analysis [32], person re-identification [33], urban traffic analysis [13], video background subtraction [78], object tracking [95].

In order to deal with these tasks, a video sequence may be observed as a set of actions from a high-level point of view or as a set of pixels in a low-level fashion. On the one hand, high-level involves an understanding that integrates each part within the scene; for instance, in trajectory analy-

✉ Helio Pedrini
helio@ic.unicamp.br

Berthin S. Torres
berthin@liv.ic.unicamp.br

¹ Institute of Computing, University of Campinas, Campinas, SP 13083-852, Brazil

sis where a moving object is considered to be an amorphous blob that is tracked [14,68]. Despite the fact that tracking provides a useful behavior (and/or contextual) model, it is only computationally suitable for scenes with few objects (e.g. traffic monitoring [40]), but impractical in crowded or complex scenarios with superposition and occlusion. On the other hand, low-level processing extracts features to analyze the activity pattern of each pixel or groups of pixels that share spatio-temporal information; due to their low complexity, these features allow us to propose many real-time applications to address some of the aforementioned problems.

Even though the difference between high and low level processing is related to how they manage, extract and process the features, some problem specifications require one over the other. High level processing uses more complex models to describe, represent and interpret objects or scenes as a whole, whereas low-level works with assumptions about the relations among pixels in order to group them up into regions which are processed to extract shape-based, texture-based, color-based, and motion-based (in case of video processing) features.

In contrast to most of the common low-level features used in computer vision tasks that are obtained from motion and texture descriptors, in this work we focus on exploring and proposing a framework to analyze video sequences using a low-level feature descriptor obtained from visual rhythms of the video sequences. The visual rhythm, as further described in the following sections, depicts a different representation of a video sequence. Instead of managing a set of frames over time, we deal with slices formed by one-spatial dimension (axis X or Y) over time (T). This constitutes a compact and effective scheme that has not been widely explored in the literature.

This work presents a deep literature review about the concept of visual rhythm and its usage in other problems as well as a description of the problems we are addressing. Then, we describe our proposal to reduce computer vision tasks into image classification and image matching problems. Due to the great diversity of tasks, we have selected three specific problems: abnormal event detection, human action classification, and gesture recognition. Experiments are conducted on UMN, Weizmann, KTH, and SKIG datasets. The obtained results are promising when compared to state-of-the-art approaches; and, to the best of our knowledge, this is the first attempt to apply visual rhythms to the mentioned problems.

Section 2 presents and discusses relevant works found in the literature related to the visual rhythm and its applications. Moreover, we also describe the epipolar-plane image, which is a specific visual rhythm obtained from a static scenario and a moving camera. Section 3 provides a more formal definition of the visual rhythm and describes our framework for extracting and classifying the low-level features from video

sequences. Experimental configuration parameters, dataset specifications, protocols, and comparative results are presented in Sect. 4. Finally, Sect. 5 concludes the work with limitations, final remarks, and directions for future work.

2 Background

A review of relevant visual-rhythm related concepts is presented in this section, including epipolar-plane images analysis of video sequence (XT and YT) slices, visual rhythm, and a general and powerful structure to address video analysis problems.

Then, we define and present a literature review of issues associated with the problems we cover up in this work. Available results from the literature are shown and compared in Sect. 4.

2.1 Epipolar images and XT - YT slices

Instead of considering video sequences as a collection of frames, we define them as a collection of slices having one-spatial dimension (X or Y axis) and one temporal dimension (T axis). By taking the specific configuration where the camera moves along a straight line and the scene remains static, these slices capture spatio-temporal information through a better structure than just frames as spatial images. This scheme was initially proposed and analyzed by Bolles [10] with the name of epipolar-plane images (EPI) in order to describe the motion features in terms of geometric stereo [57].

EPIs can be used to segment video sequences based on the 3D space continuity such that, through the concepts of epipolar geometry, the problem of segmenting the EPI volume is reduced into a new problem of how to analyze the epipolar curves [6]. Such curves, even with the requirements that EPI needs (the camera configuration), preserve useful information about video motion.

The inverse of the previous assumption, a static camera recording objects in motion, generates slices containing other kind of spatio-temporal information. Nigoyi and Adelson [64] observed that video sequences, under this new context, produced XT slices (a cut in the X image plane over time T) with particular patterns when the video records people walking. The reason is that these new slices, in special the ones capturing the motion of human legs, depict a braided pattern snake. Years later, Ran [70] used these patterns to perform person identification tasks in video sequences.

Figure 1 shows braided snake patterns obtained from a person walking in a scene and one camera transition. Thus, spatio-temporal slices were also applied to detect three different camera shots or transitions: cut, wipe, and dissolve [59].

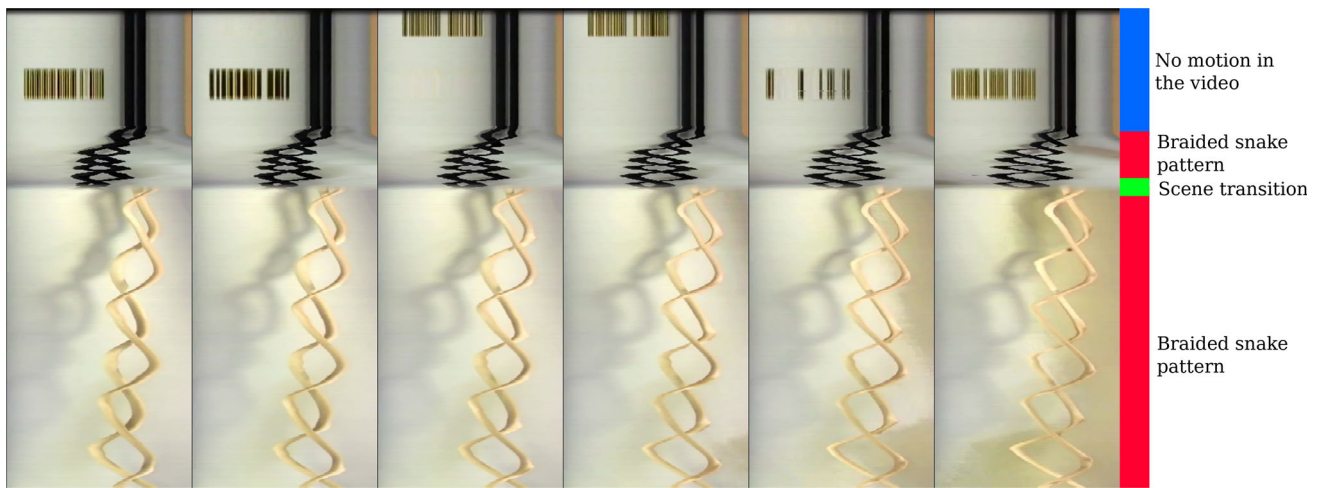


Fig. 1 Braided snake patterns obtained from a person walking in a room. The visual rhythm (XT slices) for the first video frames does not contain any directional information since the person remains static

within a certain amount of time, then the person walks and the braided pattern can then be observed. The discontinuity in the images describes a cut transition in the video

Cut corresponds to a sharp transition from one shot to another, wipe gradually replaces the shots giving the effect of traveling from one side to another, and dissolve overlaps two shots where one of them smoothly disappears whereas the other appears. Another attempt, for the same problem, was done by Guimarães et al. [35] through the use of mathematical morphology operations on the slices for detecting video transitions. Ngo [60] used Gabor decomposition as texture features and Markov models to represent the dependencies of the slices. They also proposed a more complex analysis over these XT and YT slices to identify the camera motion types (static, pan, zoom, and tilt) and segment the foreground (and background) from the videos. Slices were described by tensor histograms that aimed to model the motion distribution.

2.2 Visual rhythm

Due to the applicability of the (XT and YT) slices from video sequences, new proposals have been made to address other computer vision problems. Moreover, a new term is adopted in the literature to indicate a set of rows and/or columns extracted from each frames, which are later concatenated to form a new image: the visual rhythm (this concept will be explored in more details in Sect. 3). Based on this idea, Valio et al. [86] extracted the visual rhythms for caption detection in video sequences. Captions preserve a well-defined rectangular structure and they could be easily segmented.

Another application using the visual rhythm is the video face spoofing detection problem by Pinto et al. [72]. In that work, instead of directly handling in the image spatial

domain, they worked on the Fourier spectrum; thus, the visual rhythms contained data from the frequency domain. For the detection step, features were extracted using gray-level co-occurrence matrices (GLCM), but this could represent a drawback since the Fourier spectrum contains values which vary in high ranges of real and imaginary numbers. Although they opted to use the logarithm of the Fourier spectrum, a quantization step is still necessary because the GLCM works with positive integer values and some useful information could be missed in the process. However, they obtained higher accuracies during their experiments.

In the remainder of the text, we adopt the concept that events are atomic low-level spatio-temporal entities [15] that express: (i) the action being performed and (ii) the actor who performs the action [80].

2.3 Abnormal event detection

Problem definition When we refer to the process of analyzing surveillance videos as a task where the monitor needs to watch for uncommon occurrences, unintentionally we are modeling the problem of identifying abnormal events in surveillance systems. Even if it is not a direct application for crime prevention, it can help to prevent future events to keep people safe or to learn new patterns according to the observed object behavior.

For this specific problem, events are commonly categorized as normal and abnormal, such that abnormal events are those with very low likelihood of occurrence. Additionally, as described by Chandola et al. [17], they are observations that do not conform to a well-defined notion of normal behavior.

In contrast, abnormal (also known as anomalous)¹ events are situations that deviate their behavior from the normal occurrences, both spatially and temporally. For instance, in a video-sequence of an underground train station, people falling into the train tracks will represent the abnormal events since the common behavior is to have trains on the tracks with people waiting in the platforms.

One way to see this problem is, for instance, to suppose that a normalcy² model is given to describe the behavior in a scene. This model can be divided into temporal and spatial normalcy submodels [50]. Temporal normalcy is related to know how normal events are recurrent over time, so the normal behavior model is learned while the time passes by; on the other hand, spatial normalcy relates events within a group (e.g. a crowd). A road, where the usual event is to see people walking from west to east (or vice-versa), can depict an abnormal event when suddenly a car appears. The car itself cannot be described as an anomaly; however, in the middle of the crowd, it definitely is. Detection of such anomalies is based on the spatial context. From another point of view, a normalcy model can also describe the behavior of a crowd as a global unit (e.g. a crowd walking) or local units (e.g. a person running). Therefore, events may also be categorized into local and global instances. Following our examples, if a person starts running within the crowd (where the common behavior is walking), it will trigger an abnormal-local event; however, if the crowd starts running, that will be considered as an abnormal-global event. Crowds, at the same time, can be classified into (a) structured crowds, where the main motion is directed by environmental conditions (e.g. elevators), and (b) unstructured crowds, which are those where objects can move freely (e.g. walking on the street) [66].

Many strategies for solving this problem have been proposed in the literature. We can differentiate them by considering the features that are used in the approaches. From now on, it is important to define two issues in order to correctly detect abnormal events: (i) event representation and (ii) anomaly measurement. In the following paragraphs, we describe some state-of-the-art approaches that used low-level features.

Literature review Feng et al. [31] proposed an approach that applies an online SOM (self organized map) to cluster adaptively motion patterns. Video sequences were considered as sets of clips, each clip describing the optical flow patterns. SOM clustering, based on the estimated parameters of a Gaussian distribution for a clip, is able to distinguish

¹ Although some efforts have been made to differentiate among these terms, in this work, abnormal events will be assumed to similar to unusual, rare, suspicious, anomalous, irregular, outlying or atypical events.

² Normalcy or normality is the state of being normal or usual.

between normal and abnormal behavior in just a fixed scale. This drawback was overcome by Biswas and Baby [7], who considered histograms of motion magnitudes at different scales using pyramids and a Gaussian of mixture model to represent the normal behavior. Detection of anomalies started at the coarsest level moving towards the finer scales only if anomalies were likely to be found. The novelty in this approach was that motion vectors were obtained from H264/AVC compressed videos, which improves drastically the total execution time.

Dealing with crowds as global entities, Menhran et al. [53] came up with the Social Force model to capture the dynamics of interaction forces. In crowded scenarios, the actual force can be modeled as a result of the personal desire forces and the interaction forces, since the individual motion is restricted when people are densely packed. However, interaction forces are not enough to detect anomalies and, thus, motion patterns were created from the forces over periods of time. Later, Zhang et al. [99] improved the social force model in order to represent the concepts of contact, consistency, and exclusion. The method used the fourth-order Runge–Kutta algorithm with bilinear interpolation to generate the optical flow, unlike most popular methods for calculating the optical flow such as the ones proposed by Lucas and Kanade [49], Horn and Schunck [37], and Farneback [28]. Chen et al. [18] clustered the optical flow to obtain groups of human crowds and, for each cluster, they modeled the force field from the orientation, position, and crowd size.

Wang et al. [93] used a covariance matrix of the optical flow and the image intensity over the whole frame as the feature descriptor, then a non-linear classifier SVM was applied in an online fashion. Thida et al. [83] learned a model of regular crowd behavior based on the magnitude and direction of the local motion vectors. A video sequence was represented as a fully connected graph with local regions as vertices and the connectivity among these regions (weighted edges in the graph) represented their similarity. The graph, analyzed with spectral graph theory, embeds local motion patterns. Another graph-based method was proposed by Saligrama et al. [73], where abnormal events were defined by ranking composite scores for video segments. They adopted a grid-like graph structure in 3 dimensions, nodes represented motion features and their spatio-temporal relations were kept using the edges. The graph, along with Markov assumptions, allowed the composite score reconstruction.

Cong et al. [19] detected abnormal events via sparse reconstruction cost considering histograms of optical flow at different scales. To represent spatial and temporal relations, they proposed three different neighbor arrangements that allow them to retain spatial, temporal and spatio-temporal information. Tang et al. [81] addressed the problem using

sparse coding with motion features without any pre-learned dictionary. Both methods claimed to work in an online fashion, which is a great advantage. Due to the nature of the abnormal event detection problem, we cannot assume to know beforehand any negative data point (abnormal event) and we train with only positive points (normal events), which is more complex when the normalcy concept changes over time. Differently from the bag-of-words approach, sparse coding has been extensively studied to update the dictionary adaptively [51, 100]. Other online approach, proposed by Li et al. [46], improved the mixture of dynamic textures (MDT)—initially modeled by Mahadevan [50]. Because the MDTs are not scale-invariant, they hierarchically learned MDTs at multiple scales. Spatio-temporal abnormalities were integrated into a global anomaly map using online conditional random fields.

Nam [58] extracted motion features from the optical flow at different scales to calculate a histogram of orientations and magnitudes. Spatial relations of neighbors within blocks were used to build a probability distribution of the crowd. Entropy and the normalized mutual information defined a metric for the video frames, which were further analyzed using a set of rules to determine whether a set of frames describe an abnormal event or not. However, during their experiments, the directional crowd energy obtained better results than the mutual information.

Hung et al. [38] showed a direct application of the scale-invariant feature transform (SIFT) algorithm [48] with bag-of-words to achieve the 1.00 AUC (Area Under the ROC Curve) value on the UMN dataset (see Table 3), they also run cross-scene training experiments. We believe that the idea of using transfer learning in this specific problem will address the lack, within the training step, of having only positive data; and, in consequence, new ideas could be tested to achieve better accuracies.

2.4 Human action classification

Problem definition Similar to the abnormal event detection problem, two basic components are identified in the action recognition task [41, 87]: (i) action representation and (ii) action classification. Action representation can also be divided into: shape-based models, motion-based models, geometric human-body models, interest-point models, and dynamic models [36]. Shape-based models are one of the most successful representations that estimate the silhouette of an object (a person) through time forming a 3D-silhouette (or tunnel). Motion-based models extract characteristics of the object movements and their deformations. Since human actions can be represented by the motion of some body parts (torso, hands, and legs), under a controlled environment, these parts are easy to identify in order to construct a parametric model using the geometry of the body (e.g. Kinect).

Interest-points have also been applied to represent actions (e.g. Laptev [44]), actions that can also be described by using dynamic models considering temporal variations with state-space transition models.

Literature review Gorelick et al. [34] analyzed actions directly from the space-time volumes of the tunnels by solving the Poisson equation. Since tunnels contain information about the human pose and the motion of the body, the Poisson equation extracts the following shape properties: space-time saliency, space-time orientations, and weighted moment. The classification task was done by the nearest neighbor algorithm. Tunnels contain 3D information; however, instead of rather than considering the whole shape of an action, just the moving parts can be mixed into a single image and have the cumulative motion shapes (CMS). This approach, proposed by Alcantara et al. [2], addressed the drawback of creating time consuming complex models; so that, after the concatenated image of the motion shapes was created, they extracted interest points (set of coordinates) equally distributed in the bounding box that covers the CMS. A support vector machine (SVM) and the nearest neighbor algorithm were applied in the classification process.

Based on motion-based models, Wang et al. [92] divided a frame into blocks and created histograms of optical flow; but instead of using them as the feature descriptor, they opted for some statistical values (obtained from each block) such as: the portion of active moving pixels, the average speed, the predominant direction in a block, the bin-index of the predominant direction, variance and divergence of the direction distribution. AdaBoost, using a weak classifier for each feature dimension, performed the classification step. Fawzy et al. [30] used a 2D-HOOF (Histogram of Oriented Optical Flow) extracted from the contour of the person who performs the action. Due to high dimensionality of the 2D-HOOF features, they applied 2D-PCA (principal component analysis) to produce a better representation of the dominant features.

Guo et al. [36] combined local silhouettes and local motion features with a covariance matrix. Classification was tested with nearest-neighbors and sparse linear approximation. Even though this method seems to be simple, it obtained high-accuracy rates in many datasets.

Yang et al. [96] addressed the problem by using only a single-shot clip as training data-set. They divided the optical flow into four channels: *x-positive*, *x-negative*, *y-positive*, *y-negative* (since the optical flow is actually a vector), and then these are densely sampled along the frame using a similarity metric for any two samples. Since people can perform same actions with some variations, just one training data per action cannot be used as an action template; hence, they defined a more general distance function which compares and looks for the best pair-matching blocks from any two different videos.

Another attempt that minimized the training dataset, proposed much earlier by Schindler et al. [74], only used short sequences (up to 10 frames) for training. They extracted local edges (from a bank of log-Gabor filters) and motion information (dense optical flow mapped into different flow filters) to compare later with templates previously learned in the training step. Rather than using the comparison between the features as result, they defined a similarity score which was passed as input to a classifier that makes the final decision.

2.5 Hand gesture recognition

Problem definition Human computer interaction has currently received much emphasis on the developing of new technologies. Devices such as stereo cameras, Kinect or motion sensors (e.g. Asus Xtion) generate a new type of information that not only contains the video sequence with gestures, but also a depth image. Compared to the human classification problem, action representation can be depicted in a similar fashion, however, two major categories are commonly adopted in this area: 3D models and appearance-based methods. The first includes 3D texture volumetric models, 3D geometric models, and 3D skeleton models, whereas the second category includes color-based models, silhouette-based models, geometry-based models, deformable models and motion-based models [12, 71].

Literature review Some methods available in the literature employ depth information to build 3D models. Fanello et al. [27] represented the actions by applying to them 3D histograms of optical flow (HOOF) on the RGB images and global histogram of oriented gradients (HOG) features on the depth images. Additionally, sparse codes produced a compact portrait of the actions. De Rosa et al. [22] used the same feature descriptors to learn an incremental non-parametric prediction system for online action recognition. Devanee et al. [24] studied a form to depict actions as trajectories using depth map sequences. They reconstructed trajectories by taking each point of the skeleton of an action, then the distance between the projected trajectories was measured in a shape space. The elastic distance between any two trajectories defined a similarity metric since the shape of a trajectory can be viewed as a point in a shape space of open curves. Moreira et al. [56] extended their previous work [2] using the depth images to extract shapes and then obtain the geometric interest points from the cumulative motion shapes (CMS). Vo et al. [88] also used geometrical properties from the silhouettes, then graphical models were applied to the action recognition process. Yu et al. [97] fused RGB and depth local flux features images to have a binary descriptor. They adopted the local flux features (LFF) to describe the local flux for each pixel and combine the RGB and depth LFFs into the Hamming space. Discriminative rep-

resentations were learned through the structure preserving projection (SPP) that keeps pairwise structure of local features and the relations between samples and classes.

More complex approaches, such as graphical models and convolutional neural networks (CNN), have been recently applied to the gesture recognition problem. Antonucci et al. [5] modeled discrete-time sequences through imprecise probabilistic information with hidden Markov model (HMM). Molchanov et al. [55] proposed a recurrent 3D CNN for dynamic hand gesture recognition. A video sequence was divided into clips, then spatio-temporal filters were applied to each clip. From each clip, they extracted blocks that served as input to a recurrent layer of their CNN, then the output of the CNN was used to train an SVM that performed the classification step.

3 Proposed method

Our proposal, based on visual rhythms extracted from video sequences, generates the histograms of oriented gradients to train a classifier, detects abnormal events and recognizes actions. As expected, different preprocessing and classification steps are required to solve each problem. All required stages are detailed and justified considering our own definition of visual rhythm, which is given in the following paragraphs. This definition differs from the previous works since it allows us to define not only vertical or horizontal slices, but also other types of structures to obtain the visual rhythm.

3.1 Visual rhythm

Let $V = (D_V, I)$ be a video sequence, where $D_V \in R^{W \times H \times T}$ and I represents the intensity values that each frame in V can take (for gray-scale videos, $I \in [0, 255]$). Let $\pi = p_1 \rightarrow p_2 \rightarrow \dots \rightarrow p_r$ be a sequence, not necessarily consecutive, of points defined in the $X - Y$ image plane, and $f_\pi(t)$ be a function that extracts all pixels $p(x_i, y_j, t_k) \in D_V$ such that they follow the sequence defined by π with $t_k = t$. The visual rhythm \mathcal{VR} , illustrated in Fig. 2, is defined as the combination of all pixels for every value $t \in T$, and it can be seen as a matrix-like structure with several $f_\pi(\cdot)$ as rows, such that:

$$\mathcal{VR}(V, \pi) = \begin{bmatrix} f_\pi(t = 1) \\ f_\pi(t = 2) \\ \dots \\ f_\pi(t = T) \end{bmatrix} \quad (1)$$

An informal analysis, presented as follows, shows that the visual rhythm contains spatio-temporal information allowing us to work with trajectories in a low-level processing fashion.

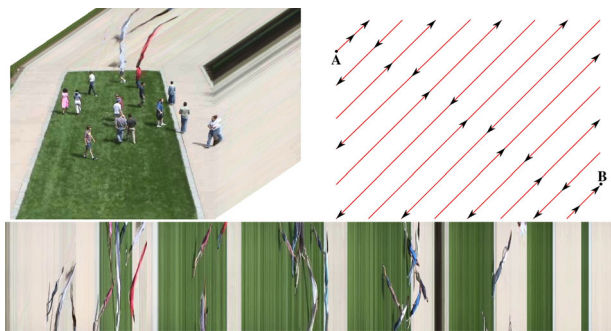


Fig. 2 Visual rhythm (*bottom figure*) for the video sequence (*left*) considering the path π as shown on the *right* side. The *arrows* in π show the direction of the path that starts at the *upper-left corner* (point *A*) following a zigzag configuration of points, and ends at the *bottom-right corner* (point *B*). Note that π contains gaps in the borders

Given two image frames $H = V(t)$ and $G = V(t + \Delta t)$ extracted from the video sequence V at times t and $t + \Delta t$ respectively, the optical flow defined as the displacement vector indicates, for all pixels in G , how much those pixels have moved over the previous image H . If we assume that the brightness difference of the pixels in both images is almost constant (since Δt is usually small), H can be modeled as $H(x, y) = G(x + u, y + v)$, where the components u, v form the optical flow.

Now, suppose that we have a rigid object³ with a rectangular-regular shape S of $N \times M$ (as a collection of pixels in the $X - Y$ image plane) which is moving with a constant velocity in a period of time Δt . Ideally, the optical flow for all points in S will be (u, v) . If $|u| < N$ and $|v| < M$, we could ensure that at least two points p_i and p_j in S will appear in the same (x, y) coordinates because there is an intersection and the object cannot be found in more than u or v units in G with respect to H . Hence, if $(x, y) \in \pi$, p_i and p_j will also be in π and they will appear in the visual rhythm. Therefore, for short intervals Δt with small values of $|(u, v)|$, the visual rhythm captures the object trajectory defined by π (as shown in Fig. 2).

3.2 Histogram of visual rhythms

We have seen that the visual rhythm contains trajectory information; however, the difficulty is how to process it in order to solve real-world problems. In this section, we introduce a method (Fig. 3) which obtains low-level feature descriptors from the visual rhythm of a video sequence. Nevertheless, these features may either be directly used for classification or used to build a codebook and obtain histograms of visual words. The decision basically depends on the problem we are dealing with.

³ An object that fits the principle of distance conservation [54].

Since our method does not use color information, the visual rhythms are extracted from each video sequence in gray-scale. Then, the initial idea is to segment the background to get only the trajectory. However, since the visual rhythm usually captures texture, it is complex to determine which regions should be removed. According to how the sequence π is defined, some segmentation approaches can be ineffective. Alternately, if π captures static points in the video, the visual rhythm will have approximately a constant texture segment; otherwise, we will see a combination of regions with some parts formed by constant texture structures and other parts containing the trajectory information.

The first step of our method takes the visual rhythm and (depending on the problem) performs preprocessing operations. Operations that aim at achieving a better representation of the trajectory; thus, for a given region $R = R_B \cup R_F$ extracted from the visual rhythm, we remove the background R_B to keep just the foreground R_F that contains the trajectory. Since detecting anomalies and classifying actions are different problems; the main idea in this step considers that, under a controlled environment (e.g. with no illumination changes), we could take a segment of a vertical line h representing the background R_B , and the pixels which differ drastically from h should be considered as foreground. More details about the preprocessing are explained separately for each problem in the next subsections. Therefore, at this point, we assume that we have a good description of the trajectory.

Once we obtained the trajectory representation, it is divided into small patches called cells b_{ij} of $B \times B$. For each cell, we obtain a 1-dimensional histogram of the edge orientations from all pixels in the cell. This histogram is L1-normalized. Note that this procedure is similar to the histogram of oriented gradients (HOG) [21]; however, HOG merges groups of cells into blocks and then constructs the normalized histogram for each block.

3.3 Abnormal event detection

We noted that the visual rhythm describes how the objects move in time with respect to how the sequence π is defined. For instance, Fig. 4 shows the visual rhythm extracted from a video sequence, where it is easy to observe that something occurs within the red interval due to trajectory interruption of the curved lines and/or their abruptly change in direction.

Problem formulation Let a video sequence V be a collection of several patches τ of visual rhythms, such that $V = \{\tau_p\}_{p=1}^N$. The abnormal event detection problem, as a reconstruction problem, is modeled as follows:

$$\tau_p \text{ is abnormal} \iff S(\tau_p, \tau_q) \leq \gamma \quad \forall q \quad \text{and} \quad p \neq q \tag{2}$$

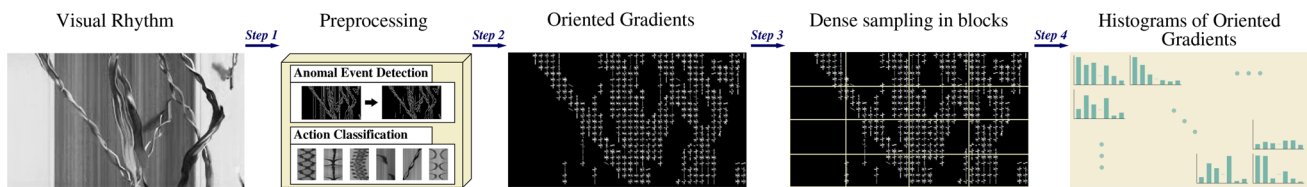


Fig. 3 Proposed methodology for obtaining the histograms of oriented gradients for the visual rhythm

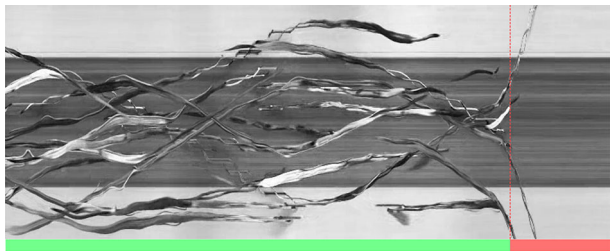


Fig. 4 Visual rhythm for UMN dataset (video sequence 2). Green region denotes the normal events, whereas red corresponds to the abnormal events

where \mathcal{S} represents a similarity function and γ is a threshold value.

With this brief definition, the rest of this subsection covers all the aspects considered to detect the anomalies by using only the visual rhythm features; for convention, we denote these features as $\phi(\tau)$.

3.3.1 Preprocessing

Given a region R in τ , our goal here is to extract R_F which represents the trajectory of the objects. Let suppose that we have segmented R_F , and now we aim at finding a finer representation (Fig. 5). One common step is to consider the derivatives and construct a gradient map, but this map still contains coarse directional information. Another alternative is to emphasize the edges through the Laplacian of Gaussian filter, then set a threshold value to have a black-white image containing directional information.

Even though the results are promising, from our experiments we have seen that trying to obtain a proper approximation of R_F is quite difficult. Moreover, after applying the first order derivative or the Laplacian of Gaussian, we must find a finer representation which will depend on the algorithm to binarize the image. In this sense, a common choice to use is the Canny edge detection filter [16]. This filter smooths the image for noise attenuation, then finds the gradients and uses non-maximum suppression to remove low edge responses. A double threshold is used to generate a binary image and the final image is found after removing weak edges. The issue now is how to obtain the foreground.

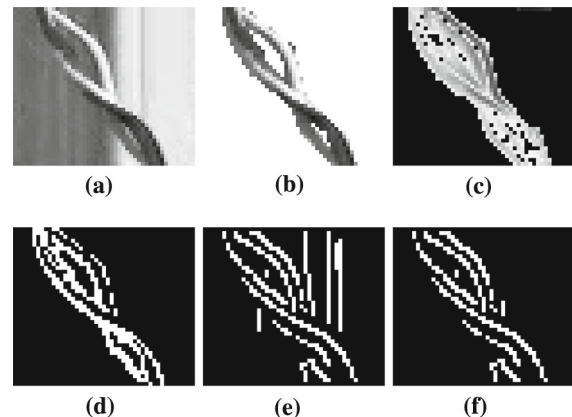


Fig. 5 Process of obtaining a finer representation of the trajectory in a given region R (a) with the foreground R_F (b). The first order derivative of R_F (c) shows a coarse representation, and a binarized Laplacian of Gaussian (d). An alternative solution was considered by using Canny edge detector (e) in R and then applying an opening morphology operation with a vertical line as the structuring element (f)



Fig. 6 Preprocessing step for the abnormal event detection problem. From the visual rhythm, edges are found by Canny detector (a) to represent directional information of the trajectory; but the background produces a noisy pattern as vertical lines that is removed by using opening morphology (b) with a vertical line as the structuring element

Since the background is almost constant with small lighting changes, we can use h to assume that it is the pattern that represents R_B , so that the edges detected in h should repetitively appear in the background. If a point $p(x_i) \in h$ responds to the Canny filter as an edge, R_B will contain a vertical line $y = x_i$. After removing these vertical lines (Fig. 6), we will obtain a finer approximation of the edges we search for in the preprocessing step. Vertical line removal can be performed by morphology operations with an adequate structuring element or even more complex techniques such as Hough transform [26].

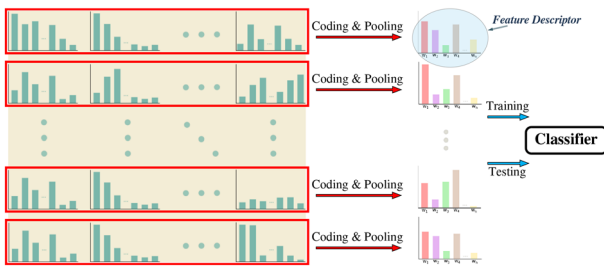


Fig. 7 Final feature descriptors for a video sequence used for training and testing

3.3.2 Feature descriptor

Instead of having a global representation for the entire visual rhythm by concatenating all histograms, we use the histograms as block-features. Although these histograms contain trajectory information in space-time dimensions, they cannot explain by themselves the behavior of an event; hence, we build a codebook \mathcal{C} using the block-features as words. To preserve spatial information, the final feature descriptor pools the codewords from all cells that are neighbors in a horizontal region (Fig. 7). Finally, these features are used in the classification step for training and testing.

3.3.3 Classification

Due to the nature of the problem, abnormal events are assumed to be those which are different from previous observed events. According to the literature, we only have positive training data points representing normal events.

A widely studied technique to deal with this kind of situation is the support vector data description (SVDD) [82], which finds the minimal circumscribing hypersphere in a high-dimensional space for a set of positive training data points. In other words, it solves the following optimization function:

$$\min_{R, \mathbf{b}, \xi_i} f(R, \mathbf{b}, \xi_i) = R^2 + \frac{1}{vn} \sum_{i=1}^n \xi_i \tag{3}$$

subject to $\|\phi(\tau_i) - \mathbf{b}\| \leq R^2 + \xi_i$ and $\xi_i \geq 0$, where n is the number of data points, ξ_i are the slack variables, v is the user-defined parameter to control how much slack we are going to admit, R and \mathbf{b} are the radius and the center of the hypersphere, and $\phi(\tau_i)$ represents our feature vector.

After applying Lagrange multipliers and using the Golf duality of $f(R, \mathbf{b}, \xi_i)$, the optimization function (3) is reduced to find the best values of the dual variable α_i such that:

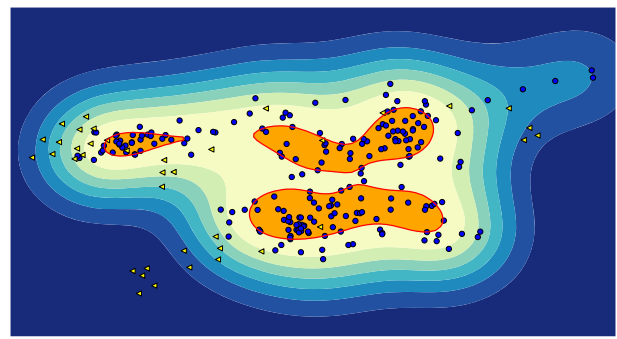


Fig. 8 One-Class SVM with a RBF kernel. Orange regions represent the hypersphere plot. Blue dots are the normal points, whereas triangles the abnormalities. This image was obtained from the UMN dataset (first scenario) and PCA for dimension reduction

$$\operatorname{argmax}_{\alpha} \left(\sum_{i=1}^n \alpha_i k(\tau_i, \tau_i) - \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j k(\tau_i, \tau_j) \right) \tag{4}$$

subject to $0 \leq \alpha_i \leq \frac{1}{vn}$ and $\sum_{i=1}^n \alpha_i = 1$, where $k(\tau_i, \tau_j)$ is the kernel that represents the inner product $\phi(\tau_i) \cdot \phi(\tau_j)$.

Assuming that the kernel is invariant to translation (e.g. RBF kernel); $k(\tau_i, \tau_i)$ will take a constant value $\forall i$, and the first term of the equation is removed; so that, it becomes:

$$\operatorname{argmin}_{\alpha} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j k(\tau_i, \tau_j) \tag{5}$$

This new optimization function represents the core of the one-class SVM (OC-SVM) classifier (Fig. 8) for the novelty detection problem [75], and it is what we use for the classification stage.

3.4 Human action classification

In this problem, the input is a video sequence that contains a person performing an action and our goal is to detect which action is being performed. For the abnormal detection problem in crowd scenes, we have observed that the visual rhythm contains trajectory information of multiple objects (people). Based on the braided pattern from a person walking, as seen in Fig. 1, our intuition assumes that different human actions can describe distinct patterns and same actions should contain similar patterns because they require our body to act in a particular way (Fig. 9).

The description of the preprocessing step (Fig. 10) to deal with the extraction of these patterns from the visual rhythm, as well as the algorithm for performing the classification task, is detailed next.

Problem formulation Suppose a video sequence V as a collection of several patterns ρ obtained from the visual rhythms, such that $V = \{\rho\}_{\rho=1}^N$. Let C_i ($i = 1, 2, \dots, m$)



Fig. 9 Visual rhythms for the Bend action

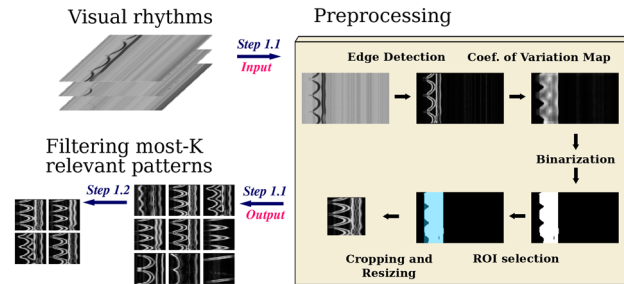


Fig. 10 Preprocessing stage for action classification

be a class used to indicate that an event is a container of several patterns ρ_{C_i} in the way that, the classification problem is depicted by the following optimization function:

$$\operatorname{argmin}_i \sum_{\hat{\rho} \in \rho_{C_i}} \min \mathcal{D}(\rho_p, \hat{\rho}) \tag{6}$$

The purpose of this function is to find the class C_i for which the patterns $\rho_p \in V$ have the lowest dissimilarity value with respect any pattern in C_i .

3.4.1 Preprocessing

As previously mentioned, a region R contains some background R_B that will almost be removed through Canny edge detector and vertical line pruning. However, in this problem we can see (Fig. 9) many regions with irrelevant information because no actions are performed by external agents (we just have one actor); and, consequently, $R = R_B$ in these regions. Even using the last approach to successfully ignoring the useless information, the Canny detector is not able to identify the borders of the patterns with high accuracy (at least not for our purposes). Hence, we applied the Sobel edge detection operator since it keeps most of the relevant edges at different gray-scale tones. However, the background still contains information, then a region of interest (ROI) that contains the pattern of the action should be found. One option is to apply a thresholding technique directly to the Sobel image, however, if the background contains lines or if the image is too noisy, the thresholding will probably fail.

After empirical tests, we opted to create a map of the coefficient of variation over the Sobel image. The coefficient of

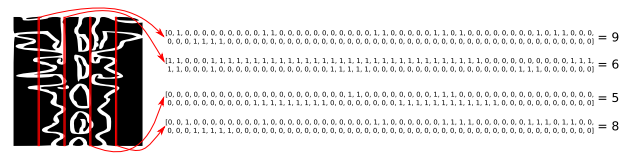


Fig. 11 The pattern has a weight of 28 considering the number of white segments from the highlighted columns in red

variation ($c_v = \frac{\sigma}{\mu}$) represents the dispersion of the pixel intensities in a window $W^{w \times w}$, where μ denotes the mean of all pixels in W ($\mu = \sum_{p \in W} \frac{p}{w^2}$), and σ the standard deviation in W ($\sigma = \sqrt{\sum_{p \in W} \frac{(p-\mu)^2}{w^2}}$). The map of the coefficient of variation was built through a sliding window strategy. In this new image, R_B must be close to zero since pixels inside a small window W are almost constant with small changes. Even if a line is present in W , c_v will remain small, making the thresholding technique (e.g. Otsu’s method [65]) suitable to be applied. Finally, the ROI can be easily found in the binary image. In order to standardize the dimensions of the patterns, we resized each ROI to 100×100 to have a set of patterns $\{\rho\}_{i=1}^n$ for any given video sequence V .

Not all generated patterns are useful to differentiate actions. Some of them may not have any trajectory information due to the absence of actions performed under certain visual rhythm configuration. Therefore, we need to filter the most K relevant patterns. This filtering process is one of the most important steps for action classification since an adequate strategy drives a robust classification.

Suppose that each pattern is a black-white image, and $g(\rho_i)$ assigns a weight to ρ_i such that, the most K relevant patterns will be the ones with higher weights. Let $g(\cdot)$ be a function that counts how many vertical white segments a pattern contains. In other words, we are going to identify those patterns that have traces with many variations in the vertical axis. For instance, Fig. 11 illustrates the strategy that assigns a weight of 28 to the pattern.

It is worth mentioning that patterns are gray-scale images, and $g(\cdot)$ is defined only for binary images. Therefore, given a pattern ρ_i in gray-scale, the weight of ρ_i is expressed as:

$$g(I_{\text{plane}}(\rho_i, \{5, 6\})) + g(I_{\text{plane}}(\rho_i, \{6, 7\})) \tag{7}$$

where $I_{\text{plane}}(\rho_i, L)$ extracts the image planes from ρ_i at levels depicted in L and returns a binary image with white intensity for every pixel greater than zero. For instance, Fig. 12 shows the binary images generated by $I_{\text{plane}}(\cdot)$ at levels $\{5, 6\}$ and $\{6, 7\}$.

3.4.2 Feature descriptor

Histograms of gradients are used as raw feature descriptors just with the modification that cells b_{ij} can overlap (different

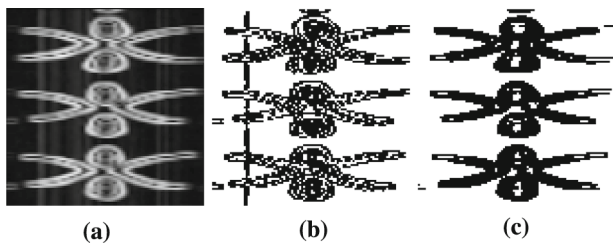


Fig. 12 **a** A pattern ρ_i in gray-scale. **b** The returned image by $I_{\text{plane}}(\rho_i, \{5, 6\})$ considering only the 5th and 6th gray-levels, and **c** the returned image by $I_{\text{plane}}(\rho_i, \{6, 7\})$ considering the 6th and 7th gray-levels

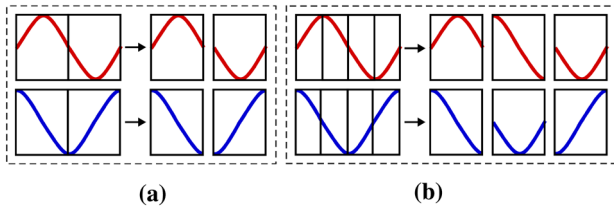


Fig. 13 Illustration of a sine function (with different phase values) to explain the reason why we opted to use overlapping windows to extract the histograms. **a** Non-overlapping scheme of two similar functions will not match any of the windows, whereas **b** overlapping windows will

from the abnormal event problem where we considered non-overlapping cells to build the histograms).

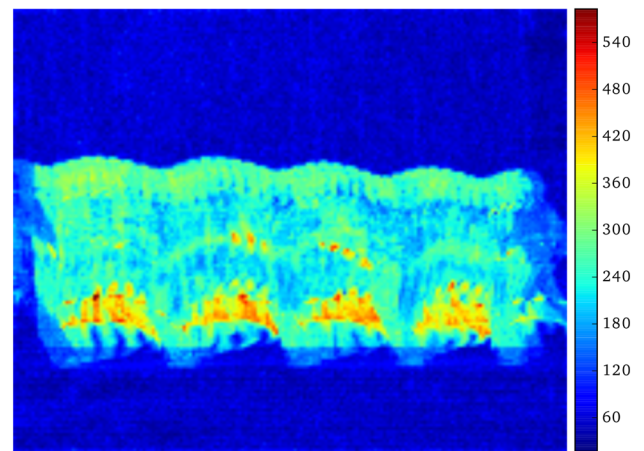
An action is a repetitive sequence of movements, so the visual rhythm contains a sequence of small patterns that are also repeated but with different initializations. For instance, suppose that we have two sine waves with some displacement in the x -axis (Fig. 13); working with a non-overlapping window, we are not able to match these sine functions; however, with an overlapping scheme, two (out of three) windows are identical, which means that if we repeat some information we will (at some point) have the same data for similar patterns (but with different initializations).

Some actions (for instance, running and walking) sketch similar patterns. To help our method discriminate among these features, we employed a video signature, defined as:

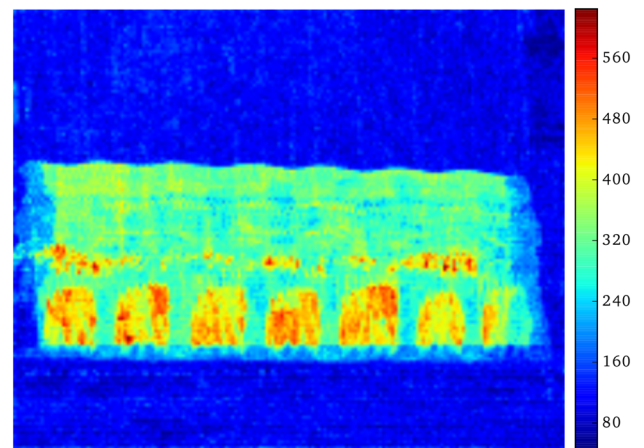
$$\text{Sig}(V) = \sum_{t=0}^{T-1} |V(t) - V(t + 1)| \tag{8}$$

where $V(t)$ returns the frame at time t , such that $\text{Sig}(V)$ is the sum of the absolute difference between consecutive frames of a video sequence V . This can be interpreted as the cumulative sum of a naive optical flow (see Fig. 14).

Our feature vector for the human action classification problem will be given by the oriented histograms for the most K relevant patterns and the oriented histograms of the signature previously normalized between 0 and 1.



(a)



(b)

Fig. 14 Signatures for **a** running and **b** walking actions

3.4.3 Classification

For the classification task, we trained a linear SVM with a variation of the Histogram Intersection Kernel (HIK). This kernel, known as the generalized HIK, has proved to be useful not only in image classification, but also in many other contexts [11]. It is defined as:

$$K(X, Y) = \sum \min(X^\alpha, Y^\beta) \tag{9}$$

where α and β are normalization parameters.

3.5 Gesture recognition

This problem is analogous to the human action classification problem, however, with the slight difference that we take a depth map into consideration. Thus, we will keep the core ideas for action recognition as well as the preprocessing and classification steps. Only one variation that is applicable in the preprocessing and the feature descriptor will be shortly discussed.

3.5.1 Preprocessing

In this stage, we aim at removing the background and selecting the most K relevant patterns for a video sequence. We used depth images to segment the background and foreground with the Triangle’s method [98]. Background removal will increase the chances of selecting more relevant patterns.

3.5.2 Feature descriptor

The feature descriptor is given by the concatenation of all oriented histograms for a pattern. We do not need to use the video signatures once the patterns obtained from the visual rhythms are strong enough to differentiate among all the classes, as will be shown in the experiments.

4 Experiments

This section presents and discusses the experimental results to analyze the performance of the visual rhythm according to our proposed method for the abnormal event detection, action classification, and gesture recognition problems.

All experiments were executed on an Intel(R) Core(TM) i7-3770K CPU @ 3.50 GHz with 32 GB RAM and 16 GB Swap, Linux version 3.11.0-19-generic (Ubuntu/Linaro) using Python with the following libraries: scipy and numpy [90], scikit-learn [67], and scikit-image [91]. Some heavy processes were executed parallelly (in particular, the training and extraction of codewords), however, for the other steps, we used a sequential programming approach.

4.1 UMN dataset for abnormal event detection

Description The UMN dataset [85] represents a dataset for abnormal global event detection problem that contains three escape scenarios of crowd people walking in any direction (the normal event) and suddenly they start to run simulating a panic scene (the abnormal event)—see Fig. 15. First scenario has 2 video sequences, whereas the second and third contain 6 and 3 video sequences respectively. All videos have a resolution of 320×240 color frames.

Evaluation methodology This dataset contains global events, which means that frames are classified into normal or abnormal frames. An abnormal frame contains an abnormal event, and, analogically, a normal frame does not. All comparisons are conducted through the AUC metric.

Since no standard protocols were provided to evaluate this dataset, we adopted the following criterion⁴: each video

⁴ Many works from the literature consider the first k frames for training, where k varies from 200 to 300.

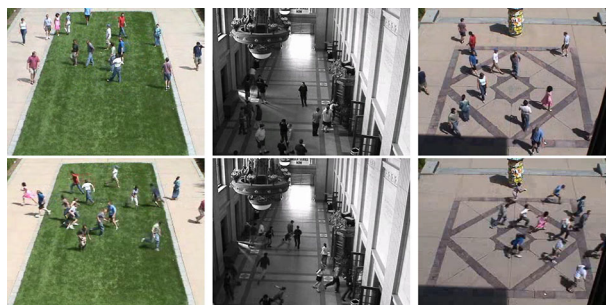


Fig. 15 Three different scenarios for the UMN dataset [85]. *Top figures* represent normal instances, and *bottom figures* are the abnormal event

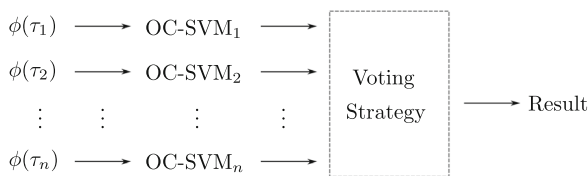


Fig. 16 Bagging-like approach adopted for the abnormal event detection problem. The AUC is obtained from making a decision at different threshold values

sequence was divided into training and testing sets, first 160 frames (the data training set) were assumed to contain just normal events. Moreover, we conducted two types of experiments that considered each video sequence and each scenario.

Multiple visual rhythms were extracted considering two different types of sequences π_1 and π_2 in order to capture more information. $\pi_1 = p(x, 1) \rightarrow p(x, 2) \rightarrow p(x, 3) \rightarrow \dots \rightarrow p(x, H)$, $\pi_2 = p(1, y) \rightarrow p(2, y) \rightarrow p(3, y) \rightarrow \dots \rightarrow p(W, y)$, where $H = 240$ and $W = 320$. The parameters x and y varied in intervals of 20 pixels such that we obtained 12 and 16 visual rhythms for π_1 and π_2 respectively.

We observed that the abnormal event was contextually defined by each one of them, which means that, for a particular visual rhythm, it is possible to have a different abnormal event behavior than other visual rhythm. Therefore, we decided to train one specific OC-SVM classifier for each visual rhythm. AUC values were calculated by considering a voting strategy, but it is not necessary to give an answer to the classification problem; however, that answer could be obtained by setting a certain threshold parameter so that it establishes the minimum number of votes that a data point needs to assign it as a normal or an abnormal event (see Fig. 16).

In the preprocessing step, the size of the structuring element used to remove the vertical lines was set to 11, since higher values remove lines that can be a part of the person’s trajectory, and smaller values are not very effective to remove the vertical lines as we would expect. Cells b_{ij} were assigned to 8×8 pixels, and each block contains the histogram of

Table 1 Experimental results for the abnormal event detection problem on the UMN dataset

Scenario	Video seq.	AUC ¹	AUC ²
1	Seq. 1	0.991	0.961
	Seq. 2	0.947	
2	Seq. 3	1.000	0.984
	Seq. 4	1.000	
	Seq. 5	0.997	
	Seq. 6	1.000	
	Seq. 7	1.000	
	Seq. 8	0.999	
3	Seq. 9	1.000	0.977
	Seq. 10	0.938	
	Seq. 11	0.938	
Average		0.983	0.974

AUC¹ values were calculated for each video sequence independently, and for AUC² values experiments were carried for each scenario

oriented gradients with 9 orientations. The codebook \mathcal{C} was learned using the K-Means algorithm with 30 centroids—each centroid represents a codeword. We used hard coding and sum pooling strategies for producing the final descriptor $\phi(\tau)$.

Table 1 shows the AUC values and Table 2 reports the running execution time for each video sequence, as well as for the three scenarios. Although we used the second experiment for comparison with the state-of-the-art because many approaches tend to use similar protocols, our results are still comparable to those of the literature (see Table 3) and they also achieve real-time performance—28.246 fps (frames-per-second) including all steps required by the method.

From the experiments, we observe that AUC values decreased if we compare each kind of experiment. To understand this phenomenon, we executed 150 times both experiments for video sequences 1 and 2 and for scenario 1. Once we plotted this data (Fig. 17), we see that weak classifiers are learned for the first experiment since we have only 20 data points for training; on the other hand, for the second experiment (with 40 data points) our classifiers are more robust, but on average, they produce worse results than in the previous case.

In an additional experiment, we evaluated the performance of the classifier by considering only half of the training data (80 frames per video) for the second scenario, once it has more available samples. We observed that the achieved AUC value was 0.985, which is very similar if all samples were used. This test confirms that using more data does not always produce better results.

Table 2 Total running time for the experiments on the UMN dataset

Scenario	Video seq.	# Frames	Time ¹	Time ²
1	Seq. 1	625	30.259	29.586
	Seq. 2	828	35.811	
2	Seq. 3	549	28.120	26.642
	Seq. 4	685	30.569	
	Seq. 5	768	33.747	
	Seq. 6	579	29.001	
	Seq. 7	895	37.628	
	Seq. 8	667	29.872	
3	Seq. 9	658	31.102	28.509
	Seq. 10	677	28.044	
	Seq. 11	808	34.891	
Average		703.545	31.731	28.246

Columns Time¹ and Time² show the frame rate (in frames per second) obtained on average with our solution for each type of experiment. In Time¹, we excluded the training time that each experiment requires, but Time² includes it

Table 3 State-of-the-art results for abnormal event detection on the UMN dataset

Approach	AUC
Optical flow [53]	0.840
Wang et al. [93]	0.928
Chen et al. [18]	0.940
Li et al. [46]	0.952
Biswas and Babu [7]	0.954
Mehran et al. [53]	0.960
Proposed method	0.974
Thida et al. [83]	0.977
Cong et al. [19]	0.980
Nam et al. [58]	0.983
Zhang et al. [99]	0.986
Tang et al. [81]	0.989
Saligrama and Chen [73]	0.995
Hung et al. [38]	1.000

4.2 Weizmann dataset for human action classification

Description The action recognition dataset [9] (known as the Weizmann dataset) contains video sequences showing 10 natural actions from 9 different actors. Table 4 shows sample frames from all the following actions defined in the dataset—running, walking, skipping, jumping-jack (jack), jumping-forward on two legs (jump), jumping-in-place on

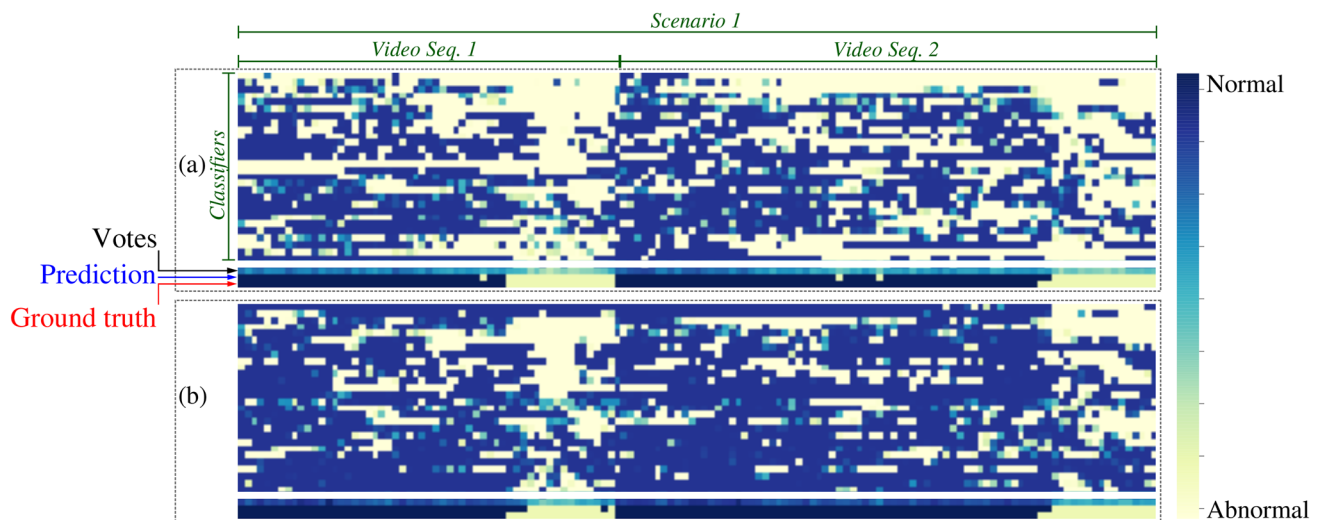


Fig. 17 The heat map shows the average prediction of running 150 times for: **a** first experiment with videos sequences 1 and 2, and **b** the second experiment with scenario 1. Dark blue indicates a normal event and light yellow an abnormal event

Table 4 Sample frames for different actions from Weizmann dataset [9]

Action	Sample video frames
Bend	
Jack	
Pjump	
Walk	
Wave1	
Wave2	
Jump	
Side	
Run	
Skip	

two legs (pjump), galloping-sideways (side), waving-one-hand (wave1), and waving-two-hands (wave2). All videos have a resolution of 180×144 color frames.

Evaluation methodology For comparative purposes, the established protocol used is the leave-one-out (LOO) cross-validation scheme.

Sequence π was defined as a set of horizontal cuts in intervals of 5 pixels such that we obtained 36 visual rhythms per video. We did not consider vertical visual rhythms because some actions are more discriminative in the Y -coordinates, and zigzag visual rhythm was not very discriminative to differentiate actions among some classes. In the pattern filtering step, we selected the best 5 patterns per video. Table 5 shows some samples for the filtered patterns and the video signature.

From each of these patterns, we obtained cells b_{ij} of 20×20 considering an overlap of 5 pixels and calculated the histograms of oriented gradients at 8 orientations. For the signatures, the oriented histograms were obtained from 21×21 cells, an overlap of 5 pixels, and 21 orientations. Experimentally, the normalization parameters for HIK kernel were setup to $\alpha = 0.75$ and $\beta = 0.7$.

Our method obtained an accuracy of 74.4 % from the visual rhythm patterns, while the signature resulted in 67.7 %. The combination of both features, visual rhythm patterns and signature, improved the accuracy to 78.89 %. The confusion matrix (Fig. 18) shows that patterns for bend, jack, pjump, wave1 and wave2 actions can be clearly distinguished, however, the weakness of the method resides in the walk, jump, side, run and skip actions. They contain similar patterns, making the differentiation among them more difficult.

A comparison with state-of-the-art algorithms is presented in Table 6, where accuracy rates and adopted protocols for training and testing are shown. We also report the execution time for each step of our method in Table 7. Note that the heaviest process locates on constructing the oriented histograms from the visual rhythm patterns and the video signature.

Table 5 Visual rhythm patterns obtained from Weizmann dataset

Action	Patterns			Signature
Bend				
Jack				
Pjump				
Walk				
Wave1				
Wave2				
Jump				
Side				
Run				
Skip				

bend	1	0	0	0	0	0	0	0	0	
jack	0	1	0	0	0	0	0	0	0	
pjump	0	0	1	0	0	0	0	0	0	
walk	0	0	0	0.56	0	0	0.22	0.22	0	
wave1	0	0	0	0	1	0	0	0	0	
wave2	0	0	0	0	0	1	0	0	0	
jump	0	0	0	0.11	0	0	0.89	0	0	
side	0	0	0	0.11	0	0	0.33	0.44	0.11	
run	0	0	0	0.11	0	0	0	0.67	0.22	
skip	0	0	0	0.11	0	0	0.11	0.33	0.11	
	bend	jack	pjump	walk	wave1	wave2	jump	side	run	skip

Fig. 18 Confusion matrix for Weizmann dataset

4.3 KTH dataset for human action classification

Description The KTH dataset [76] contains 2391 videos, categorized into six classes (walking, jogging, running, boxing, hand-waving, hand-clapping), where 25 actors wearing different clothing performed the actions in 4 scenarios with

Table 6 State-of-the-art results for human action classification on the Weizmann dataset

Approach	Accuracy (%)	Protocol
Blackburn and Ribeiro [8]	61.00	LOO ^a
Niebles and Fei-Fei [61]	72.80	LOO
Antonucci et al. [5]	74.70	LOO
Proposed method	78.89	LOO
Yang et al. [96]	87.00	Own setup
Wang et al. [92]	93.30	LOO
Gorelick et al. [34]	97.54	LOO
Fawzy et al. [30]	97.79	LOO
De Rosa et al. [22]	98.61	LOO
Alcantara et al. [3]	98.90	–
Guo et al. [36]	100.00	LOO
Schindler et al. [74]	100.00	LOO

^a Experiments reconducted by Almotairi [4]

Table 7 Execution times in seconds and frames per second (fps) for all required steps on the Weizmann dataset
















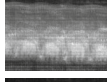



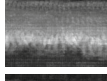



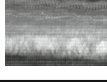
Steps	Time (s)	Time (fps)
Visual rhythm patterns	7.871	702.83
Signature	0.689	8,029.02
Features	11.809	468.43
Training	0.514	10,764.74
Testing	0.164	33,711.15

Table 8 Sample frames for different actions from KTH dataset [76]

Action	Sample video frames			
Boxing				
Hand-clapping				
Hand-waving				
Walking				
Running				
Jogging				

particular changes, making the dataset challenging due to outdoor/indoor scenes, scaling variation, and very shiny/dark videos. All video sequences have a spatial resolution of 160 × 120 pixels. Table 8 shows some frames extracted from the performed actions in the KTH dataset.

Table 9 Visual rhythm patterns obtained from KTH dataset

Action	Patterns			Signature
Boxing				
Hand-clapping				
Hand-waving				
Walking				
Running				
Jogging				

Evaluation methodology The evaluation protocol for KTH consists in partitioning the videos with respect to the subjects into training set (8 persons), validation set (8 persons), and test set (9 persons). Similar to what occurs in the Weizmann dataset, the walking, running and jogging actions have same visual rhythm patterns. The walking action is slightly different since it presents a well defined braided pattern, whereas running and jogging do not, they are noisy (see Table 9).

The feature vector was obtained by calculating the oriented histograms over the visual rhythm patterns and the signature images. For the patterns, we used cells of 20×20 , 5 overlapping pixels, and 20 orientations. The signatures were processed with cells of 18×18 pixels, 5 overlapping pixels between cells, and 15 orientations. Both HIK parameters are equal to 0.3.

The patterns resulted in an accuracy of 83.33 %, while its combination with the signature provided an accuracy of 87.96 %. The confusion matrix (Fig. 19) shows that the classification errors occur for running and jogging which was expected because of the similar patterns for those actions. A comparison with the state-of-the-art approaches is shown in Table 10. The execution time of our method is presented in Table 11.

4.4 SKIG dataset for gesture recognition

Description The Sheffield Kinect Gesture (SKIG) dataset [47] contains 2160 hand gesture videos captured with a Kinect sensor (1080 RGB and 1080 depth videos). There are the following 10 classes: circle, triangle, up-down, right-left, wave, Z, cross, come here, turn-around, and pat. Videos were recorded with 3 different backgrounds and 2 illumination conditions (Table 12).

boxing	1	0	0	0	0	0
handclapping	0.028	0.97	0	0	0	0
handwaving	0	0.056	0.94	0	0	0
walking	0	0	0	1	0	0
running	0	0	0	0.056	0.72	0.22
jogging	0	0	0	0.28	0.083	0.64
	boxing	handclapping	handwaving	walking	running	jogging

Fig. 19 Confusion matrix for KTH dataset**Table 10** State-of-the-art results for human action classification on the KTH dataset

Approach	Accuracy (%)	Protocol
Ke et al. [42]	62.96	Split
Schuldt et al. [76]	71.72	Split
Antonucci et al. [5]	72.50	LOO
Yang et al. [96]	75.71	Own setup
Dollar et al. [25]	81.16	LOO
De Rosa et al. [22]	83.20	LOO
Niebles et al. [62]	83.30	LOO
Raja et al. [69]	86.60	Split
Wong and Cipolla [94]	86.20	LOO
Proposed method	87.90	Split
Ji et al. [39]	90.20	Own setup
Fathi et al. [29]	90.50	Own setup
Alcantara et al. [3]	91.30	–
Laptev et al. [45]	91.80	Split
Schindler et al. [74]	92.70	5-Fold
Sun et al. [79]	94.00	LOO
Guo et al. [36]	98.50	LOO

Table 11 Execution times in seconds and frames per second (fps) for all required steps on the KTH dataset

Steps	Time (s)	Time (fps)
Visual rhythm patterns	199.537	1452.27
Signature	69.146	4190.86
Features	65.546	4421.05
Training	4.778	60,655.36
Testing	2.754	105,233.32

Evaluation methodology The protocol for testing is defined as a 3-fold cross-validation scheme.

Features are constructed by concatenating oriented histograms for each pattern with cells of 18×18 , 5 pixels

Table 12 Sample frames for different actions from SKIG dataset [47]

Action	Sample video frames			
Circle				
Triangle				
Up-down				
Right-left				
Wave				
Z				
Cross				
Come here				
Turn-around				
Pat				

Table 13 Visual rhythm patterns obtained from SKIG dataset

Action	Patterns		
Circle			
Triangle			
Up-down			
Right-left			
Wave			
Z			
Cross			
Come-here			
Turn-around			
Pat			

overlapping pixels, and 18 orientations. The normalization parameters of the HIK kernel are equal to 0.35.

As seen in Table 13, the extracted patterns from the visual rhythm are distinct for each class. The proposed method achieves an accuracy of 97.96 % even with the visual similitude for the wave and right-left patterns, our oriented histograms can describe the thickness on the right-left patterns. The confusion matrix (Fig. 20) shows that almost all classes are classified with a small number of false positives.

A comparison with state-of-the-art approaches is presented in Table 14 and the corresponding execution time in Table 15.

5 Conclusions and future work

The visual rhythm provides information that can be fully exploited to solve many problems in computer vision that demand a trade-off between accuracy and speed. Detection of abnormal events and classification of actions set are clear examples of this fact, which through the proposed general framework, we achieve state-of-the-art results with efficiency in terms of time complexity.

circle	0.99	0	0	0	0.0093	0	0	0	0	
triangle	0	1	0	0	0	0	0	0	0	
updown	0.0093	0	0.94	0	0	0	0	0.037	0.0093	
rightleft	0	0	0	0.96	0.028	0	0	0.0093	0	
wave	0	0	0	0	1	0	0	0	0	
z	0	0	0	0	0	1	0	0	0	
cross	0	0	0	0	0	0.0093	0.99	0	0	
comehere	0	0	0	0	0	0	0.0093	0.96	0.028	
turnaround	0.019	0	0.0093	0	0	0	0	0	0.97	
pat	0	0	0.019	0	0	0	0	0.0093	0	0.97
	circle	triangle	updown	rightleft	wave	z	cross	comehere	turnaround	pat

Fig. 20 Confusion matrix for SKIG dataset

Experiments conducted on public datasets not only demonstrate our promising results, but also explain how the proposed methodology addresses the problems and the reasons of failure and success. The visual rhythm was showed to be a powerful source of information that makes some problems

Table 14 State-of-the-art results for human action classification on the SKIG dataset

Approach	Accuracy (%)	Protocol
Liu and Shao [47]	88.7	3-Fold
Moreira et al. [56]	93.5	3-Fold
Yu et al. [97]	93.7	Own setup
Tung and Ngoc [84]	96.5	10-Fold
De Rosa et al. [22]	97.5	3-Fold
Nishida and Nakayama [63]	97.8	3-Fold
Proposed method	97.9	3-Fold
Molchanov et al. [55]	98.6	–

Table 15 Execution times in seconds and frames per second (fps) for all required steps on the SKIG dataset

Steps	Time (s)	Time (fps)
Visual rhythm patterns	259.485	2416.37
Features	13.030	48,121.01
Training	6.672	93,973.80
Testing	2.211	283,600.34

easier to be solved since we reduce the issue of working with video sequences to images.

Even though that our proposal for the abnormal event detection works under the assumption of having global events, the case of detecting local events still needs further research in order to have an approach that uses the visual rhythm as local features so that we can detect local anomalies.

The visual rhythm for the human action classification and gesture recognition problems described relevant information to discriminate among actions. However, some actions produced similar patterns and these were difficult to differentiate. To address this drawback, features from the video signatures were obtained to have a naive perception of motion in the space, but different paths π must be explored so that they could perform better when comparing patterns of distinct actions.

As directions for future work, we intend to explore new methods and techniques of processing the visual rhythm in other video analysis tasks. We conjecture that it is still possible to improve our results and even further investigate local feature descriptors from the visual rhythm.

Acknowledgements The authors are grateful to FAPESP, CNPq and CAPES for the financial support.

References

- Aggarwal, J.K., Ryoo, M.S.: Human activity analysis: a review. *ACM Comput. Surv.* **43**(3), 16 (2011)
- Alcantara, M., Moreira, T., Pedrini, H.: Real-time action recognition based on cumulative motion shapes. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2917–2921 (2014)
- Alcantara, M., Moreira, T., Pedrini, H.: Real-time action recognition using a multilayer descriptor with variable size. *J. Electron. Imaging* **25**(1), 013,020–013,020 (2016)
- Almotairi, S.M.: Using variations of shape and appearance in alignment methods for classifying human actions. *Florida Institute of Technology, Melbourne* (2014)
- Antonucci, A., De Rosa, R., Giusti, A., Cuzzolin, F.: Robust classification of multivariate time series by imprecise hidden Markov models. *Int. J. Approx. Reason.* **56**, 249–263 (2015)
- Berent, J., Dragotti, P.: Segmentation of epipolar-plane image volumes with occlusion and disocclusion competition. In: *IEEE 8th Workshop on Multimedia Signal Processing*, pp. 182–185 (2006)
- Biswas, S., Babu, R.V.: Real time anomaly detection in H.264 compressed videos. In: *Fourth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics*, pp. 1–4. *IEEE* (2013)
- Blackburn, J., Ribeiro, E.: Human motion recognition using isomap and dynamic time warping. In: *Human Motion: Understanding, Modeling, Capture and Animation*, pp. 285–298. *Springer* (2007)
- Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. In: *International Conference on Computer Vision, Beijing*, pp. 1395–1402 (2005)
- Bolles, R.C., Baker, H.H.: Epipolar-plane image analysis: a technique for analyzing motion sequences. In: *3th IEEE Workshop on Computer Vision, Representation, and Control*, pp. 168–178. *IEEE* (1985)
- Boughorbel, S., Tarel, J.P., Boujemaa, N.: Generalized histogram intersection kernel for image recognition. In: *IEEE International Conference on Image Processing*, vol. 3, pp. III–161. *IEEE* (2005)
- Bourke, A., O'Brien, J., Lyons, G.: Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. *Gait Posture* **26**(2), 194–199 (2007)
- Buch, N., Velastin, S., Orwell, J.: A review of computer vision techniques for the analysis of urban traffic. *IEEE Trans. Intell. Transp. Syst.* **12**(3), 920–939 (2011)
- Calderara, S., Heinemann, U., Prati, A., Cucchiara, R., Tishby, N.: Detecting anomalies in people's trajectories using spectral graph analysis. *Comput. Vis. Image Underst.* **115**(8), 1099–1111 (2011)
- Candamo, J., Shreve, M., Goldgof, D., Sapper, D., Kasturi, R.: Understanding transit scenes: a survey on human behavior-recognition algorithms. *IEEE Trans. Intell. Transp. Syst.* **11**(1), 206–224 (2010)
- Canny, J.: A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **6**, 679–698 (1986)
- Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: a survey. *ACM Comput. Surv.* **41**(3), 1–58 (2009)
- Chen, D.Y., Huang, P.C.: Motion-based unusual event detection in human crowds. *J. Vis. Commun. Image Represent.* **22**(2), 178–186 (2011)
- Cong, Y., Yuan, J., Liu, J.: Abnormal event detection in crowded scenes using sparse representation. *Pattern Recognit.* **46**(7), 1851–1864 (2013)
- Cui, L., Li, K., Chen, J., Li, Z.: Abnormal event detection in traffic video surveillance based on local features. In: *4th International Congress on Image and Signal Processing*, vol. 1, pp. 362–366. *IEEE* (2011)
- Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 886–893 (2005)
- De Rosa, R., Cesa-Bianchi, N., Gori, I., Cuzzolin, F.: Online action recognition via nonparametric incremental learning. In: *British Machine Vision Conference*. *BMVA Press* (2014)

23. Dee, H.M., Velastin, S.A.: How close are we to solving the problem of automated visual surveillance? *Mach. Vis. Appl.* **19**(5–6), 329–343 (2007)
24. Devanne, M., Wannous, H., Berretti, S., Pala, P., Daoudi, M., Del Bimbo, A.: Space-time pose representation for 3D human action recognition. In: *International Conference on Image Analysis and Processing*, pp. 456–464. Springer (2013)
25. Dollár, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: *IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pp. 65–72. IEEE (2005)
26. Duda, R.O., Hart, P.E.: Use of the Hough transformation to detect lines and curves in pictures. *Commun. ACM* **15**(1), 11–15 (1972)
27. Fanello, S.R., Gori, I., Metta, G., Odone, F.: Keep it Simple and Sparse: Real-Time Action Recognition. *Journal of Machine Learning Research* **14**(1), 2617–2640 (2013)
28. Farneback, G.: Two-frame motion estimation based on polynomial expansion. In: *Image Analysis*, pp. 363–370. Springer (2003)
29. Fathi, A., Mori, G.: Action recognition by learning mid-level motion features. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8. IEEE (2008)
30. Fawzy, F., Abdelwahab, M., Mikhael, W.: 2DHOOF-2DPCA contour based optical flow algorithm for human activity recognition. In: *IEEE 56th International Midwest Symposium on Circuits and Systems*, pp. 1310–1313 (2013)
31. Feng, J., Zhang, C., Hao, P.: Online learning with self-organizing maps for anomaly detection in crowd scenes. In: *20th International Conference on Pattern Recognition*, vol. 1, pp. 3599–3602 (2010)
32. Fortun, D., Bouthemy, P., Kervrann, C.: Optical flow modeling and computation: a survey. *Comput. Vis. Image Underst.* **134**, 1–21 (2015)
33. Gong, S., Xiang, T.: Person re-identification. In: *Visual Analysis of Behaviour*, pp. 301–313. Springer (2011)
34. Gorelick, L., Blank, M., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(12), 2247–2253 (2007)
35. Guimarães, S., de A.-Araújo, A., Couprie, M., Leite, N.: An Approach to detect video transitions based on mathematical morphology. In: *International Conference on Image Processing*, vol. 3, pp. II-1021–4 (2003)
36. Guo, K., Ishwar, P., Konrad, J.: Action recognition from video using feature covariance matrices. *IEEE Trans. Image Process.* **22**(6), 2479–2494 (2013)
37. Horn, B.K., Schunck, B.G.: Determining optical flow. In: *1981 Technical Symposium East*, pp. 319–331. International Society for Optics and Photonics (1981)
38. Hung, T.Y., Lu, J., Tan, Y.P.: Cross-scene abnormal event detection. In: *IEEE International Symposium on Circuits and Systems*, pp. 2844–2847 (2013)
39. Ji, S., Xu, W., Yang, M., Yu, K.: 3D Convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(1), 221–231 (2013)
40. Jiang, F., Yuan, J., Tsafaris, S.A., Katsaggelos, A.K.: Anomalous video event detection using spatiotemporal context. *Comput. Vis. Image Underst.* **115**(3), 323–333 (2011)
41. Jiang, X., Zhong, F., Peng, Q., Qin, X.: Online robust action recognition based on a hierarchical model. *Vis. Comput.* **30**(9), 1021–1033 (2014)
42. Ke, Y., Sukthankar, R., Hebert, M.: Efficient visual event detection using volumetric features. In: *Tenth IEEE International Conference on Computer Vision*, vol. 1, pp. 166–173. IEEE (2005)
43. Kliper-Gross, O., Hassner, T., Wolf, L.: The action similarity labeling challenge. *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(3), 615–621 (2012)
44. Laptev, I.: On space-time interest points. *Int. J. Comput. Vis.* **64**(2–3), 107–123 (2005)
45. Laptev, I., Marszałek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8. IEEE (2008)
46. Li, W., Mahadevan, V., Vasconcelos, N.: Anomaly detection and localization in crowded scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* **36**(1), 18–32 (2014)
47. Liu, L., Shao, L.: Learning Discriminative representations from RGB-D video data. In: *International Joint Conference on Artificial Intelligence*, vol. 1, p. 3 (2013)
48. Lowe, D.: Object recognition from local scale-invariant features. In: *Computer Vision*, vol. 2, pp. 1150–1157 (1999)
49. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. *Int. Jt. Conf. Artif. Intell.* **81**, 674–679 (1981)
50. Mahadevan, V., Li, W., Bhalodia, V., Vasconcelos, N.: Anomaly detection in crowded scenes. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1975–1981 (2010)
51. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online dictionary learning for sparse coding. In: *26th Annual International Conference on Machine Learning*, pp. 689–696. ACM (2009)
52. McCahill, M., Norris, C.: CCTV systems in London: their structures and practices. Tech. rep., Centre for Criminology and Criminal Justice, University of Hull (2003)
53. Mehran, R., Oyama, A., Shah, M.: Abnormal crowd behavior detection using social force model. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 935–942 (2009)
54. Mitiche, A.: Rigid body kinematics: some basic notions. In: *Computational Analysis of Visual Motion. Advances in Computer Vision and Machine Intelligence*, pp. 31–43. Springer, US (1994)
55. Molchanov, P., Yang, X., Gupta, S., Kim, K., Tyree, S., Kautz, J.: Online Detection and classification of dynamic hand gestures with recurrent 3D convolutional neural network. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4207–4215 (2016)
56. Moreira, T., Alcantara, M., Pedrini, H., Menotti, D.: Fast and accurate gesture recognition based on motion shapes. In: *Iberoamerican Congress on Pattern Recognition*, pp. 247–254. Springer (2015)
57. Nalwa, V.S.: *A Guided Tour of Computer Vision*. Addison-Wesley Longman Publishing Co. Inc., Boston (1993)
58. Nam, Y.: Crowd flux analysis and abnormal event detection in unstructured and structured scenes. *Multimed. Tools Appl.* **72**(3), 3001–3029 (2014)
59. Ngo, C., Pong, T., Chin, R.: Detection of gradual transitions through temporal slice analysis. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1 (1999)
60. Ngo, C.W., Pong, T.C., Zhang, H.J.: Motion analysis and segmentation through spatio-temporal slices processing. *IEEE Trans. Image Process.* **12**(3), 341–355 (2003)
61. Niebles, J.C., Fei-Fei, L.: A hierarchical model of shape and appearance for human action classification. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8. IEEE (2007)
62. Niebles, J.C., Wang, H., Fei-Fei, L.: Unsupervised learning of human action categories using spatial-temporal words. *Int. J. Comput. Vis.* **79**(3), 299–318 (2008)
63. Nishida, N., Nakayama, H.: Multimodal gesture recognition using multi-stream recurrent neural network. In: *Pacific-Rim Symposium on Image and Video Technology*, pp. 682–694. Springer (2015)
64. Niyogi, S., Adelson, E.: Analyzing and recognizing walking figures in XYT. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 469–474 (1994)

65. Otsu, N.: A threshold selection method from gray-level histograms. *Automatica* **11**(285–296), 23–27 (1975)
66. Ozturk, O., Yamasaki, T., Aizawa, K.: Detecting dominant motion flows in unstructured/structured crowd scenes. In: 20th International Conference on Pattern Recognition, pp. 3533–3536 (2010)
67. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
68. Piciarelli, C., Foresti, G.: On-line trajectory clustering for anomalous events detection. *Pattern Recognit. Lett.* **27**(15), 1835–1842 (2006)
69. Raja, K., Laptev, I., Pérez, P., Oisel, L.: Joint pose estimation and action recognition in image graphs. In: 18th IEEE International Conference on Image Processing, pp. 25–28. IEEE (2011)
70. Ran, Y.: Symmetry in Human motion analysis: theory and experiment. Ph.D. thesis, University of Maryland (2006)
71. Rautaray, S.S., Agrawal, A.: Vision based hand gesture recognition for human computer interaction: a survey. *Artif. Intell. Rev.* **43**(1), 1–54 (2015)
72. da S. Pinto, A., Pedrini, H., Schwartz, W., Rocha, A.: Video-based face spoofing detection through visual rhythm analysis. In: 25th SIBGRAPI Conference on Graphics, Patterns and Images, pp. 221–228 (2012)
73. Saligrama, V.: Video anomaly detection based on local statistical aggregates. In: IEEE Conference on Computer Vision and Pattern Recognition pp. 2112–2119 (2012)
74. Schindler, K., van Gool, L.: Action snippets: how many frames does human action recognition require? In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008)
75. Schölkopf, B., Williamson, R.C., Smola, A.J., Shawe-Taylor, J., Platt, J.C.: Support vector method for novelty detection. *NIPS* **12**, 582–588 (1999)
76. Schüldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local SVM approach. In: 17th International Conference on Pattern Recognition, vol. 3, pp. 32–36. IEEE (2004)
77. Prison Service Order.: Display screen equipment health and safety issues. H.M. Prison Service (2000)
78. Sobral, A., Vacavant, A.: A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *Comput. Vis. Image Underst.* **122**, 4–21 (2014)
79. Sun, X., Chen, M., Hauptmann, A.: Action Recognition via local descriptors and holistic features. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 58–65. IEEE (2009)
80. Suriani, N.S., Hussain, A., Zulkifley, M.A.: Sudden event recognition: a survey. *Sensors* **13**(8), 9966–9998 (2013)
81. Tang, X., Zhang, S., Yao, H.: Sparse Coding based motion attention for abnormal event detection. In: 20th IEEE International Conference on Image Processing, pp. 3602–3606 (2013)
82. Tax, D.M., Duin, R.P.: Support vector data description. *Mach. Learn.* **54**(1), 45–66 (2004)
83. Thida, M., Eng, H.L., Remagnino, P.: Laplacian eigenmap with temporal constraints for local abnormality detection in crowded scenes. *IEEE Trans. Cybern.* **43**(6), 2147–2156 (2013)
84. Tung, P.T., Ngoc, L.Q.: Elliptical density shape model for hand gesture recognition. In: Fifth Symposium on Information and Communication Technology, pp. 186–191. ACM (2014)
85. UMN—Detection of Unusual Crowd Dataset (2015) <http://mha.cs.umn.edu/>
86. Valio, F.B., Pedrini, H., Leite, N.J.: Fast rotation-invariant video caption detection based on visual rhythm. In: San Martin, C., Kim, S.W. (eds.) Progress in pattern recognition, image analysis, computer vision, and applications, Lecture notes in computer science, pp. 157–164. Springer, Berlin, Heidelberg (2011)
87. Vishwakarma, S., Agrawal, A.: A survey on activity recognition and behavior understanding in video surveillance. *Vis. Comput.* **29**(10), 983–1009 (2013)
88. Vo, D.H., Huynh, H.H., Meunier, J.: Geometry-based dynamic hand gesture recognition. *J. Sci. Technol.* **1**, 13–19 (2015)
89. Wallace, E., Diffley, C., Britain, G.: CCTV: Making it work: CCTV control room ergonomics. Publication (Great Britain. Home Office. Police Scientific Development Branch). Police Scientific Development Branch (1998)
90. van der Walt, S., Colbert, S.C., Varoquaux, G.: The numpy array: a structure for efficient numerical computation. *Comput. Sci. Eng.* **13**(2), 22–30 (2011)
91. van der Walt, S., Schönberger, J.L., Nunez-Iglesias, J., Boulogne, F., Warner, J.D., Yager, N., Gouillart, E., Yu, T.: The scikit-image contributors: scikit-image: image processing in Python. *PeerJ* **2**, e453 (2014)
92. Wang, S., Huang, K., Tan, T.: A Compact optical flow based motion representation for real-time action recognition in surveillance scenes. In: 16th IEEE International Conference on Image Processing, pp. 1121–1124 (2009)
93. Wang, T., Chen, J., Snoussi, H.: Online detection of abnormal events in video streams. *J. Electr. Comput. Eng.* **2013**, 1–12 (2013)
94. Wong, S.F., Cipolla, R.: Extracting spatiotemporal interest points using global information. In: IEEE 11th International Conference on Computer Vision, pp. 1–8. IEEE (2007)
95. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: a benchmark. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2411–2418 (2013)
96. Yang, W., Wang, Y., Mori, G.: Human action recognition from a single clip per action. In: IEEE 12th International Conference on Computer Vision Workshops, pp. 482–489 (2009)
97. Yu, M., Liu, L., Shao, L.: Structure-preserving binary representations for RGB-D action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(8), 1651–1664 (2016)
98. Zack, G., Rogers, W., Latt, S.: Automatic measurement of sister chromatid exchange frequency. *J. Histochem. Cytochem.* **25**(7), 741–753 (1977)
99. Zhang, Y., Qin, L., Yao, H., Huang, Q.: Abnormal crowd behavior detection based on social attribute-aware force model. In: 19th IEEE International Conference on Image Processing, pp. 2689–2692 (2012)
100. Zhao, B., Fei-Fei, L., Xing, E.P.: Online detection of unusual events in videos via dynamic sparse coding. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3313–3320. IEEE Computer Society (2011)



Berthin S. Torres is currently a Master's student in Computer Science at the University of Campinas, Brazil. He received his B.Eng. in Informatics Engineering from the National University of San Antonio Abad del Cusco, Peru. He participated in programming competitions and awarded an internship at Facebook. His research interests include image processing, computer vision, machine learning, pattern recognition, algorithms and complexity.



ing, and computer graphics.

Helio Pedrini is currently a professor in the Institute of Computing at the University of Campinas, Brazil. He received his Ph.D. degree in Electrical and Computer Engineering from Rensselaer Polytechnic Institute, Troy, NY, USA. He received his M.Sc. in Electrical Engineering and his B.Sc. in Computer Science, both degrees from the University of Campinas, Brazil. His research interests include image processing, computer vision, pattern recognition, machine learn-