Spatio-temporal descriptors for human action recognition

Sameh Megrhi

Sorbonne Paris Cité
Université Paris 13- Institut Galilée

PHD THESIS
in candidacy for the degree of

Doctor of Paris 13 University

speciality: Network and Information Technologies

Defended by:
MEGRHI Sameh

Spatio-temporal descriptors for human recognition recognition

Prepared at Paris 13, L2TI
Defended in December xx, 2014

Jury:

Reviewers:

Prof. Stefania Colonnese, *Sapienza University di Roma, Italy*
Prof. TiTus ZAHARIA, *Télécom Sud Paris, France* .

Examiners

Prof. Emmanuel Viennet, *Université Paris 13, France* .
Dr. Faouzi Alaya Cheikh, *Gøvick University, Norway* .

Advisors

Prof. Azeddine BEGHDADI, *Université Paris 13, France* .
Dr. Wided Souidène, *Université Paris 13, France* .

Contents

Résumé

En raison de la demande croissante des systèmes d'analyse vidéo, la reconnaissance ainsi que la détection de l'action humaine sont ciblées par les chercheurs. L'objectif étant de réaliser une description précise et rapide de la vidéo, essentielement dans les grandes bases de données. Ainsi, le but ultime de la reconnaissance de l'action humaine sur les vidéos est de déterminer de manière automatique ce qui se passe dans une vidéo donnée.
Cette thèse vise à répondre à cette question en apportant une contribution dans la phase de détection et la phase de reconnaissance d'actions. Dans cet esprit, nous introduisons de nouvelles méthodes de description de reconnaissance de l'action humaine.

Pour la partie détection des actions, nous avons introduit deux approches basées sur les points d'intérêts locaux. La première proposition est une méthode simple et efficace qui vise à détecter les mouvements humains ensuite contribuer à extraire des séquences vidéo décrivant des actions importantes. Afin d'atteindre cet objectif, les premières séquences vidéo sont segmentées en volumes de trames et groupes de points déintérêt . Dans cette méthode, nous nous basons sur le suivi du mouvement des points d'intért. Nous avons utilisé, dans un premier lieu, des vidéos simples puis nous avons progressivement augmenté la complexité des vidéos en optant pour des vidéos réalistes.

Les jeux de données simples présentent généralement un arrière-plan statique avec un seul acteur qui effectue une seule action unique ou bien la même action mais d'une manière répétitive. Nous avons ensuite testé la robustesse de la détection d'action proposée dans des jeux de données plus complexes réalistes recueillis à partir des réseaux sociaux.

Nous avons introduit une approche de détection d'actions efficace pour résoudre le problème de la reconnaissance d'actions humaines dans les vidéos réalistes contenant des mouvements de caméra et n'étant pas de bonne qualité. Le mouvement humain est donc segmenté d'une manire spatio-temporelle afin de détecter le nombre optimal de trames suffisant pour effectuer une description vidéo.

Pour ce qui est du volet de la description, nous avons proposè dans cette thése deux nouveaux descripteurs spatio-temporels. Ces descripteurs sont basés sur le suivi de la

trajectoire des points d'intérêt. Les suivis et la description vidéo sont effectués sur les patchs vidéo qui contiennent un mouvement ou une partie d'un mouvement détecté par la segmentation réalisée lors de l'étape précédante. Nous nous sommes basé sur le descripteur SURF non seulement pour son efficacité et mais essentiellement pour la rapidité de son extraction. Le premier descripteur proposé est appel ST-SURF, il est basé sur une nouvelle combinaison du (SURF) et du flot optique. Le ST-SURF permet le suivi de la trajectoire des points d'intérêt tout en gardant les informations spatiales, pertinentes, provenant du SURF.

Le deuxième descripteur proposé dans le cadre de cette thèse est un histogramme du mouvement de la trajectoire (HMTO). HMTO est basé sur la position ainsi que l'échelle relative à un SURF. Ainsi, pour chaque SURF détecté, nous définissons une région du voisinage du point d'intérêt en nous basant sur l'échelle. Pour le patch détecté, nous extrayons le flux otique d'une manière dense. Les trajectoires de mouvement sont ensuite générées pour chaque pixel en exploitant les composantes horizontale et verticale de flux optique (u, v). La précision de la description de la vidéo proposée est testée sur un ensemble de données complexe et un plus grand ensemble de données réalistes. Les descripteurs de vidéo proposés sont testés d'une manière simple puis en les fusionnants avec d'autres descripteurs. Les descripteurs vidéo ont été introduit dans un processus de classification basée sur le sac de mots et ont démontré une amélioration des taux de reconnaissance par rapport aux approches précédemment proposées dans l'état-de-l'art.

Abstract

Due to increasing demand for video analysis systems in recent years, human action detection/recognition is being targeted by the research community in order to make video description more accurate and faster, especially for big datasets. The ultimate purpose of human action recognition is to discern automatically what is happening in any given video. This thesis aims to achieve this purpose by contributing to both action detection and recognition tasks. We thus have developed new description methods for human action recognition.

For the action detection component we introduce two novel approaches for human action detection. The first proposition is a simple yet effective method that aims at detecting human movements. First, video sequences are segmented into Frame Packets (FPs) and Group of Interest Points (GIP). In this method we track the movements of Interest Points in simple controlled video datasets and then in videos of gradually increasing complexity. The controlled datasets generally contain videos with a static background and simple actions performed by one actor. The more complex realistic datasets are collected from social networks.

The second approach for action detection attempts to address the problem of human action recognition in realistic videos captured by moving cameras. This approach works by segmenting human motion, thus investigating the optimal sufficient frame number to perform action recognition. Using this approach, we detect object edges using the canny edge detector. Next, we apply all the steps of the motion segmentation process to each frame. Densely distributed interest points are detected and extracted based on dense SURF points with a temporal step of N frames. Then, optical flows of the detected key points between two frames are computed by the iterative Lucas and Kanade optical flow technique, using pyramids.

Since we are dealing with scenes captured by moving cameras, the motion of objects necessarily involves the background and/or the camera motion. Hence, we propose to com-

pensate for the camera motion. To do so, we must first assume that camera motion exists if most points move in the same direction. Then, we cluster optical flow vectors using a KNN clustering algorithm in order to determine if the camera motion exists. If it does, we compensate for it by applying the affine transformation to each frame in which camera motion is detected, using as input parameters the camera flow magnitude and deviation. Finally, after camera motion compensation, moving objects are segmented using temporal differencing and a bounding box is drawn around each detected moving object. The action recognition framework is applied to moving persons in the bounding box. Our goal is to reduce the amount of data involved in motion analysis while preserving the most important structural features. We believe that we have performed action detection in the spatial and temporal domain in order to obtain better action detection and recognition while at the same time considerably reducing the processing time.

For the description component, we propose two novel spatio-temporal descriptors. Both of them are based on the tracking of interest points' trajectories. The tracking and the video description are performed on the detected relevant video patches that describe significant motion. We have chosen the Speed up Robust Feature (SURF) due to its efficiency and rapidity with the extraction step. In the context of video description, the ability to perform accurate action recognition more quickly is most appreciated by the research community. This is especially true since recently the focus has been oriented toward increasingly large realistic datasets. In sum, this first proposed descriptor is called ST-SURF (Spatio-Temporal SURF). It is based on a novel combination of the speed up robust feature (SURF) and the optical flow. The ST-SURF allows the tracking of interest points' trajectory while capturing spatial information provided by the detected SURF.

The second proposed descriptor is called the histogram of motion trajectory orientation (HMTO). HMTO is based on the SURF region, position and scale. Hence, for every detected SURF, we define the interest point neighborhood size related to the actual scale. For the detected patch, we extract a dense displacement field based on the optical flow algorithm. Motion trajectory orientations are then generated for every pixel by exploiting horizontal and vertical optical flow components (u, v). We split the optical flow components to extract the distribution of the motion trajectory orientation in the planes (t, x) and (t, y). The generated histograms describe the distribution of the trajectory orientation angle and its displacement into what we called selective snippets (SS). The SS are the extracted

relevant video patches that describe a human action portion or part based on the bounding boxes. The proposed video description accuracy is tested over complex datasets and over large realistic datasets. The proposed video descriptors are tested in a single, fusion-based video description. The resulting video descriptors are introduced in a classification process based on the bag-of- words and demonstrate significantly improved recognition rates over previously proposed approaches.

CHAPTER 1

INTRODUCTION

Success is to be measured not so much by the position that one has reached in life as by the obstacles which he has overcome.
Booker T. Washington

Contents

Action recognition in videos focuses on exploring human behavior. It has been developed in order to duplicate the capacity of natural human vision to understand automatically the surrounding environment. Since the 1980s, it has functioned by extracting useful information from a scene in an image or a video. In recent years, a large number of innovative feature extraction approaches have been proposed. Using these methods, researchers extract a representation of the scene content to yield descriptors. Machine learning algorithms are then employed to analyze these descriptors in order to achieve a specific purpose, for instance, the recognition of actions or objects within a video, as shown in Figure 1.1.

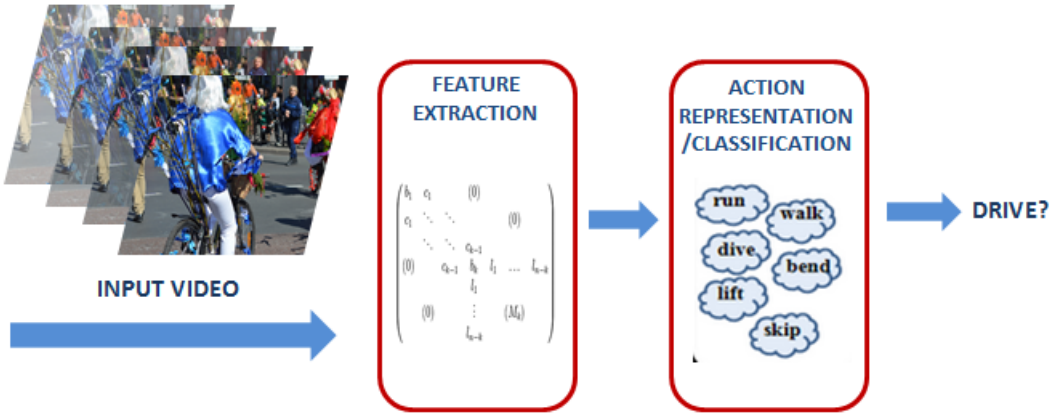
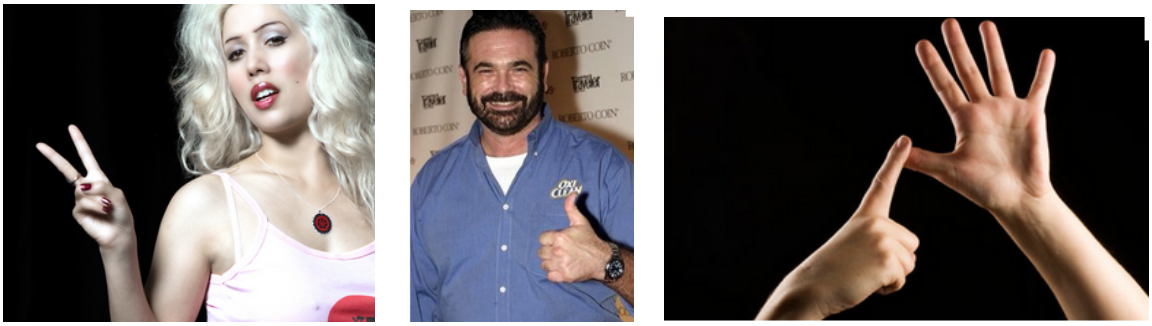


Figure 1.1. Human action recognition process

The recognition of human actions from videos is receiving increasing attention due to its wide range of applications, such as video indexing and retrieval [1], human-computer interaction, digital entertainment, and surveillance videos [2] etc. A key question now arises: What is the definition of a human action?
Several surveys were have been conducted for the purpose of answering this question. Among the most important of these surveys conducted in recent years are those by to Aggarwal and Cai in 1997 [3], Gavrila in 1999 [4], Wang and Singh with their work on 2003 [5], Buxton on 2003 [6], Aggarwal and Park on 2004 [7] , Turaga et al. in 2008 [8], and Aggarwal and Ryoo in 2011 [9]. As a result of their studies, we can subdivide "human body motion", which is a broader category than "human action", into four main categories: gestures, action, activity and events.

- At the most basic level, gestures are important components of human actions. They are generally short in temporal duration. They consist of a form of nonverbal com-

munication of the visible parts of the body in order to provide specific messages. Gestures include movements of the hands, head or other body parts. Some examples include spontaneous hand movements when talking, or those intentional movements used in sign language or to signify a particular message, such as the "victory" sign made by the fingers as shown in Figure 1.2.



(a) (b) (c)

Figure 1.2. Some known human gestures (a) Victory gesture ; (b) Chapeau gesture; (c) Sign language gesture

- An action is performed by an actor. One action requires a combination of a number of gestures in order to complete a specific task, such as a "phone call", See Figure 1.3.

Figure 1.3. Human action "phone call"

- An activity is a higher level of human movements. One activity can involve several humans and objects, as demonstrated by the examples of playing football or dancing shown in Figure 2.11. An activity, like an action, necessarily involves actors. For example, raining is neither an action nor an activity, since it does not involve actors. An activity is the combination of temporally ordered actions. There are many levels

of complexity in activities. The most complex activities involve combinations of other smaller or simpler activities.

(a) (b)

Figure 1.4. Some known human activities (a) Football ; (b) Dancing

- Events are activities that take place at a specific time. That is, each event has a beginning and an ending time. Usually, events are characterized by their temporal extents, not by the existing actor or the spatial information. An example of an event could be an actorin this case, a hair stylistgiving someone a haircut

Action recognition in videos is usually confronted to by many issues, including the necessity of handling considerable occlusions, scale changes, illumination, and the existence of background clutter, as well as viewpoint changes, as shown in Figure 1.5.

Figure 1.5. The same object captured from different view points.

The focus of this thesis is to develop robust techniques for human action detection and recognition via spatio-temporal features.

To reach this purpose, we begin by understanding human action on its different levels of complexity. We investigate motion information to develop robust human action recognition of tasks of increasing complexity. We start by treating simple actions performed by single actors. The proposed datasets are relatively small, with very small camera motion. Generally, the actors are performing simple action repetitively as shown in Figure 1.6

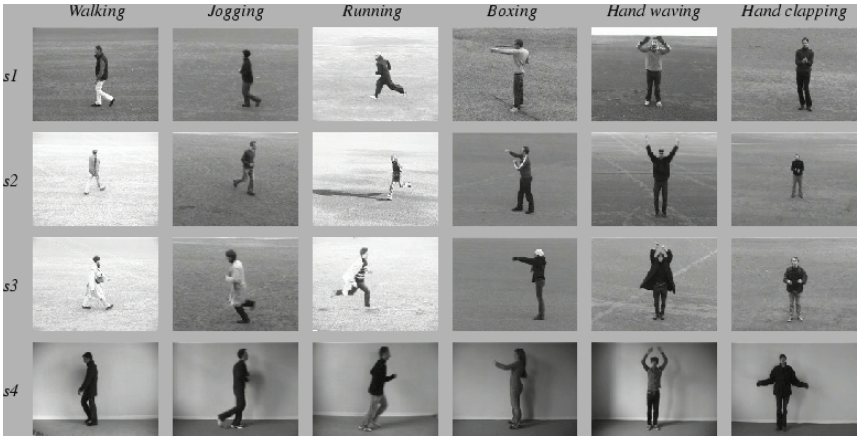


Figure 1.6. Single actions performed by an actor without background details.

In user-generated video footage, the quantity of video data containing human actions and scenes is growing exponentially, with about 48 hours of video uploaded per minute on YouTubeTM [10]. With this growth, the demand for action recognition in Amateur video is certainly colossal. Compared with professionally produced movies, realistic videos are particularly noisy with low quality, poor lighting and. This challenge motivated me to work in more complex conditions in order to make my research more relevant to the real world. At the same time, it must be said that the datasets used from realistic videos are relatively small. The actions to be analyzed are performed by one or more actors, and they involve objects as well as complete human bodies.
The final challenge in this work arises from the need to develop a robust video representation from a large dataset of realistic videos. In this dissertation, we disregard context and focus on the actions in each video analyzed. In fact, neither the background details nor the relations between actors are analyzed. We use temporal action detection in order to segment video actions into relevant action parts. We develop a spatio-temporal features representation that can be applied to realistic tasks.
We believe that human action recognition relies on three main steps:

- *Action detection:* Human action depends on the person performing it. For example,

during the action "drink a coffee" an actor can look at his "phone", or "sit down". For the same action, another person can "drink the coffee" without making any other movement. Someone can "jump" while "walking". Here we evoke the problem of "intra-classes" with which we need to deal when working with realistic videos.

- *Feature extraction:* The extraction of descriptors consists of detecting salient points, patches or volumes that most describe the actions. The dataset videos must be described by the same algorithm. Each video has its own description with a distinctive number of features. The extracted features, from all over a dataset, must have the same dimensionality to perform classification.

- *Classification/training:* Classification is the final step in the human action recognition process. This step involves arranging descriptors into groups, according to their features, so that descriptors with similar features are in the same group. Each group can be used to represent an action.

1.1 Technical issues

In order to track closely human action in videos, many methods heavily rely on detecting the video portions that can describe a specific action [11], citeniebles2010modeling,[12]. For example, in the method of Noguchi et al. [13] video sequences are subdivided into snippets of five frames. In [14], Laptev et al., researchers worked on video patches of thirty to two hundred frames with an average number of seventy frames. They used spatio-temporal descriptors to detect boosted cascade classifiers and an annotated keyframe to describe specific actions, such as drinking. Dementhon et al. [15], extracted features from video portions containing between twelve to eighteen frames, whereas Skindler et al. [12] suggested that action recognition systems need one to ten frames to recognize action. In [16], the authors claim that reconsidering segmentation by generating approximate locations over a few precise objects can increase the accuracy of recognition. The drawback of all these above-mentioned approaches is that dividing a video into equal segments is neither sufficient nor intuitive for detecting an action with all its potential inter- and intra-class variations.
Once videos are segmented, they are described by extracted features. The state-of-the-art attests to various types of low level video descriptors. Several of them yielded accurate descriptions of controlled videos. However, the more realistic the videos are, the more the

description needs to be robust and detailed. To achieve a better description of more complex videos, researchers suggest the fusion of multiple features as well as the optimization of the video description with respect to time consumption. In particular, spatio-temporal local features have been widely studied to detect human actions, objects and events in videos. Although video analysis with attention to spatio-temporal features is not a new method, it has not yet been much explored.

After the video description step comes the training or learning task. In the context of action recognition in videos, the representation of video objects as a bag-of-visual-words (BoVW) through a histogram has become a very active research field [17]. This histogram can be used in a classifier framework to show the difference between object's classes. However, the main weakness of a given BoVW is that not all words will be informative, accurate or objective in terms of describing actions. Consequently, the selection of the most informative words is required. The most used methods to select visual words are Machine learning techniques, such as Boosting [18], an adaptation process such as Multiple Instance Learning (MIL) [19] or many other State-of-the-art algorithms [20], [21]. While these approaches provide significant results for action recognition, they need to be adapted to be applied into the temporal domain.
Recent studies in both the spatial [22] and temporal [23] domains explore the descriptive and discriminative performances of these features. Although BoVW has shown promising recognition results, classification relying on the BoVW suffers from drawbacks caused by the quantization of the descriptors into codewords. Another issue of BoVW representations is the neglect the spatial information. In fact, the objects are clustered without consideration of their position in the image or video frames. The utility of the BoVW approach decreases when working with complex actions, since such actions increase the semantic gap between the descriptor and the action.
Thus, this representation is unable to bridge the semantic gap between computable low-level features (e.g. visual, audio, and textual features) and semantic information that they encode (e.g. the presence of meaningful classes such as "a person clapping", "sound of a crowd cheering", etc.) [149]. Despite much progress made in the past decade , the BoVW computational approaches involved in complex event recognition remain reliable only under certain domain-specific constraints.

1.2 Main contributions

In this thesis, we address the problem of human action recognition from unconstrained and realistic videos. To this end, feature extraction and representation algorithms are proposed. The proposed descriptors rely on spatial interest points and temporal action evolution, such that high accuracy can be achieved at low computational cost. A temporal video segmentation and action detection algorithm are also proposed to optimize the action recognition efficiency. We briefly outline our research contributions as follows:
From the action detection perspective, our contribution lies in the design of two computationally efficient action detectors.

- *Trajectory based action detection:* Trajectory based action detection: Within the purview of the work proposed in this thesis, we address spatio-temporal action detection. We suggest detecting human action and action boundaries in order to localize the processed video patch (i.e., the video parts which will be processed). In a given video, the action happens in a specific interval of time. Hence, treating the entire video will lead not only to treating the action to be detected and then described but also unnecessary video portions that contain non-significant actions or no action at all. From this comes the idea of designing an algorithm that focuses on highlighting body movements. The proposed technique relies on Interest Points (IP) detection. We choose the Speed Up Robust Feature (SURF), as it is a highly performing and fast interest points detector [24]. The detected interest point trajectory is tracked, and the moving IP are selected. The latter are divided into groups, and every group of IP is tracked. We test the efficiency of this method on increasingly complex datasets, moving from simple actions to more complex, realistic ones. Experiments prove that this new approach yields many advantages. On the one hand it permits us to reduce the dimensionality of video descriptors. On the other hand, it allows us to extract descriptors from predefined patches instead of the entire video. As a result, the processing time is considerably reduced. The detection of human action is a challenging task, because actions can be both simple and incredibly complex. The proposed idea allows the division of human movements into a succession of small motions. This provides a better detection of actions as well as a better solution to the problem of intra-class variations. The developed algorithm detects small actions. From these ordered mini-actions, it constructs an entire action.

- *Dense trajectory based action detection:* Dense trajectory based action detection: For this part of the research, we focused on optimizing human action detection. We propose to detect moving humans or objects in the scene. This is achieved by first detecting IP densely distributed throughout the scene. Then, based on the optical flow computation, we design bounding boxes surrounding the human(s) and/or objects in movement. We allow objects detection, because we believe that objects related to the context of the action can play an important role in the classification step. Then, in every detected bounding box, we extract IP that perform significant motions. By setting a minimum motion value, we take into account and reduce the small motions generally produced by the camera. Thus, we divide the human action into small portions called "actionlets". This allows the optimization of descriptor size and reduction of processing time.

  From the video description perspective, our contribution lies in the design of two relevant and efficient descriptors.

- *Spatio-temporal SURF (ST-SURF):* After video temporal segmentation and human action detection, we propose a new algorithm based on the extension of the SURF descriptors to the temporal domain. SURF is well known as a spatial visual descriptor that proves its efficiency for image retrieval, since it is not adversely affected by rotation and illumination. The SURF approach uses a Hessian detector, which is efficient and especially fast. The detected SURF descriptors are associated with their optical flow. A parametrization of the optical flow and the SURF information leads to a spatio-temporal SURF descriptor. This feature captures the spatial information provided by the SURF and the motion information gleaned from the optical flow. In addition, ST-SURF contains localization information, which is missed by the BoVW approach. Each descriptor is tested on several datasets at varying levels of complexity. We also perform an extensive evaluation of the proposed descriptors action detection accuracy both while using it by itself as well as when it is associated with other descriptors. Our approach demonstrates improvements over the traditional SURF and several other spatio-temporal descriptors of the state-of-the-art.

- *Histogram of trajectory motion orientation (HTMO):* We investigate a complementary source of information based on the tracking of the IP trajectory. We present

an efficient descriptor used to recognize human action in realistic video benchmarks. Our aim is to design a feature that describes the distribution of the motion orientation around an IP. We introduce a novel spatio-temporal descriptor based on the histogram of SURF-based patches trajectory orientation, for which we coin the term HMTO. HTMO captures joint cues between the distribution of motion and appearance of constituent IP. The Motion trajectory is extracted by optical flow computation. A parameterization step is exploited to extract HMTO in both x and y directions ($HMTO_x$) and ($HMTO_y$).

The descriptors we designed are tested in different scenarios. In order to evaluate our proposed approaches, we start by using simple yet challenging datasets with only one actor performing one action without any influence from camera motion or a challenging background. In order to prove the robustness of the proposed descriptor, we use more complex datasets with realistic videos captured by amateurs from real scenes. These videos contain changing backgrounds, camera motion, and illumination variation, among other challenges.
The introduced descriptors are employed alone as well as in different configurations based on their fusion with other descriptors from the state-of-the-art. We rely on early and late fusion processes to explore human action recognition in highly complex small and large datasets. At the end of the thesis, we propose a global high level video description based on features that are able to capture both spatial and temporal information from videos with robust camera motion and complex backgrounds.

1.3 Thesis outline

The remainder of the report is organized as follows:

- **Chapter 2** discusses relevant work on action recognition and introduces our proposal for an improved approach to action recognition.

- **Chapter 3** A new spatio-temporal descriptor we called ST-SURF is proposed in this chapter. ST-SURF is based on a novel combination of the speeded-up robust features (SURF) and optical flow. The Hessian detector is employed to find all interest points. To reduce the computation time, we propose a new methodology for video segmentation into Frame Packets (FPs), based on interest points trajectory tracking. We consider only moving interest points descriptors to generate a robust

and powerful discriminative code-book based on K-mean clustering. A subjective study has been launched to evaluate our index in comparison with current indices.

- **Chapter 4** An efficient action recognition approach is developed in this chapter. It is based on the spatial position and the trajectory information of the SURF. A novel selective video segmentation scheme based on dense sampled interest point (IP) trajectory tracking and matching is proposed. Segmented video snippets (SS) describe a part of an action called an "actionlet". A new descriptor, called histogram of motion trajectory orientation (HMTO) and based on the SURF region, position and scale, is introduced. The performances of the proposed method are evaluated on some known datasets. Extensive experimental tests conducted to ascertain the superiority of the proposed method confirm the efficiency of our method.

- **Chapter 5** In this chapter, temporal action detections are investigated based on a sampling of densely distributed interest points and by reducing camera motion effect. Once the moving actors are detected, the video temporal segmentation is performed. The video description is evaluated on big, realistic datasets.

- **Chapter 6** summarizes the results of the current research and proposes several lines of future research.

The bibliography is given at the end.

1.4 Publications

Based on the findings of the current research, some papers have been published to the international and national conferences as following:

Journal papers:

1. **S. Megrhi**, M. Jmal, A. Beghdadi, W. Souidène, "Spatio-temporal Action Localization and Detection for Human Action Recognition in Big Dataset", in *Journal of Computer Vision and Image Understanding (CVIU): Special Issue on Individual and Group Activities in Video Event Analysis*, Submitted in November, 2014.

2. **S. Megrhi**, A. Beghdadi, W. Souidène, "Selective Video Segmentation and Local Trajectory Tracking for Human Action Recognition", in *Image and Vision Computing Journal*, Submitted in September, 2014.

National and International conference papers:

1. **S. Megrhi**, A. Beghdadi, W. Souidène, "Spatio-temporal action detection for human action recognition in video", accepted in *The twenty seventh IST/SPIE Electronic Imaging, 2015*.

2. **S. Megrhi**, A. Beghdadi, W. Souidène, "Trajectory Feature Fusion for Human Action Recognition", *The 5th European Workshop on Visual Information Processing (EUVIP), 2014*.

3. **S. Megrhi**, A. Beghdadi, W. Souidène, "Classification des actions humaines basée sur les descripteurs spatio-temporels", *14èmes Journées Francophones Extraction et Gestion des Connaissances, EGC, 2014*, Acceptance rate 24%.

4. **S. Megrhi**, W. Souidène, A. Beghdadi, "Spatio-temporal SURF for Human Action Recognition", *the fourteenth LNCS/Springer Pacific-Rim Conference on Multimedia (PCM), 2013*, acceptance rate 35%.

5. **S. Megrhi**, W. Souidène, A. Beghdadi, "Spatio-temporal salient Feature extraction for Perceptual Content Based Video Retrieval", *the eighth IEEE Color and Visual Computing Symposium, 2013*.

6. **S. Megrhi**, W. Souidène, A. Beghdadi, "Video indexing using salient region based spatio-temporal segmentation approach", *the fifth IEEE International Conference on Multimedia computing and systems, pp. 170-173, 2012*.

7. W. Souidène, **S. Megrhi**, A. Beghdadi, "Perceptual non local mean (P-NLM) denoising", *the fifth IEEE International Conference on Communications Control and Signal Processing (ISCCSP), 2012*.

CHAPTER 2

LITERATURE REVIEW

Success is not final,
failure is not fatal:
it is the courage to continue that counts.
Winston Churchill

Contents

A detailed discussion of related work in the area of action recognition is presented in this chapter.

2.1 Introduction

Action recognition is a field that focuses on extracting useful information from a scene in an image or a video. In recent years, much research has been focused on extracting robust and relevant features. The methods used extract a representation of the scene content as descriptors. They then use machine learning algorithms to treat these descriptors in order to achieve a specific purpose, for instance, the recognition of actions or objects within a video.
The recognition of human actions in videos is receiving increasing attention due to its wide range of applications, such as video indexing and retrieval [1], human-computer interaction, digital entertainment, surveillance videos [2], etc. In the realm of user-generated video footage, the quantity of video data containing human actions and scenes is growing exponentially, with about 48 hours of video uploaded per minute on YouTubeTM. As the quantity of video data grows, so does the demand for action and scene recognition or content-based video data retrieval. Most often, significant events in these videos are characterized by actions, such as boxing, kissing, or the stealthier actions or behaviors found in a surveillance video. However, action recognition is usually confronted with many issues including the necessity of handling considerable occlusions, scale changes, illumination, and the existence of background clutter, as well as viewpoint changes. Several surveys have been conducted in order to analyze human actions in videos. Among the most important of these studies are those conducted by Aggarwal and Cai in 1997 [3], Gavrila in 1999 [4], Wang and Singh with their work in 2003 [5], Buxton in 2003 [6], Aggarwal and Park in 2004 [7] , Turaga et al. in 2008 [8], and Aggarwal and Ryoo in 2011 [9].

2.2 Approaches for human action recognition

This section describes the action recognition methods proposed in the literature. The topics covered are arranged according to the main components of the action recognition framework 2.1.

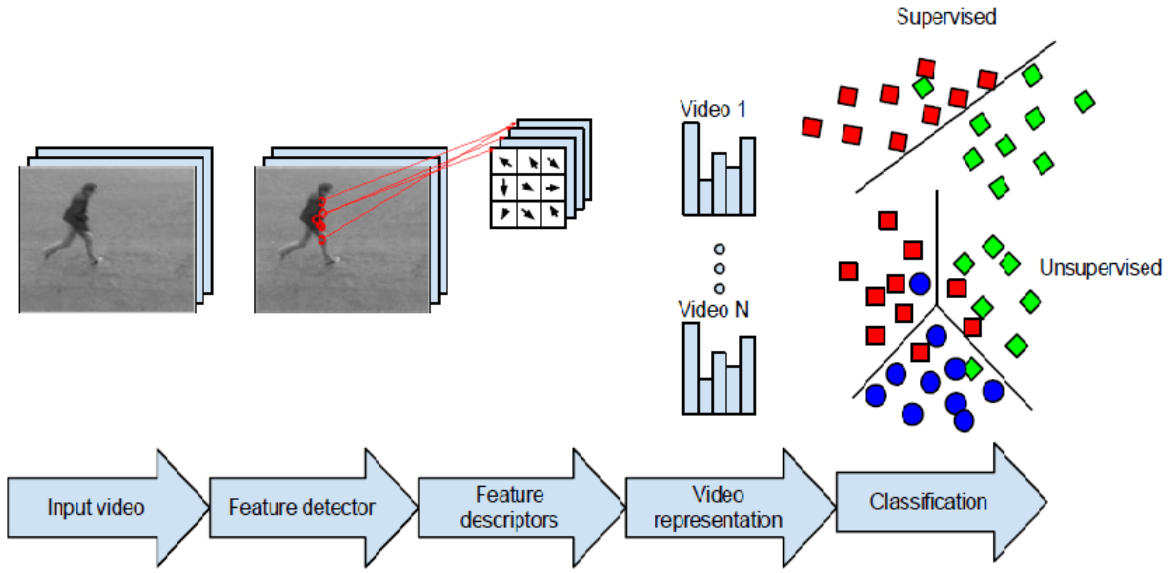


Figure 2.1. Action Recognition Framework.

2.3 Features extraction methods

Features extraction is the first step toward a video description. Multiple proposals have been developed with the common objective of producing an optimal action description. We review the particularities of these approaches and present relevant works.

2.3.1 Global features methods

In this approach, moving objects are described as a whole part of the video scene. Human parts detection and tracking are the first step and the key to the success of this approach. In fact, after this step, the segmented regions are described based on extracted information. In the same spirit, authors in [25] consider human actions as 3D shapes, having analyzed 2D shapes and generalized them into volumetric space-time action shapes. In order to achieve this transformation from 2D to 3D, they first extract the spatial information of the orientation and location of a given figure's torso and limbs. Next, they extract the silhouette's global motion. They segment space-time portions by using sliding temporal windows. The segmented blocks are then described by a high-dimensional feature vector. [25] employs a nearest-neighbor classifier to vote for the relative class. Examples of space-time shapes from the Weizmann dataset are illustrated in Fig. 2.2.

Figure 2.2. Space Time Shapes of the three actions: "jumping-jack", "walking" and "running" (coutsey from [25]).

2.3.2 Silhouette based features extraction methods

Another approach to action recognition, this time based on the human silhouette, is presented by Bobick and Davis [26]. They propose Motion Energy Images (MEI) and Motion History Images (MHI) to extract temporal information from videos. The MEI localize the motion in videos, while the MHI provide information about the most recent actions in the video, that is, temporal information, Fig. 2.3.

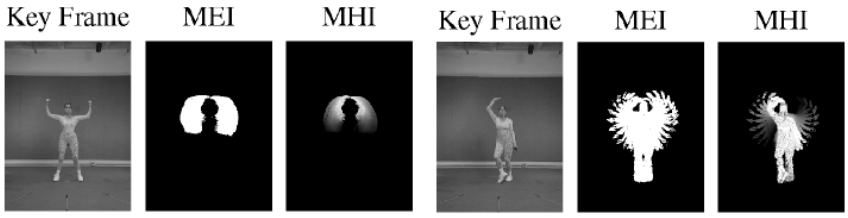


Figure 2.3. Motion Energy Image (MEI) and Motion History Image (MHI) (coutsey from [26]).

In like manner, Laptev introduces a human key-pose detector in key frames as a global approach for action localization in realistic videos [14]. Firstly, a filtering step is performed to reduce computational complexity. Secondly, cuboids regions features, based on histograms of oriented spatial gradients and histograms of optical flow, are extracted. Fig. 2.4 demonstrates the appearance and motion features used to represent a sample of the action of drinking.

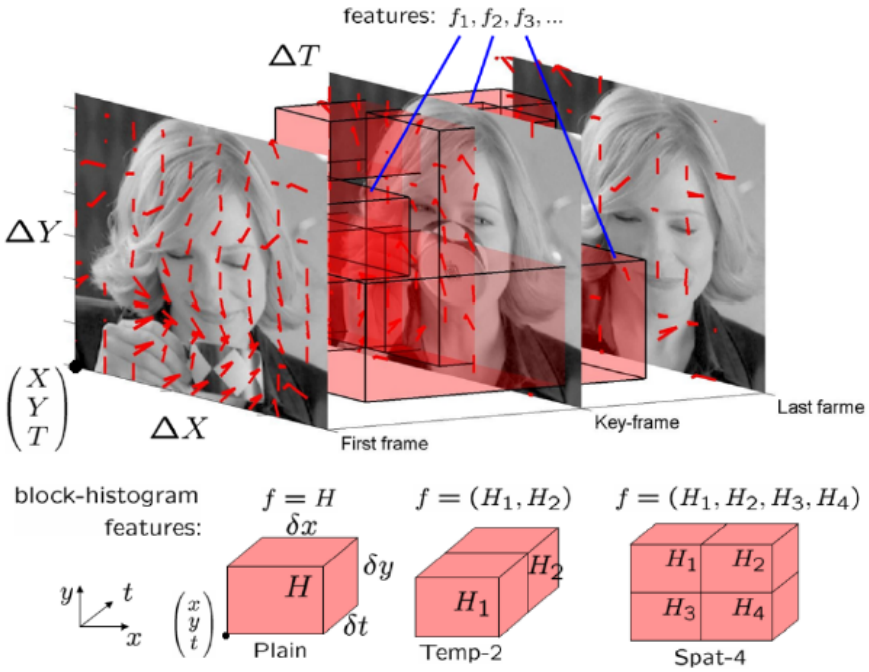


Figure 2.4. (top) Appearance and motion features; (bottom) spatial and temporal layouts for action representation (courtesy of [14]).

2.3.3 Skeletal-based features extraction methods

Skeletal approaches to action recognition are based on posture estimation. These methods assume that recognition of a persons posture is sufficient to determine the action in which he is engaged. In order to estimate accurately the posture of a human skeleton, one needs to rebuild some key points, generally the head, shoulders, feet, and hands. In [27], Gavrila et Davis use a stereo camera associated with markers to identify the head and the limb joints. Joint angles were used as descriptors and classification was done through Dynamic Time Warping. In 1998, Fujiyoshi et Lipton [28] proposed a motion analysis of humans in video stream. The boundaries extracted from moving humans serve to extract a "star skeleton". The extracted skeleton consists of a maximum of five parts representing the head along with the ends of the legs and arms, Fig. 2.5.

A more robust skeletal representation is proposed by Andriluka [29]. This method constructs a representation based on ten limbs, Fig. 2.6. The appearance of body parts is modeled using densely sampled shape context descriptors and discriminatively trained AdaBoost classifiers.

For skeletal-based features extraction methods to be successful in 3D space, several

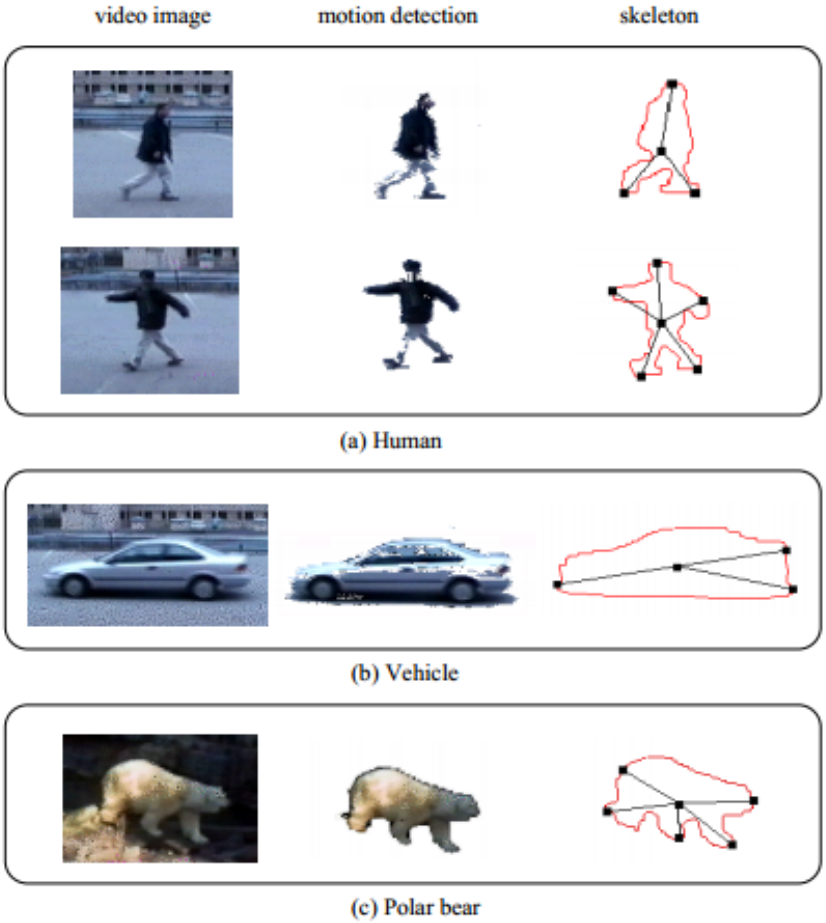


Figure 2.5. Skeleton consisting on a maximum of five parts representing the ends of the legs and arms, and the head (courtesy of [28]).

Figure 2.6. Upper and full human body poses estimation (courtesy of [29]).

approaches have been proposed. In fact, while Parameswaran et Chellappa [30] capture 3D motion to recognize single actions, Lv et Nevatia [31] present an approach that does not explicitly require a 3D pose be detected in each frame. Instead, each action is modeled as a series of synthetic 2D human poses rendered from a wide range of viewpoints. Lv and

Nevatia then construct a graph model called *Action Net* based on *key poses*, see Fig. 2.7.

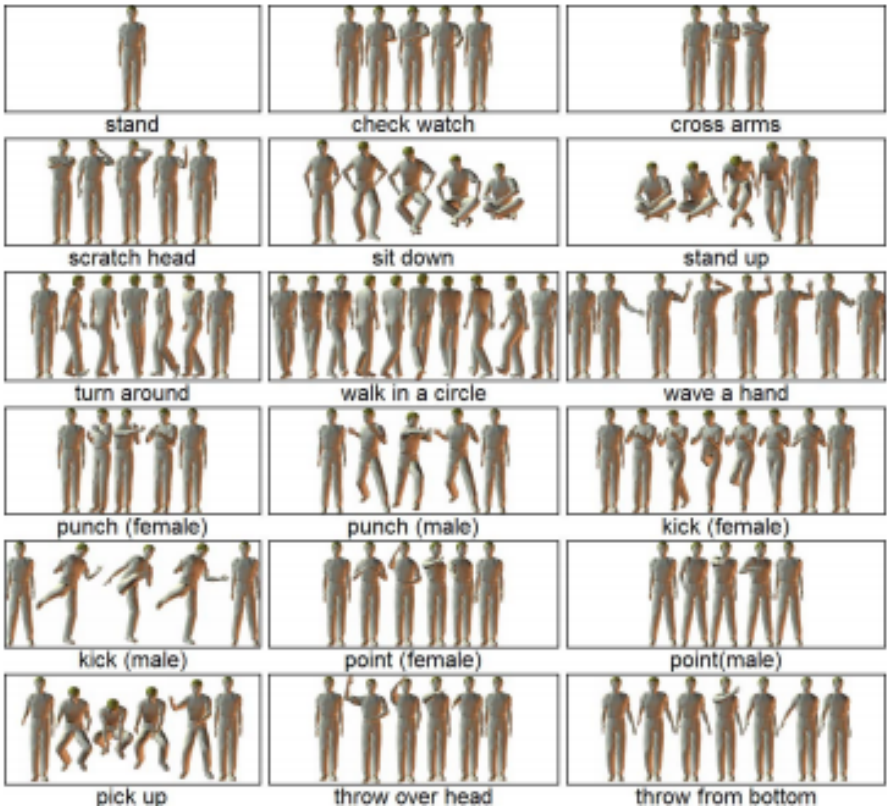


Figure 2.7. Upper and full human body poses estimation (courtesy of [31]).

2.3.4 Local features methods

Local features extraction approaches consist of describing the actions in a video as a collection of local descriptors. These features can be extracted separately or densely [32], as well as by deliberately considering or neglecting spatial information [32]. For a given region of interest, also called an interest point (IP), the extracted feature describes this interest point and its neighboring visual and motion information. For human action recognition, much progress has been made in the area of local spatio-temporal (LST) features extraction [33]. Almost all the LST descriptors now provided have been derived from the extension of 2D spatial features or detectors to the temporal domain. The method in [34] is based on the space-time derivatives of local patches. Niebles et al. [35] summarize the video by space-time interest points. They use a probabilistic Latent Semantic Analysis (pLSA) model and a Latent Dirichlet Allocation (LDA) to detect the action class in a given video.

The Cuboids descriptor was proposed in [36], while 3D-SIFT was introduced in [37] to recognize actions in video volumes. In the same vein, [38] proposed the C2-shape features. The Histogram of oriented gradient and Histogram of optical flow (HoG-HoF) based method has been proposed in [39]. Authors in [40] introduced the spatio-temporal Hessian detector and the extended SURF. Other interesting works, such as HOG3D [41], the local Trinary Patterns [42] and Space Time SURF in [13] have been introduced. ST-SURF is a spatio-temporal SURF obtained by the tracking of the detected SURF points. This descriptor stands out from the rest because not only is it compact and reliable but it also focuses only on moving objects in the scene, while ignoring small motions [43].
One of the most famous categories of local features is that of spatio-temporal interest points (STIPs) [44]. In the work of Laptev [44], the STIPs are considered as a Harris corner detector extension to 3D. The STIPs are detected by computing local maxima of the extended corner function, Fig. 2.8.

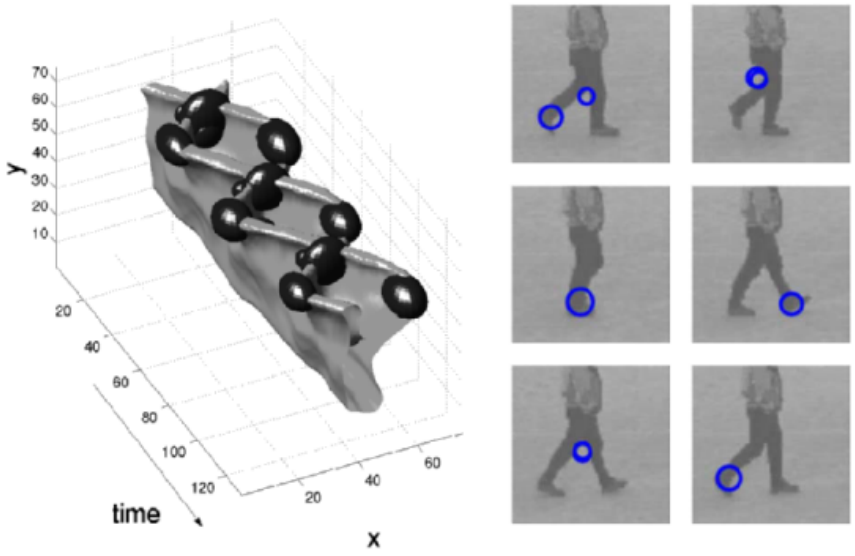


Figure 2.8. Spatio-temporal interest points based on the motion of the legs of a walking person (courtesy of [44]).

Dollár et al. [36] uses a Gabor filter temporally and a Gaussian filter spatially. Each interest point is detected by local maxima over the response function of Gaussian and Gabor filters. Willems et al. [40] propose the detection of interest points through the determinant of the Hessian matrix. Interest points are detected at different spatial and temporal scales. To speed up computations of scale-spaces, box-filters are used in combination with an integral video structure. Examples of interest points detected in human actions are

presented in Fig. 2.9.

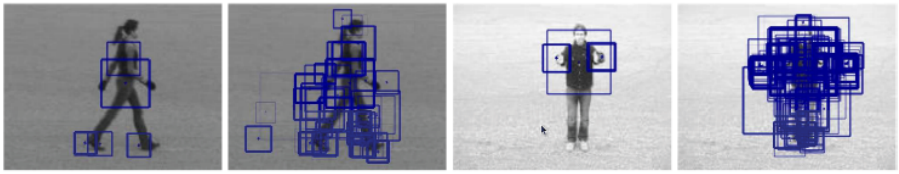


Figure 2.9. Spatio-temporal interest points when using the determinant of the Hessian. (courtesy of [40]).

Descriptors can also be represented by a grid like HOG [45]. In the work of [40], the E-SURF is an extension of the SURF to 3D. In the same way, Klaser introduces HOG3D as an extension from HOG [41].

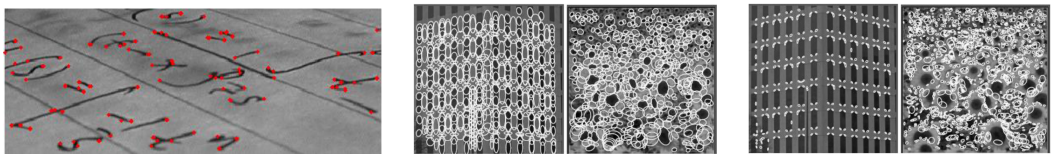
2.4 Features detection

Local features are detected based on their saliency, meaning the quantity of changes in the neighborhood of an interest point. In the image processing field, the interest points (IP) are detected based on an important contrast changes. In videos, moving objects induce motion changes that are considered salient in the temporal domain. The detected interest points are space-time interest points (STIP).

2.4.1 Interest point detectors

Interest point detectors focus on isolating those areas of an image that present significant visual features, such as edges, corners and "blobs". The basic objective of these detectors is to find the same IP on an object or a scene when the viewing conditions change. For example, the IP must be invariant to changes in scale, rotation, perspective, etc. It will suffice here to give an idea of the diversity of the methods developed. One of the most used detectors is the Harris corner detector. This detector is based on the eigenvalues of the covariance matrix of the analyzed region. However, the Harris corner detector is not invariant to changes in scale. To overcome this disadvantage, Lindeberg introduced, in [46], the concept of automatic selection of scale, which can detect points of interest in an image, each with their relative scale. Lindebergs detector is based on the detection of the maxima of the Hessian matrix determinant. In order to detect blobs, Lowe detected the maxima of the Laplacian, which corresponds to the trace of the Hessian matrix [47]. Mikolajczyk

and Schmid improve this method in [22] and create a robust detector, called Hessian-Laplace, using the determinant of the Hessian matrix to detect the region of interest and the Laplacian to select the scale. The Harris-Laplace detector relies on the Harris corner detector to select IP and the Laplacian for the choice of scale. We can see in Fig. 2.10, that the Harris-Laplace detector detects the corners, while the Laplacian detector detects structures such as blobs.



(a) (b) (c)

Figure 2.10. Different IP detectors. (a) Harris detector; (b) Laplace detector; (c) Harris-Laplace detector

2.4.2 Scale Space

To detect the same objects in different recording conditions, such as differing camera viewpoint, illumination and resolution, STIPs must be detected in different temporal and spatial scales. A new scale is obtained by a convolution with a Gaussian blur function. A video $f(x, y, t)$ is, then, represented by the following scale-space:

$$L(x, y, t; \sigma^2, \tau^2) = f * G(.; \sigma_l^2, \tau_l^2) \tag{2.1}$$

Where G is the spatio-temporal Gaussian kernel, x and y are the frame parameters, t represents the frame at time t, σ^2 is the standard deviation of the filter.

2.4.3 Background subtraction

Given a frame sequence from a fixed camera, background subtraction consists of detecting all the foreground objects in every frame. Motions of the objects are then detected. Any significant change in an object's location signifies that it is a moving object. Background subtraction has attracted the research community since 1978 [48, 49]. Several researchers have employed a Gaussian Mixture Model [50]. They extract the stationary background by computing the mean of the highest weighted Gaussian at each pixel position. They next detect the colors that appear less frequently by computing the mean of the Gaussian with the second highest weight. Finally, they compute the background subtraction result.

Stauffer and Grimson [51] focus on a mixture of Gaussian to model the pixel color, while Rittscher et al. [52] use the Hidden Markov Model (HMM) to classify image blocks as belonging to the background state, the foreground or the shadow state. All these methods work well when the background is static. For non-static backgrounds, such as raining scenes or escalator scenes, authors in [53, 54] discern and predict the motion information in a scene by computing a weighted sum of its previous values and white noise error. These approaches use the auto-regressive moving average called the ARMA processes.

2.4.4 Action tracking and segmentation

To track human action in videos more closely, many methods focus on detecting the video portions that can describe a specific action [11], [55],[12]. For example, in the method of Noguchi et al. [13] video sequences are subdivided into snippets of five frames. In [14], Laptev et al. worked on video patches of thirty to two hundred frames with an average number of seventy frames. They used spatio-temporal descriptors to depict cascades of boosted classifiers as well as an annotated key frame to describe specific actions such as drinking.

Dementhon et al. [15] extracted features from video portions containing between twelve to eighteen frames, whereas Skindler et al. [12] suggested that action recognition systems only require one to ten frames to recognize action. In [16], the authors claim that reconsidering segmentation by generating approximate locations for a few precise objects can boost recognition. The drawback of all of these above-mentioned approaches is that dividing a video into equal segments is neither a sufficient nor an intuitive way to detect an actions inter and intra-class variations.

More recently, researchers have been focusing on video description by tracking the motion of an interest point [56]. This allows the exploration of several motion cues such as velocity [57, 58], orientation [59, 60], location [61], trajectory curves [62], trajectory parts [63] and different motion cues combinations [43]. Trajectories can be extracted by matching interest points. Sun et al. [47] encodes the SIFT trajectory to extract spatio-temporal context models. Trajectories of interest points in successive frames are then extracted [64].

Trajectory patterns can be extracted by using a tracker including, but not limited to, the KLT (Kanade-Lucas-Tomasi) tracker [65], which is used to extract trajectories in videos [66, 56, 57]. Authors, in [67], used SIFT and KLT features to extract long duration trajectories in order to capture more information about actions. Wang et al. proposed dense

trajectory tracking to encode temporal information [68]. They suggested the use of dense optical flow to track densely detected interest points [33]. They proved that trajectory tracking is an intuitive and successful approach in several public benchmarks.
It is worth pointing out that trajectory smoothness and segmentation are important issues for trajectory description. To segment a trajectory, several studies used trajectory clustering [69, 70]. Other methods are based on moving object trajectory tracking. In order to detect "phoning"' and "standing-up" actions, authors in [71] used a sliding window classifier to extract temporal information and a human tracking process to extract trajectory information.
In [72], Mean Shift clustering is applied to trajectories to extract cluster centers describing rare events. A trajectory based on a Hidden Markov Model (HMM) for extracting the temporal causality is proposed in [73]. Sapienza et al. [74], proposed the extraction of space-time cues from an action. Space-time action is detected by using 3D bounding boxes or by detecting scores aggregation.
Recently, in [33], a new scheme is proposed for characterizing dense trajectories in order to preserve trajectory smoothness. The trajectories' attributes are, then, extracted by concatenating the interest points' trajectory in successive frames with a limited length of fifteen frames. Finally, a trajectory shape descriptor characterizing the displacement is computed.
Another issue in the pursuit of action tracking and segmentation is the challenge of ensuring the robustness of extracted features in spite of camera motion and varying backgrounds. The insight behind the success of several proposed video descriptors is the use of a static camera and uniform background [75, 76]. Although many methods have been proposed for reducing camera motion [77, 78, 66, 79, 80], this problem is still unresolved in some cases. It is the goal of this work to develop a video presentation which removes camera motion without sacrificing significant human action cues. To this end, the motion boundaries Histogram descriptor (MBH), derived from the optical flow gradient, is used as in [81]. The MBH removes constant motion and preserves only significant movements.
MBH has been employed in various action recognition schemes [33, 68]. It provides more interesting results when applied to videos containing important camera motion. Though MBH is not dedicated to remove camera motion, when combined with the spatio-temporal SURF (ST-SURF) proposed by [43], it will contribute significantly to camera motion compensation.
The description method using Spatio-Temporal Speeded-up Robust Features (ST-SURF)

works by detecting and tracking SURF points. This descriptor is advantageous because it is compact and reliable, and it focuses only on moving objects in the scene by ignoring small motions [43].

2.5 Machine learning and classification approaches

In terms of classification approaches, algorithms from the machine learning community have been heavily borrowed. The following sections describe the main principles of machine learning.

2.5.1 Supervised and unsupervised learning

Supervised and unsupervised learning are the main methods used in Machine Learning. In the supervised approach, machine learning is achieved by annotating the data. The supervised learning consists of a training step which leads to an output of information, generally a class assignment. After the training data is analyzed, the testing step takes place in order to predict the most likely output class for any previously unseen input instance. Unsupervised learning concerns the identification of structures from unmarked data. This means that the information fed to the system is labeled, and there is no evaluation metric which can be used to distinguish between different instances.

2.5.2 Classification approaches in human action recognition

While the authors in [26, 25, 36, 82, 83, 84, 47] chose to use NN classifiers to avoid explicit training, other researchers have focused on deep neural networks [85]. In [86], the authors performed their classification tasks using the Hidden Markov Model (HMM). Classification methods based on sparse linear representations are also used in object recognition and tracking [87, 88]. One of the most used algorithm is support vector machine (SVM). The mais goal of an SVM is to maximize the separation margin between different classes. SVM is used in various domain such as action recognition or face identification [89].
In order to map the training features onto a high-dimensional feature space, one can use a kernel function, which will lead to an optimal separation hyperplane in the feature space. The authors in [90], proposed the performance action recognition using a multi-class SVM framework. Another classification technique used for action recognition is boosting [91, 92]. All of the above proposed classification approaches improve the performance of any classifier

by combining a number of weak classifiers into a strong main one. In this thesis, we follow many methods from the literature [93, 37] by using SVM to recognize actions.

2.6 Bag of Visual Words Approach

In the context of action recognition in videos, the representation of video objects as a bag of visual words through a histogram has become a very active research field. The basic idea is to divide a set of descriptors into groups, so that objects of a similar type will be gathered into one cluster. This categorization process leads to the construction of a visual code-book. Each code, or visual word, can be used to represent an action. Because of its high performance and simplicity, the K-means algorithm has recently become widely used in the construction of visual code-books, Fig. 2.11.

In this step, a visual vocabulary is built based on the extracted features. The latter are converted into "words". The set of these words forms the visual vocabulary. A simple approach to producing a visual vocabulary is by performing clustering with a K-means algorithm applied to the set of descriptors vectors. The words in the vocabulary are then defined as the centers of the clusters. The number of clusters determines the number of words in the visual vocabulary. Thus, a word is assigned to each region, and an image can be represented by a histogram of visual words in the video. Fig. 2.11, illustrates the generation of visual words from features vectors.

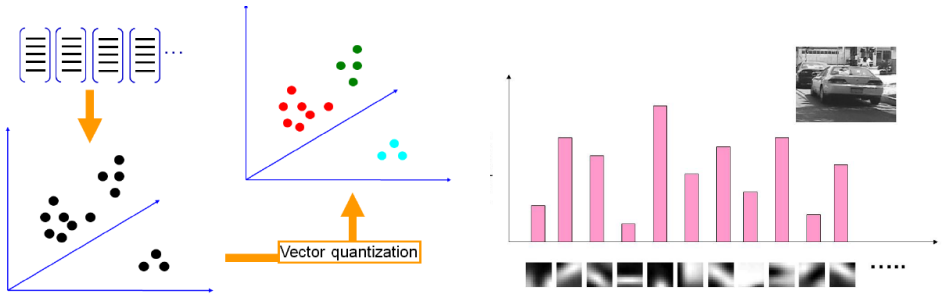


Figure 2.11. Code-book generation. (a) K-means clustering step for quantization ; (b) Example of visual words distribution

The descriptor extraction step is followed by a classification task based on the code-book generation. Several approaches have been proposed to extract a codebook for action recognition. A codebook can be generated using a wide variety of techniques, including, but not limited to, Random forest [94, 95], Sparse codebook learning [11, 96] or bag of

visual words (BOVW) [97, 39, 33].

In a BOVW scheme, descriptors are first extracted. Then, a quantization step is performed to build a visual word codebook. Finally, every video is described by the distribution of the visual words. The BOVW approach has achieved good results in action recognition for both image [98] and video analysis [99]. This is due to the fact that BOVW is an orderless feature presentation that discards features spatial position and the inter-relationships among the extracted visual words. However, the accuracy of BOVW decreases as the database grows in size and is more realistic with many actors and richer backgrounds. In [74], the authors suggested that this is due to the descriptor extraction from the whole video or the setting of the video patches sizes to extract features.

To incorporate spatial information, the spatial-temporal pyramid is a relevant choice [39, 100, 33]. This approach has been introduced for analyzing and recognizing natural scenes categories [101]. The basic idea is to divide the image into increasingly sized sub-regions then extract histograms of local features detected inside each sub-region. In our work, spatial information is injected into the video description by a pattern called Motion Distance (MD). Consequently, there is no need for extra computations to add spatial information into a BOVW approach.

2.7 Datasets

In this thesis, we conduct the proposed action recognition framework on human action classification datasets. We studied different datasets, varying from controlled datasets to realistic ones. The proposed action recognition methods are tested on the KTH dataset [34], UCF sports Dataset [102], UCF 101 [103] and UCF 11 [100]. Fig. 2.12, depicts the major actions of these datasets.

In the following section, we provide a brief overview of each dataset.

2.7.1 KTH dataset

The KTH dataset is commonly used as a public benchmark test of spatio-temporal features [13]. This dataset contains six kinds of actions, such as walking, running, jogging, boxing, hand waving and hand clapping. We consider six action classes performed by twenty-five persons in four different scenarios (indoor, outdoor, different clothes outdoors, scale out-

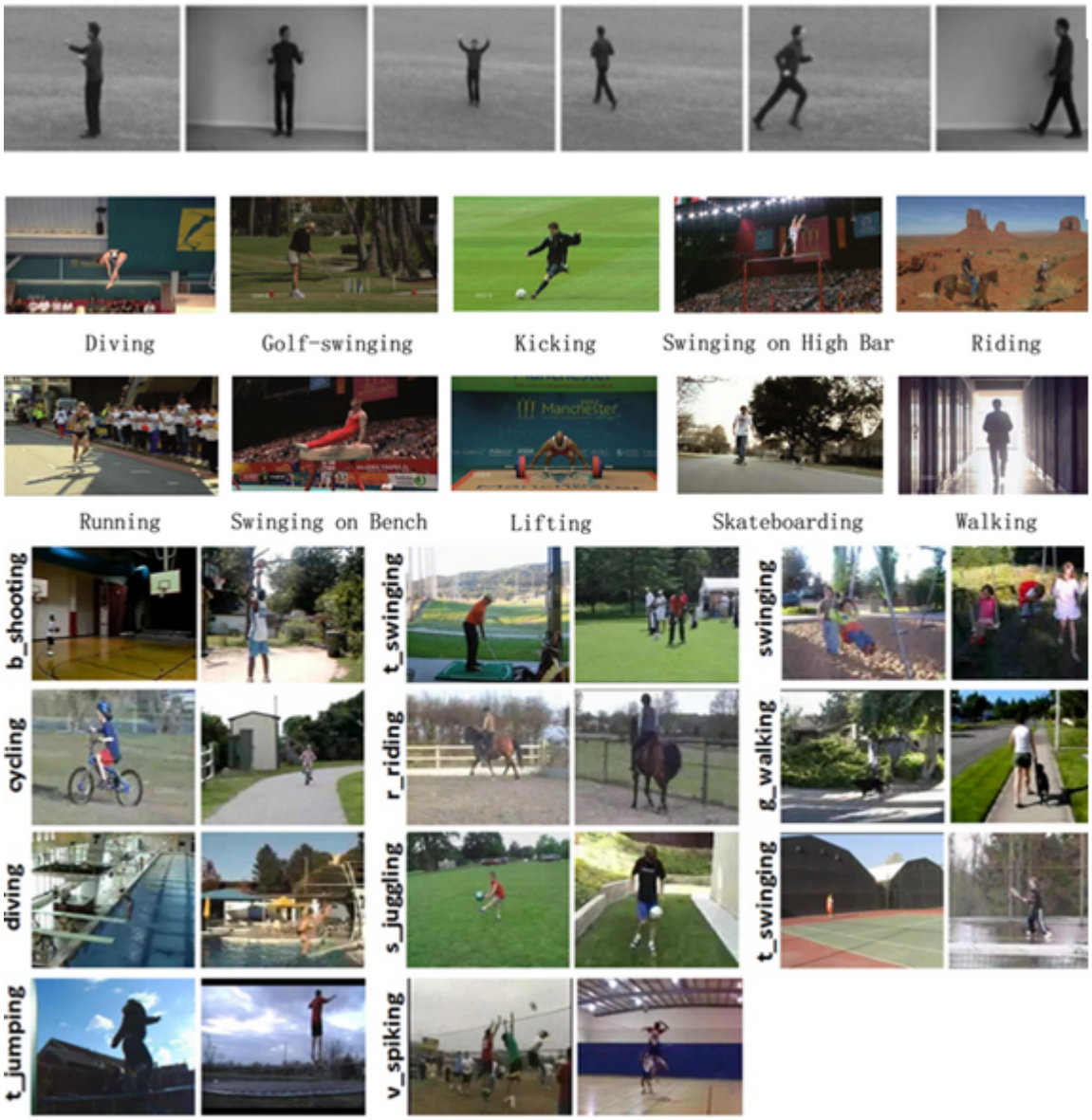


Figure 2.12. Selected frames from the evaluated benchmarks. KTH: 6 actions, UCF sport: 9 actions and YouTube: 11 actions.

doors) with a total of 2391 video samples, all with a homogeneous and static background. The average length of videos in the KTH dataset is about 20 seconds, Fig. 2.13.

2.7.2 UCF sport dataset

The UCF sports dataset is a realistic and challenging dataset obtained from broadcast sport videos by Ahmed et al. [102]. The collection represents a natural pool of actions featured in

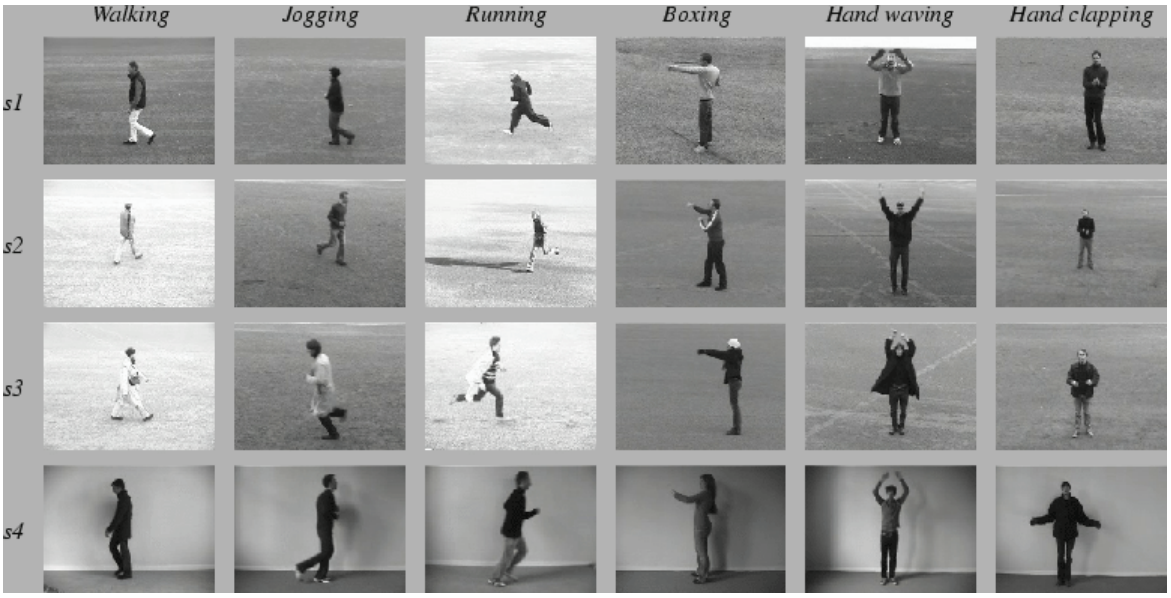


Figure 2.13. KTH dataset actions.

a wide range of scenes and viewpoints. The publicly available part of this dataset contains nine actions, namely Diving (16 videos), Golf swinging (25 videos), Kicking (25 videos), Lifting (15 videos), Horseback riding (14 videos), Running (15 videos), Skating (15 videos), Swinging (35 videos), and Walking (22 videos). This dataset contains around 200 video sequences at a resolution of 720x480 [102], Fig. 2.15.

2.7.3 UCF11 YouTube Action Data Set

UCF11 is the newest update of the YouTube action dataset. It contains 1168 videos and the following eleven action classes: basketball shooting, biking/cycling, diving, golf-swinging, horseback riding, soccer ball juggling, swinging, tennis racket swinging, trampoline jumping, volleyball spiking, and walking a dog. For each class, the videos are grouped into twenty-five sub-groups with about four to nine action clips in it. The video clips in the same sub-group share a few common cues, such as the same actor, similar appearance, same background, similar viewpoint, etc. This data set is very challenging due to large intra-class variations, and differences in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, illumination conditions, etc.

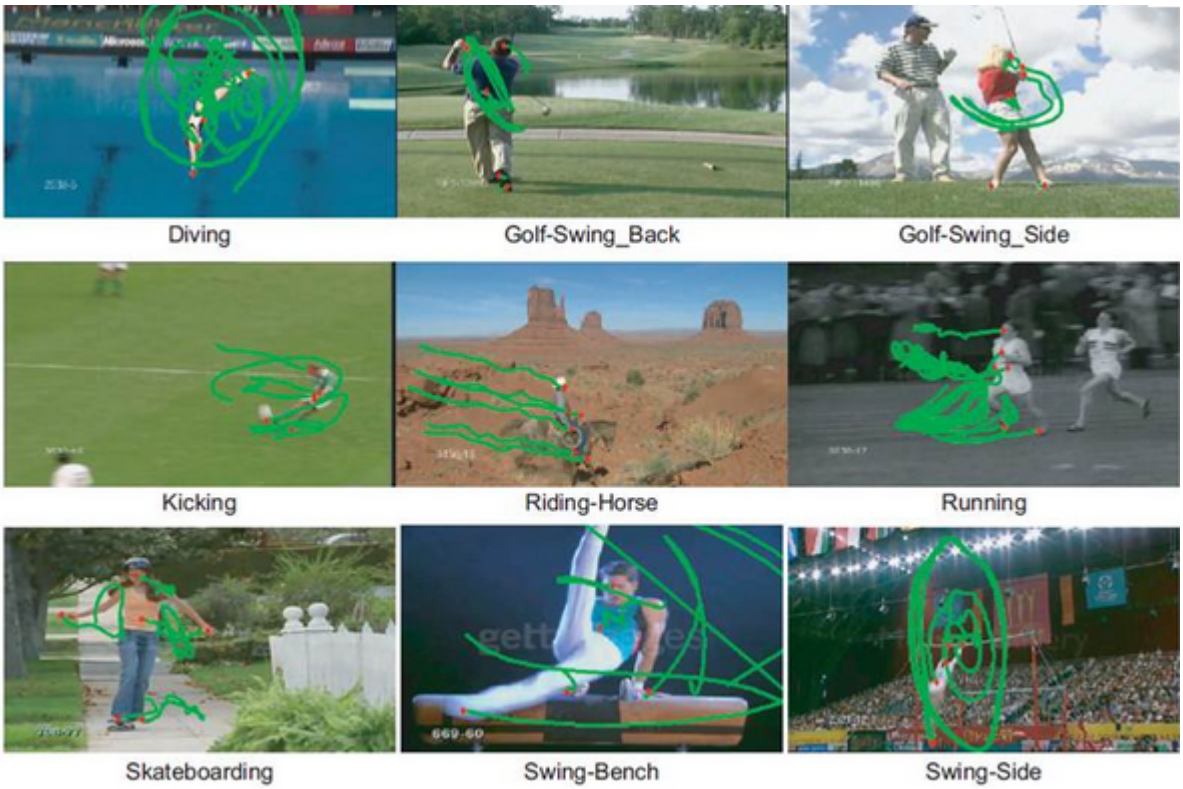


Figure 2.14. UCF dataset actions.

2.7.4 UCF101 Dataset

The final experiments are carried out on a large realistic dataset called UCF101. It includes total number of 101 action classes which we have divided into five types:

- Human-Object Interaction.
- Body-Motion.
- Human-Human Interaction
- Playing Musical Instruments.
- Sports.

UCF101 is an extension of UCF50, which included the following 50 action classes, [104]: Baseball Pitch, Basketball Shooting, Bench Press, Biking, Billiards Shot, Breaststroke, Clean and Jerk, Diving, Drumming, Fencing, Golf Swing, High Jump, Horse Race, Horse Riding, Hula Hoop, Javelin Throw, Juggling Balls, Jumping Jack, Jump Rope, Kayaking,

Figure 2.15. UCF11 dataset actions.

Lunges, Military Parade, Mixing Batter, Nunchucks, Pizza Tossing, Playing Guitar, Playing Piano, Playing Tabla, Playing Violin, Pole Vault, Pommel Horse, Pull Ups, Punch, Push Ups, Rock Climbing Indoor, Rope Climbing, Rowing, Salsa Spins, Skate Boarding, Skiing, Skijet, Soccer Juggling, Swing, Tai Chi, Tennis Swing, Discus Throw, Trampoline Jumping, Volleyball Spiking, Walking a dog, Yo-Yo.
The following 51 new classes are introduced in UCF101: Applying Eye Makeup, Applying Lipstick, Archery, Baby Crawling, Balance Beam, Band Marching, Basketball Dunk, Blow Drying Hair, Blowing Candles, Body Weight Squats, Bowlng, Boxing-Punching Bag, Boxing-Speed Bag, Brushing Teeth, Cliff Diving, Cricket Bowling, Cricket Shot, Cutting In Kitchen, Field Hockey Penalty, Floor Gymnastics, Frisbee Catch, Front Crawl, Hair Cut, Hammering, Hammer Throw, Handstand Pushups, Handstand Walking, Head Massage, Ice Dancing, Knitting, Long Jump, Mopping Floor, Parallel Bars, Playing Cello, Playing Daf, Playing Dhol, Playing Flute, Playing Sitar, Rafting, Shaving Beard, Shot Put, Sky Diving, Soccer Penalty, Still Rings, Sumo Wrestling, Surfing, Table Tennis Shot, Typing, Uneven Bars, Wall Pushups, Writing On Board.

Clip Groups: The clips of each action class are divided into twenty-five groups which contain four to seven clips each. The clips in each group share some common features, such

Figure 2.16. UCF 101 actions.

as the background or actors. The colors on each bar denote the durations of different clips included in that class. The chart shown in Fig. 2.17, illustrates the average clip length (green) and total duration of clips (blue) for each action class.

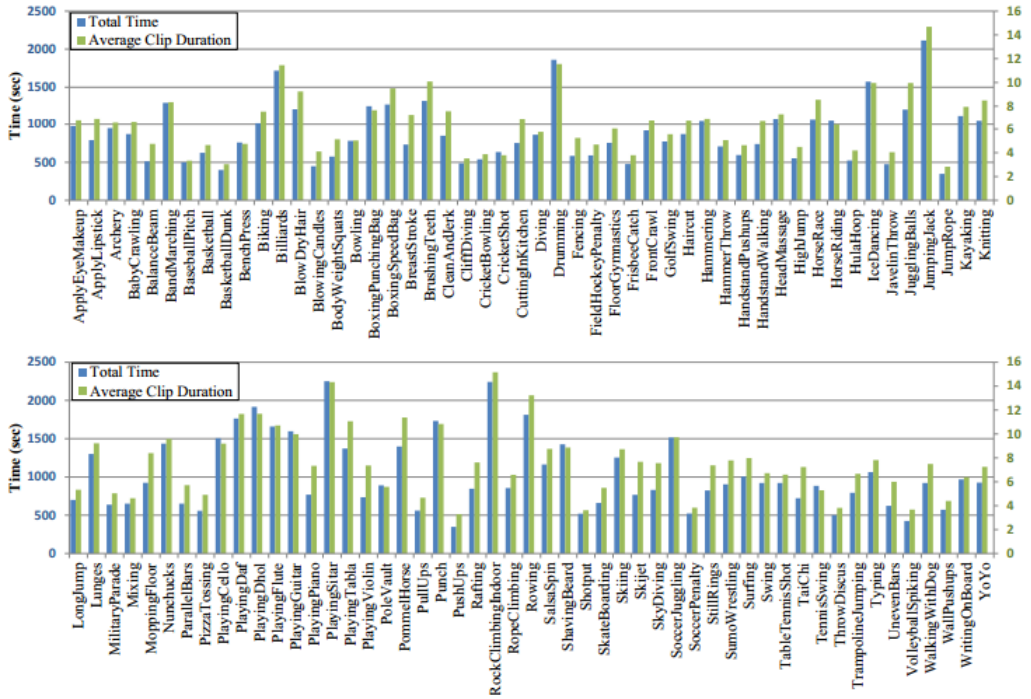


Figure 2.17. UCF 101 action duration.

The videos are downloaded from YouTube [10] and the irrelevant ones are manually removed. All clips have a fixed frame rate and resolution of 25 FPS and 320×240 respectively. See Fig.2.18.

2.8 Conclusion

This section has presented the most relevant works related to Human Action Recognition. It has been shown that the first step toward action recognition is feature extraction techniques. Relevant feature extraction methods have been reviewed. Video segmentation techniques were highlighted. We can conclude that recent approaches are looking to capture valuable information for the action recognition with respect of time consumption. They converge to an automatic action recognition in big datasets. As we can see, the problem of human action recognition is not close from being solved yet and new contributions to the field have to be proposed in future years.

Actions	101
Clips	13320
Groups per Action	25
Clips per Group	4-7
Mean Clip Length	7.21 sec
Total Duration	1600 mins
Min Clip Length	1.06 sec
Max Clip Length	71.04 sec
Frame Rate	25 fps
Resolution	320×240
Audio	Yes (51 actions)

Figure 2.18. UCF 101 characteristics.

CHAPTER 3

SPATIO-TEMPORAL SURF

Happiness lies in the joy of achievement
and the thrill of creative effort.
Franklin D. Roosevelt

Contents

3.1 Introduction

In this chapter, we propose a new spatio-temporal descriptor we called ST-SURF. It is based on a novel combination of the speed up robust feature (SURF) and the optical flow. The Hessian detector is employed to find all interest points. To reduce the computation time, we propose a new methodology for video segmentation into Frame Packets (FPs), based on the interest points (IP) trajectory tracking. We only consider moving interest points descriptors to generate robust and powerful discriminative code-book based on K-mean clustering. We use a standard bag-of-visual-words Support Vector Machine (SVM) approach for action recognition.
The first step to describe videos is descriptor extraction. To extract video descriptors, many researchers have been investigating in tracking major parts of human bodies then extracting features from these regions [105]. However, they need to setup many hypothesis. These hypothesis are often difficult to set. So that, methods based on spatio-temporal features are promising for action recognition.
Some of them are based on the extraction of low-level optical flows from cuboids[92] this method gives good results in terms of feature selection and a good classifications accuracy [92]. But they presents limits concerning the long computational time they require [13]. Dollar et al. detect local cuboids to apply 1-D Gabor filters in the temporal direction and 2-D Gaussian kernels in the spatial space [36], and they produce video visual words based on vector-quantization in the same way as bag-of-visual-words for object recognition [98].

In the same direction, Laptev et al proposed STIP (Spatio-Time Interest Points) to detect cuboids [46]. This method is considered as an extension of Harris detector. Nevertheless, the limits of the aforementioned methods not only concerns the hardness of finding the best cuboid size, but also the high computational requirements [13]. In view of the above and to overcome these problems, we propose to detect interest points using SURF/Hessian [24]. Then we segment the videos into Groups of interest points (GIP) and Frame Packets (FPs) to reduce the computation time. We use Sun, D at al. [106] optical flow detection methods which allows to extract spatio-temporal SURF by tracking interest points instead of cuboids.

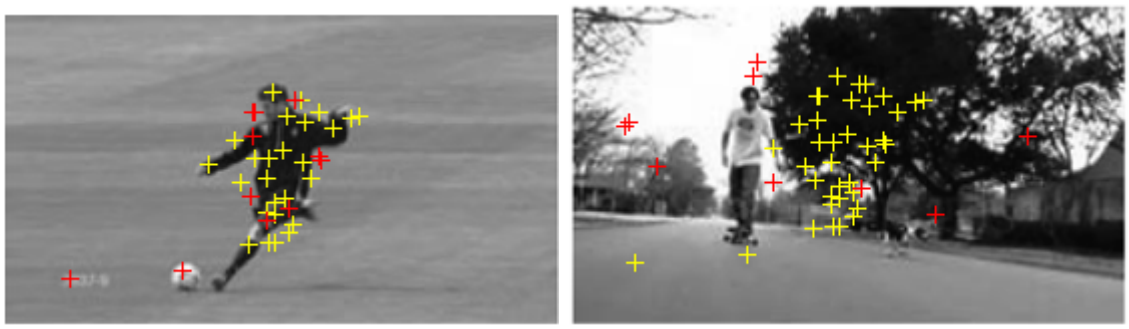


Figure 3.1. Example of SURFs found using Hessian detectors on different frames from UCF sports dataset.

3.2 The proposed approach for human action detection and recognition

The proposed method aims at detecting human actions, to reach this goal, first video sequences are segmented in Frame Packets (FPs) and Group of Interest Points (GIP). Second, based on a novel combination of optical flow computed by [106] and the local descriptor called the **S**peed-up-**R**obust-**F**eature (SURF) bay2006surf, Fig. 3.1, the interest points **ST-SURF** are localized and extracted, from all training video FPs. Then, the extracted ST-SURFs are clustered using K-means clustering algorithm. The video clips are represented as a K-bins histogram of the quantized descriptors "bag of spatio-temporal visual words" BoSTVW. Finally, an SVM classifier is trained using these histograms, Figure 3.2

3.3 Speed up robust features SURF

Generally descriptors extraction is achieved in tow major steps. As first step to SURF extraction is to analyze the video frames to discriminate salient regions as IP. These points are considered salient as they, naturally, attract the attention of humans and shows significant human movement. The major purpose of an IPs detection algorithm is to attempts to insulate regions of the video frames that have remarkable visual information, such as edges, corners. The reliability of an IP detector rely of its robustness against scale, rotation and view points changes. For example, in Figure 3.3, a robust detector is able to detect the bird despite the changes in scale, background, rotation and view point difference.

Several IP detectors are proposed in the state-of-the-art [23, 41, 40, 13] . In 2004, Lowe [24], presented SIFT for extracting invariant features from images that can be robust

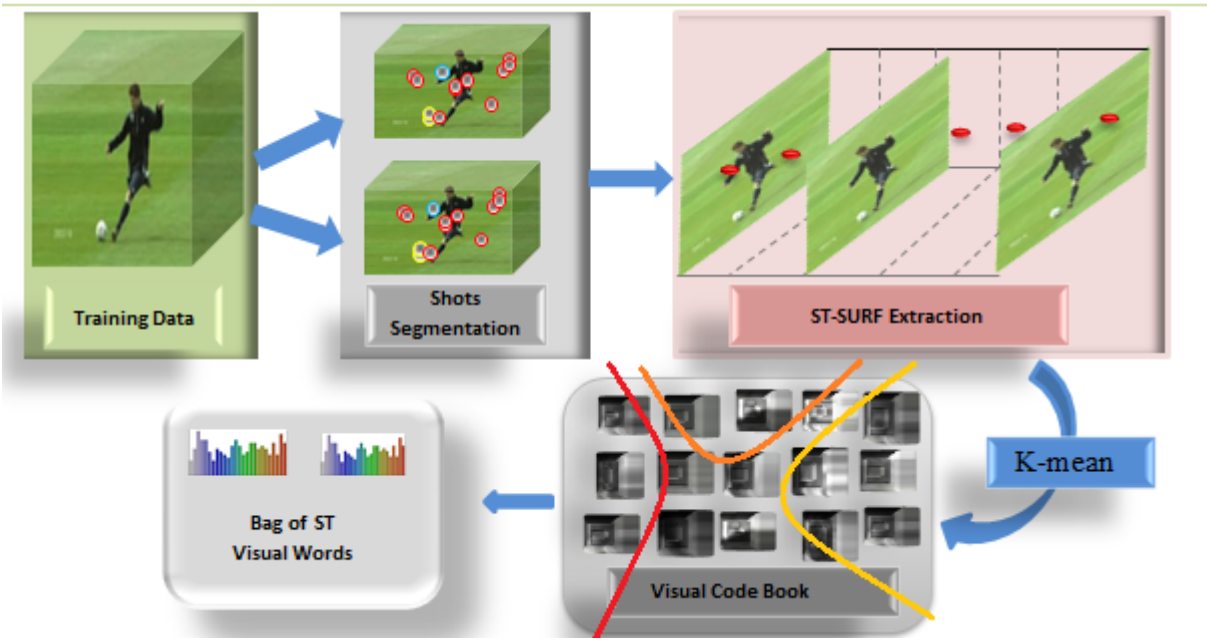


Figure 3.2. Training pipeline

Figure 3.3. Example of changes in scale and view point.

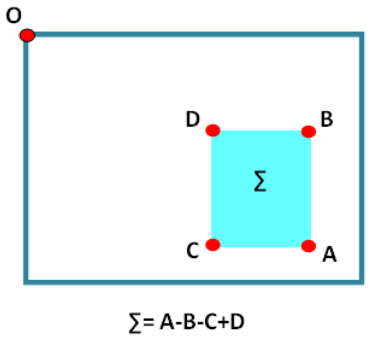


Figure 3.4. Integral image computation.

against image scale changes and rotation. Then it was widely used in image, recognition and retrieval etc. However, extracting robust features approaches are very slow [24]. Bay et al. speeded up SIFT by using integral images for image convolutions and Fast-Hessian detector [24]. Their experiments turned out that SURF was faster and it works well. We use the extraction solution given by [24] to extract interest point feature. This choice is motivated by the robustness, the smaller size of this feature and their excellent performances attested in various datasets for action recognition [13]. The SURF feature is a 64-D vectors that describes spatial patterns around detected points.

3.3.1 integral images

The integral images are a representation of a given image that allows for the fast implementation of box type convolution [107]. In an integral image the value of a pixel at a given point "p" is the sum of all pixels located in the rectangle formed by the origin till the point "p" of the original image. Once the integral image is computed, only four additions are required to compute the sum of pixels intensities of any upright rectangular region of the original image regardless size, Figure 3.4.
Let I being an input image, $I_{\sum x}$. A point p located at $p = (x, y)$ represents the sum of all pixels in the input image I of a rectangular region formed by the point a and the origin such as:

$$I_{\sum p} = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j) \tag{3.1}$$

$$H(f) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1{}^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n{}^2} \end{bmatrix}$$

Figure 3.5. Matrix of second derivative.

3.3.2 Fast-hessian detector

For interest points detection we choose the Hessian detector [108]. It searches for image locations that exhibit strong derivatives in two orthogonal directions. It is based on the matrix of second derivatives, the so-called Hessian (HM) [108], see Figure 3.5.

In fact, HM is not only fast and accurate, but it also allows to extract both scale and location cues [24]. For a given $IP = (x, y)$ located in a frame f, the HM located at IP with the scale σ is defined as

$$H(IP, \sigma) = \begin{pmatrix} L_{xx}(IP, \sigma) & L_{xy}(IP, \sigma) \\ L_{xy}(IP, \sigma) & L_{yy}(IP, \sigma) \end{pmatrix} \tag{3.2}$$

Where $L_{xx}(IP, \sigma)$ is the result of the convolution of the frame f in IP with the Gaussian second order derivative $\frac{\partial^2 g(\sigma)}{\partial x^2}$. This filter is approximated by using box filter (see Figure 3.6). Henceforth the determinant of the approximated HM becomes

$$det(H_{approx}) = D_{xx}D_{yy} - (0.9D_{xy})^2 \tag{3.3}$$

The "box filters" are introduced in [24]. They consists on an approximation of the Gaussian second derivatives called "box". Using integral images, the box filters can be evaluated fast and the computation time is independent of the size of the filter, Figure 3.6.

One of the major challenges faced by an interest point detector is the scale changes. In fact, the performances of a detector are evaluated according to its robustness against scale changes. A robust detector must be able to find salient points at different scales (the same object can be represented in different sizes in different frames). This is often handled by creating a pyramid images such as the scale invariant feature transform SIFT [24], the difference is illustrated in Figure 3.7. In such case the images are repeatedly filtered with

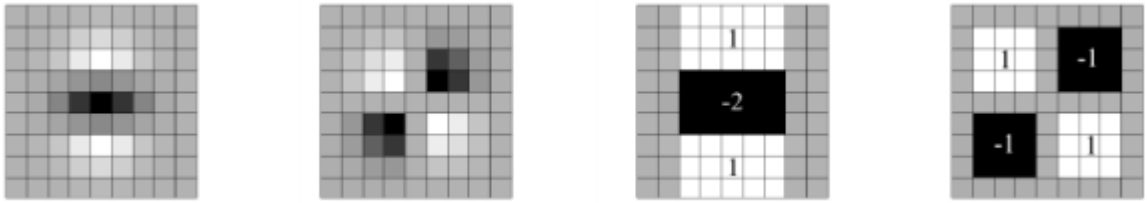


Figure 3.6. Partial derivative of the Gaussian. First discretized (both left images) and then approximated by a box filter according to y and xy direction. The gray areas are equal to zero.

a Gaussian and then sub-sampled into smaller image size. Each level of the pyramid define a scale.

The SURF has the advantage to proceed differently due to box filters and integral images. Instead successively applying the same filter to the output of a filtered and sub-sampled image using scale space representation based on pyramid decomposition. In [24] the filter size is up-scaled while keeping the original image size (See Figure 3.7). The "blob" response maps at different scales are constructed by enlarging the filter rather than reducing iteratively the image size. This allows on one hand to reduce the computation time on an other hand to avoid an eventual aliasing due to the under-sampling of the image, Figure 3.7).

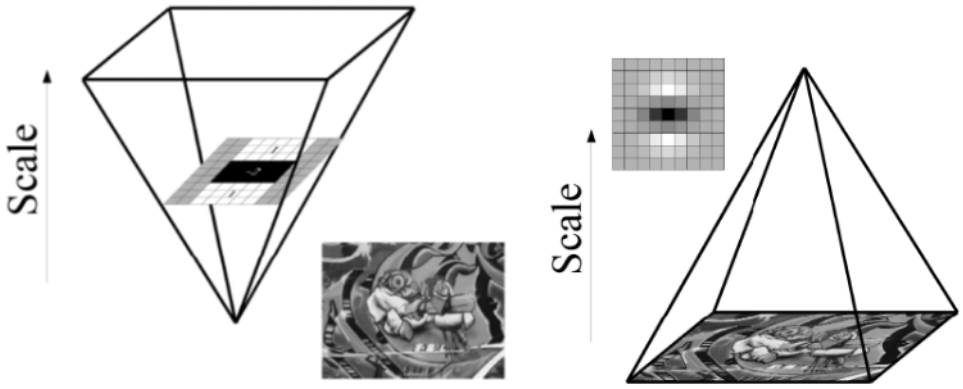


Figure 3.7. A general outline of SIFT SURF extraction. (a) Scales space in the SURF extraction; (b) scale space in the SIFT extraction

In order to localize interest points in the image and over different scales, a gradually increasing filters are applied. Generally the smallest filter used is a 9×9 filter considered

as the initial scale layer as in Figure 3.8. In this layer many detected IP are not relevant and unnecessary. The number of interest points detected decreases with the increase of masks sizes.

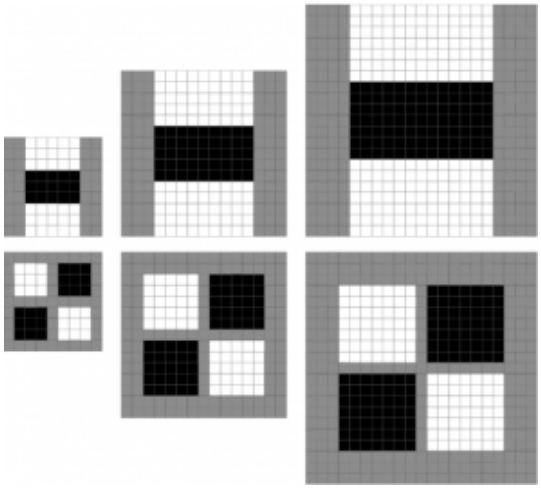


Figure 3.8. Representation of different scales filters

For every detected IP, a square region is defined. It is centered by the detected interest points and characterized by a reproducible orientation. This region is then divided into 4×4 sub-regions. A four Haar wavelet responses are extracted from every sub-region in both x and y directions. They are then weighted by a Gaussian window centered on the IP and represented as a point in space. The response value of the horizontal (x direction) represent the IP abscissa and the ordinate represents the value response from the vertical (y direction). The major orientation vector is obtained by calculating the sum of all the wavelet responses located in a $\pi/3$ window rotating around the center of the region of interest, Figure (3.9). The direction of the longer vector defines the main direction of the region of interest as shown in Figure 3.10.

3.3.3 SURF extraction

The SURF extraction is made in several steps:

- the first step rectangular region centered around the IP and oriented along the main direction selected is build. The size of this window depend on the actual scale. An example of these windows is shown in Figure 3.11.
- Every rectangular region is divided into 16 sub-regions (see Figure 3.11). In each of these sub-regions, responses to a Haar wavelet are calculated. Then a vertical and

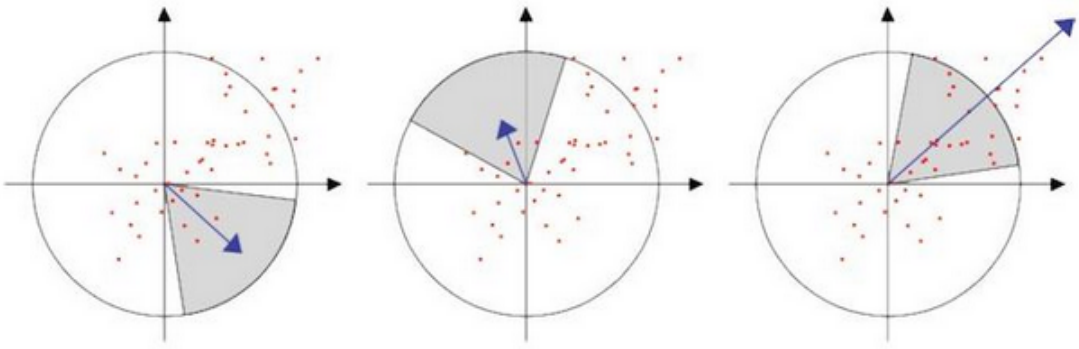


Figure 3.9. Dominant direction

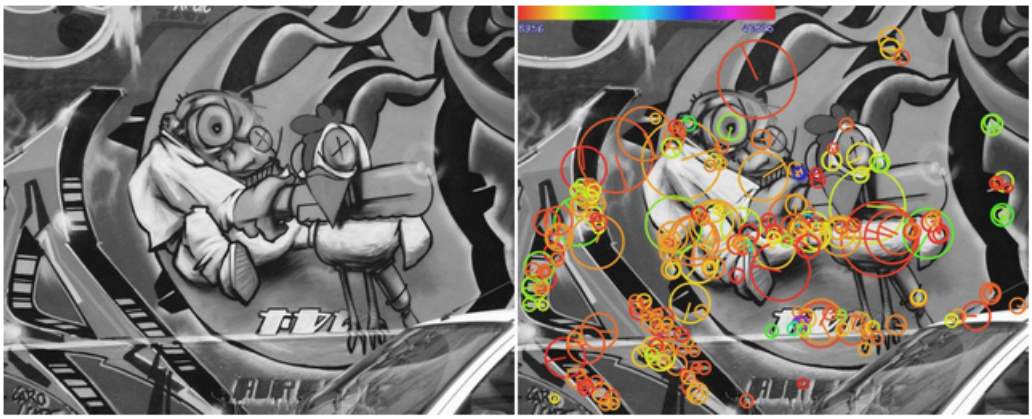


Figure 3.10. Detected SURF descriptors with the main direction choice

horizontal directions are defined based on the orientation of the area of interest. The response in the horizontal direction of the selected sub-region is denoted d_x and d_y in the vertical direction.

- The weighted responses are summed for each sub-region into $\sum dx$ and $\sum dy$. To add additional information about changes in intensity, the sums of the absolute values of the weighted responses are also extracted $\sum |dx|$ and $\sum |dy|$. This results in a descriptor vector for all 4×4 sub-regions of length 64. Figure 3.12 shows three sub-regions showing different intensities levels. Indeed, in the first case in the upper left is an uniform intensity sub-region, $\sum dx$, $\sum dy$, $\sum |dx|$ and $\sum |dy|$ are very low. In the second case presenting different intensity levels we notice two types of responses: either very positive (transition from black to white) or very negative (transition from white to black). In the third case in the right sub-region with gradually increasing intensity, we always have a positive response (transition from black to white) but

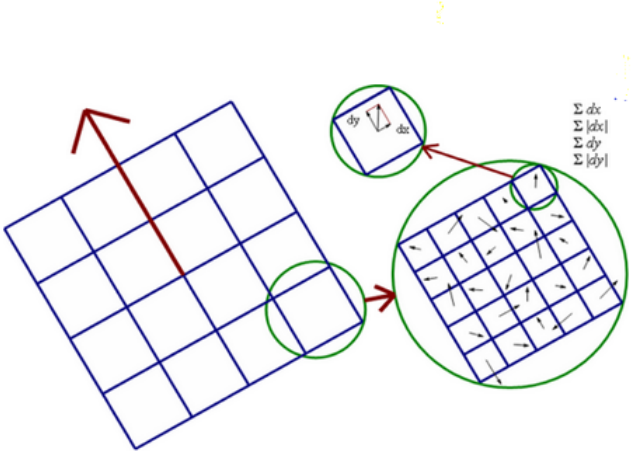


Figure 3.11. Detected SURF rectangular regions oriented following the main direction.

quite small. We thus find good $\sum dx$ and $\sum |dx|$ are equal and quite significant.

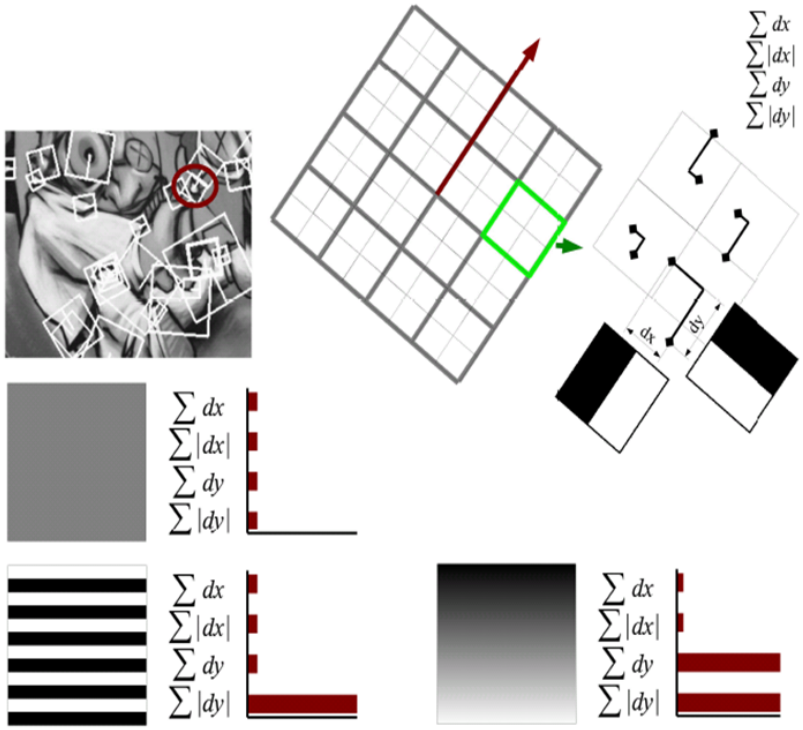


Figure 3.12. The descriptor entries of a sub-region represent the nature of the intensity pattern. upper-Left: In case of a homogeneous region, all values are low. Middle: In presence of frequencies in x direction, the value of $\sum |dy|$ is high, but all others are low. If the intensity is gradually increasing in x direction, both values $\sum dy$ and $\sum |dy|$ are high.

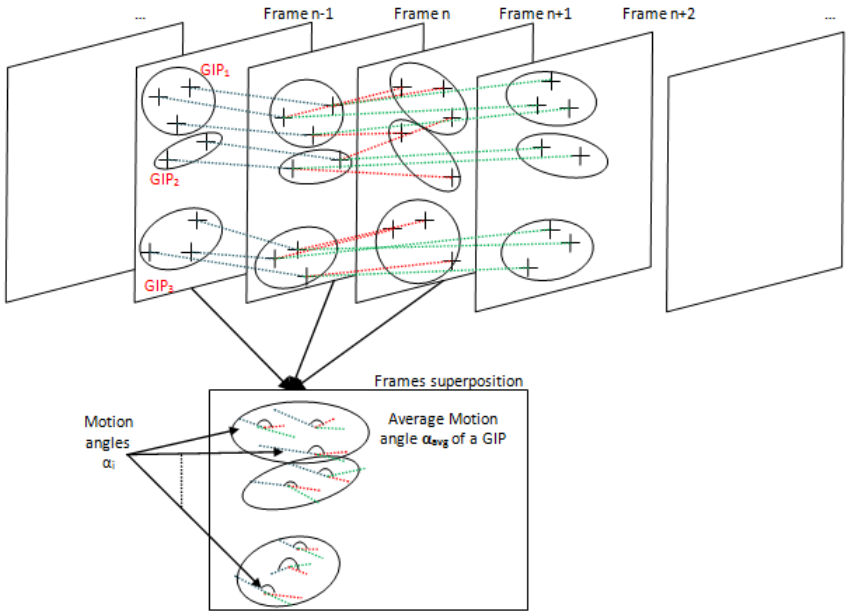


Figure 3.13. IPs trajectory tracking for FPs segmentation

Finally the 64D SURF descriptor is extracted. The experiments turned out that SURF is three times faster than SIFT with a reasonable accuracy.

3.4 Frame packets (FPs) and group of interest points (GIP) segmentation

To track human action into videos, key frame extraction techniques such as the ones proposed in [109] can be exploited. Other methods are based a video segmentation into patches or snippets [13, 110, 12]. Noguchi et al. choose to divide video sequences into snippets of five frames [13]. In this thesis, we propose and use the concepts of Frame Packets (FPs) and Group of Interest Points (GIP). We assume that, between three successive frames (n-1, n and n+1), an interest point (from one frame to another) can have three possible states: still, moving and disappear. The first and last states are not treated. In fact, in the first case, no motion is detected. In the last case the interest point IP disappears and cannot be tracked anymore. The second state is the one that concerns us the most, since there is a displacement and we can track the displacement angle. From now on, we assume that α is the angle between the lines segments supporting the motion of an IP from the couple of frames (n-1, n) and (n, n+1), Figure 3.13.

By comparing α to α_{max} (a parameter fixed experimentally at the beginning of the

processing) we are able to segment a succession of frames, that we call here Frame Packets (FPs), in which each IP has an α lower than α_{max}. By calibrating this angle of tolerance, we are able to certify that, within this FP, all IPs movements are within this tolerance parameter. This means that we cannot miss any significant movement likely to influence the remaining computing. In order to be able to have more control over the size (in number of frames) of the FP, we introduce, the concept of Group of Interest Points (GIP). In fact, a NGIP is a parameter defining the number of IP that must be grouped together (this constant has to be experimentally fine-tuned to find the most suitable N). This grouping is performed over successive IP in a frame. By defining this number NGIP we can compute an average angle (α_{avg}) for a certain GIP and compare it to the α_{max}. The higher NGIP is the less the α_{avg} will be sensitive to motion and the more the FP will contain frames. Here are the steps of our segmentation algorithm. Let us suppose that we are beginning the computation of a new FP:

- We extract the IP of the frames one and two.
- We define the GIPs based on the NGIP parameter fixed at the beginning of the algorithm.
- We compute the line supporting the motion for each corresponding IP within these two frames.
- We apply the above three steps to the frames two and three.
- We compute the angle between each motion line and we extract the average angle for each GIP.
- We compare each average angle to the α_{max} (fixed at the beginning of the algorithm).
- We continue performing the above six steps over the next frames (taking, always, the first motion direction as reference to all remaining comparisons) until finding an average angle of a GIP higher than the maximum angle. In this case we can define the FP and assume, with confidence, that the first and last frames of this FP can fully describe the motion within.

3.5 SURF tracking into 3D feature space

Features' tracking is performed by estimating optical flow. To increase optical flow estimation accuracy, many researches are inspired from the Horn and Schunck (HS) Optical

flow formulation. In fact, they focuses on optimizing an objective function which combines the image's properties and its spatial motion prediction. Sun end al. [106] proposed a new algorithm to approximate an optimized computationally tractable objective function, based on the original HS formulation. They first, use median filtering to denoise the flow, Exploiting connections between median filtering and L1-based denoising. They proved that algorithms relying on a median filtering step are approximately optimizing a different objective that regularizes the flow over a large spatial neighborhood [106]. The resulting algorithm ranks 1st in both angular and end-point errors in the Middlebury evaluation in March 2010 [106].

In our work, we considered every Frame Packet as a volume of frames in the 3D space called FP Volume (FPV), this cubic volume is characterized by its frames' number(FN) going from 1 to t_{max}, its frames' surfaces dimensions (FS) and its center ($FPVc$). A given interest point $IP = (x, y, t)$ is defined by its position (x, y) and its frame t. In frame $(t + n)$, the IP moves by a displacements u in the x direction, and v in the y direction. IP becomes, $IP(t + n) = (x + u, y + v, t + n)$. In all our experiments, unless mentioned otherwise, we assume that due to the video segmentation into FPs, the motion vectors trajectory remain stable. For stagnant interest points $u = v = 0$. Thus, in the FPV, the 3D direction (u, v, n) represent the direction of the IP motion. The motion vector is calculated by the Sun et al. [106] optical flow approach.

Our contribution consists on the use of motion orientation and position to characterize the motions, instead of using the direction vector (u, v, n) generated from optical flow computation. We suppose that the motion vector in the 3D space can be defined as the intersection of two planes perpendicular respectively to the plane (t, x) and the plane (t, y). This parametrization is one among several possible representations of 3D lines [110]. To extract IP orientation, we project its motion vectors onto the planes (t, x) and (t, y) of the FPV to define an angle for each projection, the first angle α_x between optical flow and the plane (t, x), the angle α_y between the plane (t, y) and the motion vector.

$$\propto_x = 90 - \frac{180}{\Pi} arctan(u), \propto_y = 90 - \frac{180}{\Pi} arctan(v). \tag{3.4}$$

For each IP, we project its motion vector onto the planes (t, x) and (t, y) and obtain two lines L_x and L_y. The orthogonal projection of $FPVc_x$ and $FPVc_y$ onto the lines L_x and L_y allows the computing of both distances D_x and D_y between the cube center and the lines supporting the motion vectors (L_x and L_y).

For an IP located at (x, y, t):

$$D_x = D_{xu} - D_{tv}, D_y = D_{yv} - D_{tu} \tag{3.5}$$

where

$$D_{xu} = (x - x_{max}/2)cos(180/\Pi arctan(u)) \tag{3.6}$$

$$D_{tv} = (t - t_{max}/2)sin(180/\Pi arctan(v)) \tag{3.7}$$

$$D_{yv} = (y - y_{max}/2)cos(180/\Pi arctan(v)) \tag{3.8}$$

$$D_{tu} = (t - t_{max}/2)sin(180/\Pi arctan(u)) \tag{3.9}$$

where t_{max}, x_{max} and y_{max} are the dimensions of the Frame Packet volume with t_{max} depend on the number of the frames into a segmented (FPV). In the following, D_x and D_y describe the motion distances of a given interest point. Figure 3.14, is a graphical illustration of the cube center and its projection into the planes (t, x) and (t, y).

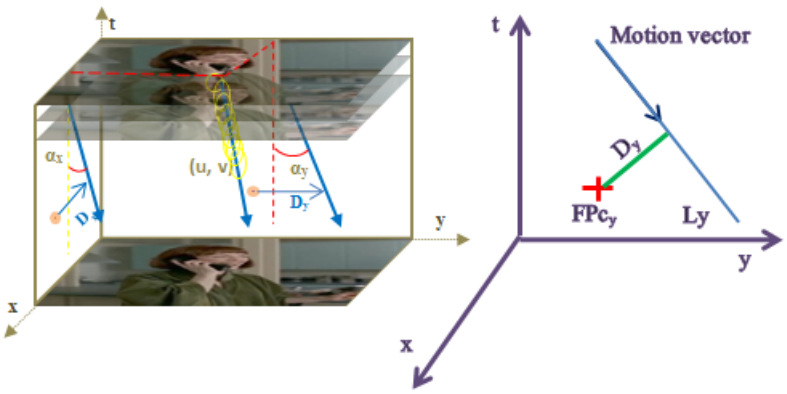


Figure 3.14. The projection of a motion vector in the adjacent planes.

3.6 ST-SURF extraction:

This step consists in the generation of the novel ST-SURF that we designed. This descriptor is represented by spatial feature 64-D vector, and temporal 4-D feature ($\propto_x, \propto_y$,D$_x$, D_y),

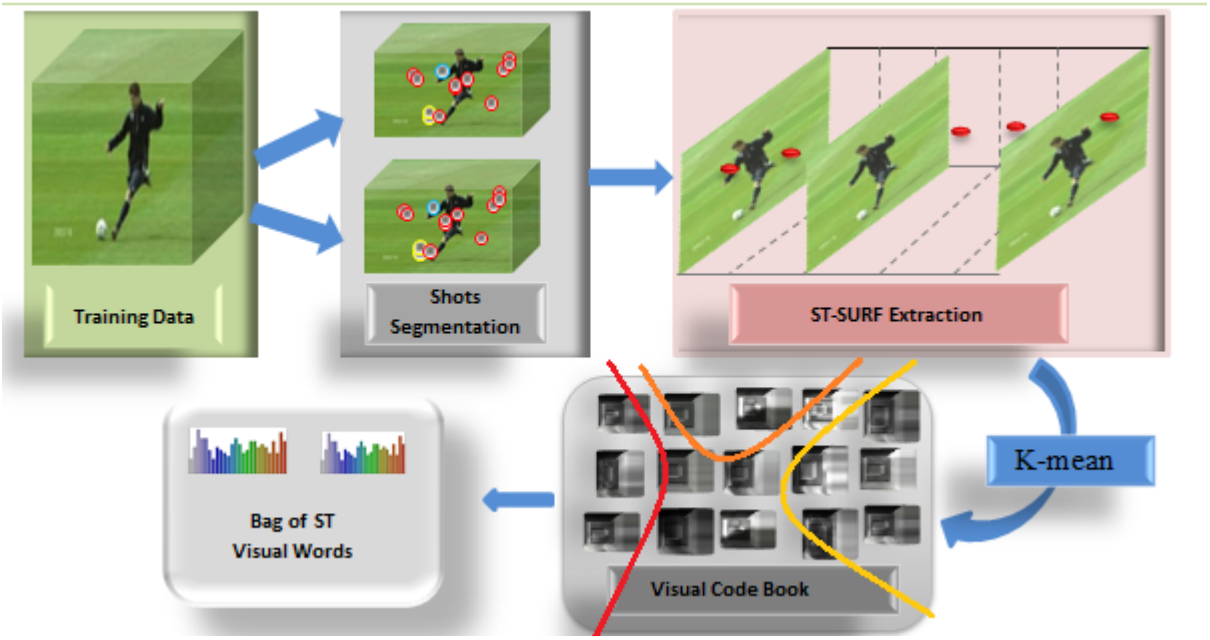


Figure 3.15. Training pipeline

we concatenate both vectors into one 68-D spatiotemporal descriptor vector, and thus we extend the image SURF descriptor [24] to videos. These features will tracks the interest point through time in each FP we defined. The size of a FP depends on its average frames number. In our work we consider only moving interest points (where $\propto_x \neq 0$ and $\propto_y \neq 0$).

3.7 ST-SURF training pipeline:

After the extraction of ST-SURF descriptors, we define a spatio-temporal words dictionary. The basic idea is to assign a set of objects into groups so that the objects of similar type will be in one cluster, in order to construct a visual code-book, which can be used to represent an action, a scene or en object. Recently, the K-means algorithm has been widely used to construct the visual code-book because of its high performances and simplicity. Fig. 3.15, illustrate the training steps of a given videos.

3.8 ST-SURF evaluation pipeline:

After the extraction step, the generated ST-SURFs are quantized into visual words using k-means clustering. Each video sequence can then be represented as the frequency histogram over the visual words. Generally, using a large-sized code-book allows to obtain high

recognition accuracy, yet an oversized code-book leads to high quantization errors. The resulting histograms of visual word occurrences are used as classification inputs. We use a non linear support vector machine to classify human actions.

3.9 Experiments

In the following we describe the datasets used for the evaluation of the proposed work. We evaluate the ST-SURF in a bag-of-features based action classification task and compare our approach to the state-of-the-art employ [23, 13, 41, 40]

3.9.1 Experimental setup and data

To demonstrate the performance of the proposed action recognition approach, we have tested our algorithm tow realistic datasets described below:

3.9.1.1 Dataset:

The proposed framework is tested on the KTH dataset [34] and UCF sports Dataset [102]. The KTH dataset is commonly used as a public benchmark test of spatio-temporal features [13]. This dataset contains six kinds of actions: walking, running, jogging, boxing, hand waving and hand clapping. We consider 6 action classes by 25 persons in 4 different scenarios with a total of 2391 video samples. The average length of videos in the KTH dataset is about 20 second long and about 500 frames. The second one is the UCF sports dataset, more realistic and challenging data obtained from broadcast sport videos by Rodriguez et al. [102]. The collection represents a natural pool of actions featured in a wide range of scenes and viewpoints. The publicly available part of this dataset contains nine actions namely diving, golf, swinging, kicking, lifting, horseback riding, running, skating, swinging and walking. This dataset contains close to 200 video sequences at a resolution of 720x480 [102].

3.9.1.2 Parameter settings:

In all our experiments, we explored optimal parameter settings. We evaluate the classification rates of both KTH and UCF datasets while changing the codebook and the FPs sizes. The results shows that the empirically optimal size book is k = 4000 with $\alpha_{max} = 24\circ$ and NGIP = 38. These settings gave us empirically satisfactory results.

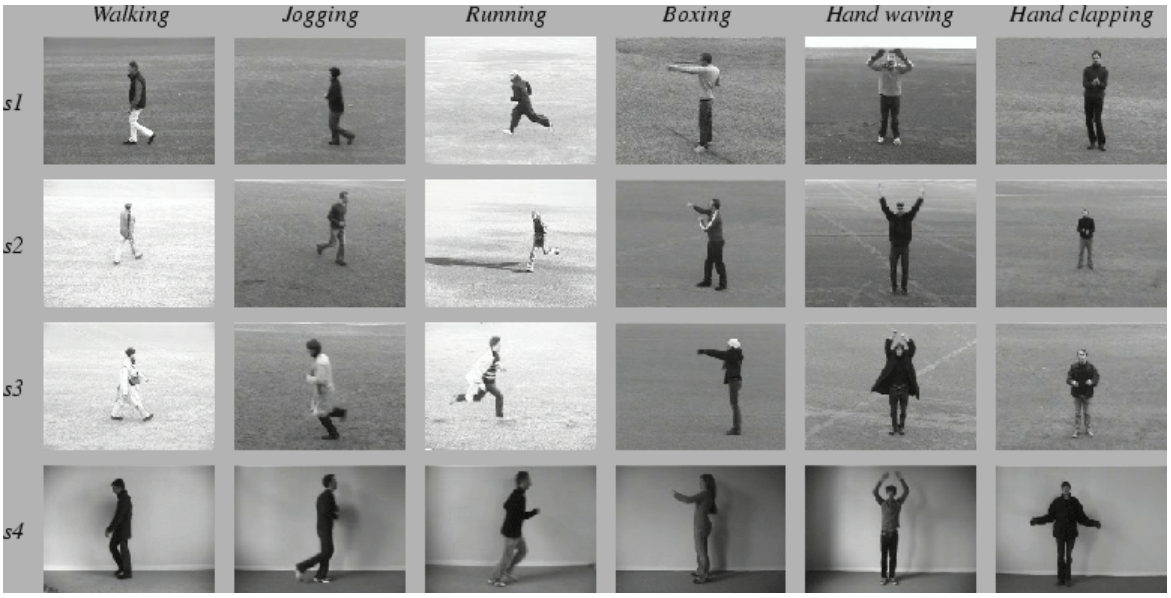


Figure 3.16. KTH dataset actions [34].

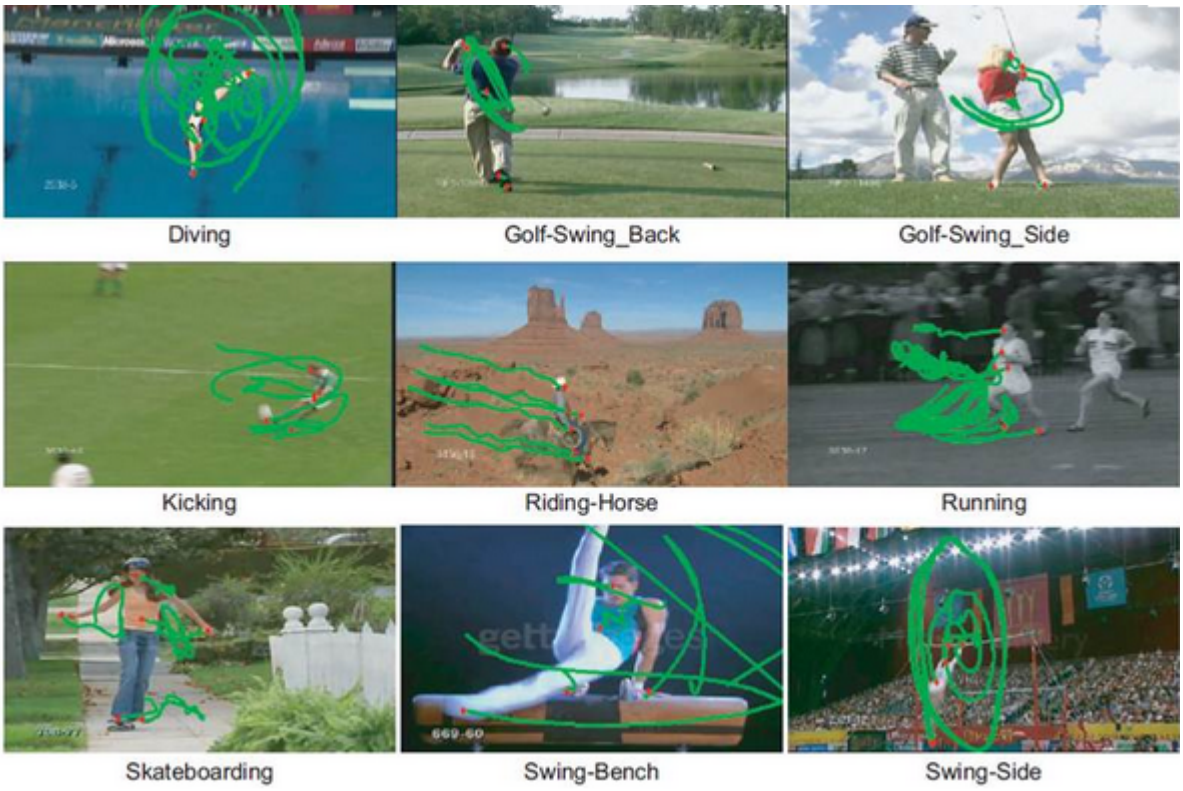


Figure 3.17. UCF dataset actions [102].

Table 3.1. Average accuracy for various detector/descriptor combinations on the KTH dataset.

.	HOG3D	HOG/HOF	HOG	HOF	E-SURF	t-SURF	ST-SURF
BAA	90%	91.8%	82.3%	92.1%	81.4%	86%	**88.2%**
HAA	84.6%	88.7%	77.7%	88.6 %	81.4%	86%	**88.2%**

Table 3.2. Average accuracy for various detector/descriptor combinations on the UCF sports dataset.

.	HOG3D	HOG/HOF	HOG	HOF	E-SURF	t-SURF	ST-SURF
BAA	85%	81.6%	77.4%	82.6%	77.3%	-	**80.7%**
HAA	78.9%	79.3%	66.0%	75.3%	77.3%	-	**80.7%**

3.9.2 Experimental results on KTH dataset:

In the table 3.1, the first row compares the **b**est **a**verage **a**ccuracy (**BAA**) for the different detector/descriptor combinations reported by other researchers, on the KTH dataset. The **a**verage **a**ccuracy for **H**essian (**HAA**) detector/descriptor combinations on the KTH dataset are drawn in the second row.
From the recently reported results of the state-of-the-art, we can clearly conclude that using Hessian detector, Laptev et al. [23] obtained 88.7% using a combination of HOG (histograms of gradient orientations) and HOF (histograms of optical flow) descriptors, 88.6% using HOF, and 77.7% with HOG . We note that Kläser et al. [41] achieved an accuracy of 84.6% using HOG3D descriptor, which is a comparable results with HOG/HOF [23]. The combination SURF/Hessian detector gives 84.6% for williems et al. [40], 86% for Noguchi [13].

3.9.3 Experimental results on UCF dataset:

We note that Kläser and al. [41] achieved an accuracy of 85% using HOG3D/Gabor descriptor, Laptev and al. [23] obtain 81.6%using HOG/HOF, 82.6% using HOF, and 77.4% with HOG. In the first row of Table 3.2, we compare the **b**est **a**verage **a**ccuracy (**BAA**) for the differents detector/descriptor combinations reported by other researchers, on the UCF sports dataset. The **a**verage **a**ccuracy for **H**essian (**HAA**) detector/descriptor combinations on the UCF sports dataset are drawn in the second row.

Tables 3.3 and 3.4 are the confusion matrices of the actions classification results based on two type of features. the first matrix describe the classification result for the visual SURF

Table 3.3. SURF confusion matrix action recognition on the KTH dataset.

KTH	Boxing	clapping	Waving	Walking	Jogging	Running
Boxing	0.6	0.01	0.03	0.13	0.1	0.1
clapping	0.06	0.58	0.25	0.04	0.02	0.05
Waving	0.05	0.08	0.74	0.03	0.01	0.09
Walking	0.01	0	0	0.7	0.16	0.13
Jogging	0	0.01	0	0.12	0.58	0.29
Running	0	0	0	0.1	0.21	0.59

Table 3.4. ST-SURF confusion matrix action recognition on the KTH dataset.

KTH	Boxing	clapping	Waving	Walking	Jogging	Running
Boxing	0.9	0.07	0	0.03	0	0
clapping	0. 07	0.9	0.03	0	0	0
Waving	0.01	0.06	0.93	0	0	0
Walking	0	0	0	0.91	0.06	0.3
Jogging	0	0	0	0.04	0.85	0.1
Running	0	0	0	0.06	0.14	0.8

Table 3.5. ST-SURF confusion matrix action recognition on the UCF sports dataset.

UCF	Dive	Golf	Kick	Lift	Ride	Run	Skate	Swing	Walk
Dive	0.8	0.17	0	0	0	0	0	0.03	0
Golf	0	0.78	0.2	0	0	0	0	0	0.02
Kick	0	0	0.9	0	0	0.07	0	0	0.03
Lift	0	0	0	0.92	0	0	0	0.08	0
Ride	0	0	0.2	0	0.62	0.18	0	0	0
Run	0	0	0.02	0	0	0.88	0	0	0 .1
Skate	0	0	0.08	0	0	0	0.6	0	0.32
Swing	0	0	0	0	0	0	0.21	0.79	0
Walk	0	0	0	0	0	0.04	0	0	0.96

feature reported in [13]. The result among a single visual feature give bad classification results for all the actions. Based on Table 3.5, confusion matrix show that the original combination, that we proposed, of both visual and motion features (ST-SURF) boosted significantly the classification accuracy. Regarding the average result over each of the six actions' KTH dataset, ST-SURF produced good result, however, less accuracy is observed in the jogging and running actions because these actions are almost similar. lastly but not least, comparing with results driven by the best result of the state-of-the-art, our method achieve 88.2% better than the 86% reported by Noguchi et al. [13] using Spatio-temporal SURF. Outperforming the results of the Cuboids/HOG combination obtained by [46] 82.3% and the 81.4% reported by Willem et al. [40]. Based on the confusion matrix of UCF sports Dataset given in Table. 3.5, the ST-SURF outperform the best result driven

by the state-of-the-art using Hessian detector and achieves 80.7% of accuracy. We note that ST-SURf/Hessian gave better results in realistic videos. We are still below the results driven by Laptev and al. with 85% using the HOG3D/Gabor combination, and their 91.8% reached using Harris3D/(HOG/HOF), this can be due to different code-book generation and the use of different interest points detectors. This motivates further investigations of different interest points detectors and realistic video settings. Regarding all these results our method is equivalent to the state-of-the-art [23, 41, 40, 13] and shows significantly better performance, outperforming many results driven in the same setup.

3.10 Conclusion

We have investigated a novel scheme to efficiently segment video sequences into a new concept we called Frame Packets. Then we proposed a novel spatio-temporal descriptor based spatio-temporal interest points. The designed descriptor is an extension of the SURF to the temporal domain. The proposed feature extraction consists on detecting of the Surf points and mapping them into a 3D feature space based on an original exploitation of the optical flow orientation and position. Only the moving SURF are then selected. The extracted features are embedded into a bag of visual word pipeline, to finally classify six actions from KTH dataset and then nine actions from the UCF sport dataset. Furthermore, the proposed framework demonstrate promising recognition performance on tow standard benchmarks with the accuracy about 88.2% in KTH and 80.7% in UCF sports. In this chapter we tested the proposed descriptors in a controlled dataset then a realistic small dataset. In the next chapter we propose to enrich the action detection step and to introduce more spatio-temporal descriptors. We propose also to fuse several descriptors in order to handle camera motion effects and to work with more complex and challenging video benchmarks.

CHAPTER 4

TRAJECTORY BASED ACTON RECOGNITION

Le grand orateur du monde,
c'est le succès.
Napoléon Bonaparte

Contents

4.1 Introduction

Human activity understanding and recognition attracted considerable attention during the past decades. It plays a prominent role in a wide range of applications, including video analysis, surveillance video, gesture interpretation and content based video search, just to name a few. These various applications rely on action recognition systems. Significant advances in action recognition have greatly boosted the state of the art methods [9, 111]. Generally, the recognition of action can be realized by following the three major tasks described in the below:

4.1.1 Video temporal segmentation

While several researches are based on spatial segmentation [112]. Many video description methods rely temporal evolution by encoding the entire video sequence [113], [114]. This obviously leads to a huge number of descriptors. Most of them do not describe the action since they focus on non-moving humans/objects in the scenes. In other works video sequences are described by a fixed number of frames leading to a miss interpretation of the observed scene. In-fact, a fixed frame number can be exploited in a non realistic video dataset with one person performing one action with static background and discarding camera motion [34, 25]. However for realistic benchmarks like those introduced in [100, 115, 116], moving object disappear in many sequences due to occlusions or change in viewpoints. Moreover, actions can be continuous, not-continuous, superposed (jumping to avoid an obstacle while walking, or stop to drink while walking) etc.
In these cases a selective video segmentation needs to be addressed carefully. That is why, many approaches are based on the visual segmentation to rely on a significant video sequence rather than encoding the entire video [16] or a randomly fixed frame number. To avoid unnecessary computations, authors in [43], introduce a video segmentation into frames packets based on the trajectory tracking. In this case, authors perform video segmentation based on the SURF's motion trajectory tracking. In order to reduce computational load, due to exhaustive human action detection, selective video segmentation into snippets covering actions saliency in a video sequence is reconsidered.
In this chapter, we build on the results gained in the previous work of [43] detailed in the previous chapter. In fact we propose to track all moving objects/humans whose actions need to be recognized. A dense SURF extraction is then performed to capture the maximum of spatial information contained in the video frames. Then, a tracking process is

employed to extract selective video snippets describing the detected action. A motion angle is empirically settled to track significant motion and in the same time reduce camera motion effect by ignoring small horizontal displacements. This technique allows investigating a sufficient frame number to recognize significant human small actions "actionlets". The combination of ordered actionlets leads to describe an entire human action. We consider that every SS (describing an actionlet) forms a 3D volumetric cube (VC). The proposed technique offers various advantages on the computational complexity and time consumption. First, it is based on a limited number of relevant frames. Furthermore, it allows investigating the sufficient number of relevant frames to describe an actionlet. Second, it allows to track linear vectors of displacement to avoid additional trajectory shape feature computation.

4.1.2 Video description

A key success of an action recognition process is the choice of the relevant features for video description. Among the various types of features used for activity detection, the silhouette based features and the spatio-temporal local features are the most used [117]. The basic idea of silhouette based approaches is to track the evolution of a body shape over time, then extract the features describing this evolution [118, 43]. Since, the approaches rely on perfect body segmentation, they are sensitive to camera motion, occlusion and illumination changes [119]. Another group of approaches are based on spatio-temporal interest points. Almost all the proposed methods used for detecting spatio-temporal descriptors are based on two major approaches.

The first class of methods is based on the extension of a 2D interest point (IP) to the temporal domain (1D). Indeed, Willems et al. [40], proposed a method based on the extension of the Hessian matrix to the temporal domain and extract the determinant of a spatio-temporal hessian matrix to extract IP. Laptev and al.[44], extended the volumetric features corner detector to extract space-time local structures. In the same spirit, the 3D spatio-temporal volumetric feature was proposed by [120]. Local descriptor were also extended to the temporal domain such as the histograms of oriented 3D spatio-temporal gradients proposed in [41], the E-SURF descriptor in [40] and the 3D-SIFT feature introduced in [37]. Noguchi et al. [13], proposed a spatio-temporal SURF using Lucas-Kanade optical flow.

The limitation of these methods is that they handle spatial and temporal information in a common 3D space. However, they have different characteristics and associating them

differently in a new scheme deserve to be more investigated [33, 68].
In the second class of methods, various approaches are based on IP tracking upon a video sequence in order to detect spatio-temporal features. This approach provides excellent performances in activity recognition [33]. In fact, Sun et al. [67], proposed efficient action recognition by leveraging the motion information of trajectories. Sameh et al, in [43], proposed a method based on trajectory tracking of the SURF interest points from a frame packet. One of the latest work has been proposed in [33]. The proposed descriptors are based on appearance (histograms of oriented gradients), motion (histograms of optical flow) and trajectories to characterize shape (point coordinates).
A trajectory is the path that a moving object/human follows through time. Various trajectory based descriptors have been proposed in the last decades, [63, 58, 62, 61, 64]. The trajectories features can be extracted from optical flow [33, 43, 121, 92], or by matching the interest points in different frames [64, 33]. The number of the exploited frames to set the trajectory length depends on the used approaches. In [64], the authors propose to set the trajectory length into a fixed interval $Lmin \leq Lmax$ with $Lmin = 5$ frames, $Lmax = 25$ frames, not exceeding one second in duration. Wang et al. [68], propose a fixed trajectory length to extract a displacement vector. Several other methods propose a trajectory length depending on the trajectory shape, [122, 123].
In this thesis we dress the problem of human action recognition by introducing and evaluating a novel local spatio-temporal descriptor coined Histogram of motion trajectory orientation. For every detected SURF, we define the interest point neighborhood size related to the SURF scale. For the detected patch, we extract dense displacement field based on optical flow algorithm introduced in [106]. Motion trajectories orientations are then generated for every pixel by exploiting horizontal and vertical optical flow components (u, v). We split the optical flow components to extract the distribution of the motion trajectory orientation in the planes (t, x) and (t, y). The generated histograms describes the distribution of the trajectory orientation angle and its displacement into a SS. An early fusion step is used to associate each histogram to the corresponding SURF. Thus, we extract a substantial of cues about spatial and temporal evolution of a moving region of interest in a predefined SS. Note that, the spatial information is captured by the SURF descriptor.

4.1.3 Training/learning

In the context of human activity recognition in video, the representation of video objects as a dictionary of visual words is an active research area [97, 39, 84]. The notion of bag

of visual words (BoVW) has been introduced in [124]. The main weakness of the bag of visual words (BoVW) is that, the visual words are not all informative and accurate to describe an action. Therefore, one has to pay more attention to the selection of visual words. The most used approaches to select visual words are based on Machine learning techniques Boosting [18], random forest [95], adaptation process like Multiple Instance Learning (MIL) [19] or many other State-of-the-art algorithms [20, 21]. In this thesis we exploit the K-mean clustering algorithm to extract a codebook. The extracted visual words are classified based on a χ^2 Kernel Support vector machine (SVM). Non linear SVM is a fast and efficient algorithm that maps histograms in a higher dimensional space. SVM demonstrated high classification results under challenging realistic conditions, including intra-class variations and background clutter [125].

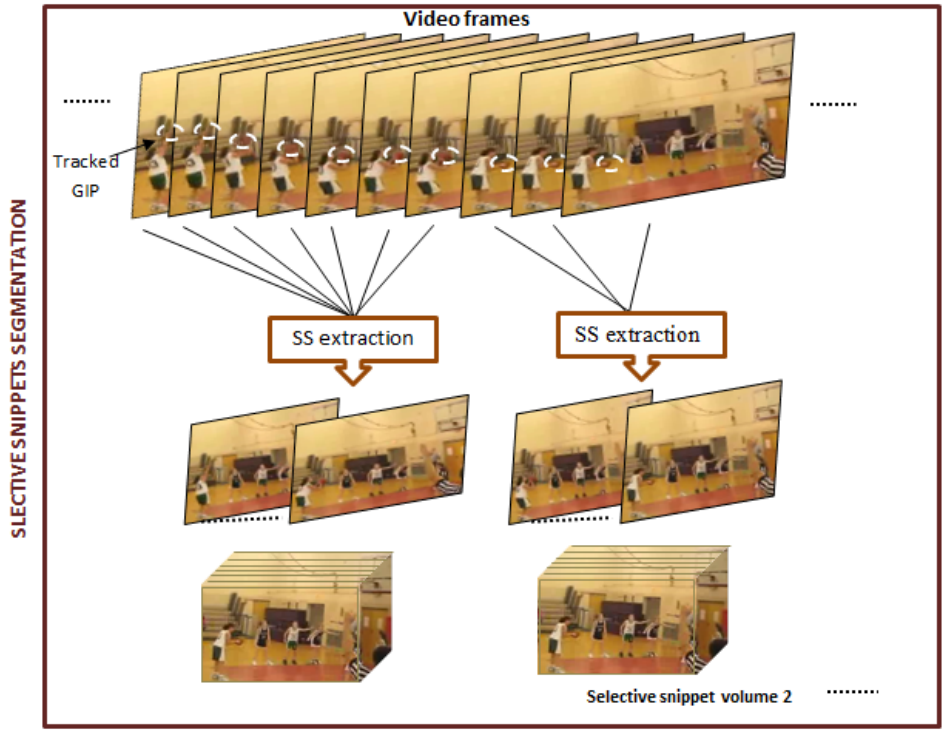
4.2 Proposed architecture for human action recognition

The main goal of our work is to develop an efficient framework to achieve accurate action recognition. In this section, we highlight the main parts of the proposed system. The overall proposal of our trajectory descriptor and the associated architecture are shown in Fig 4.1.

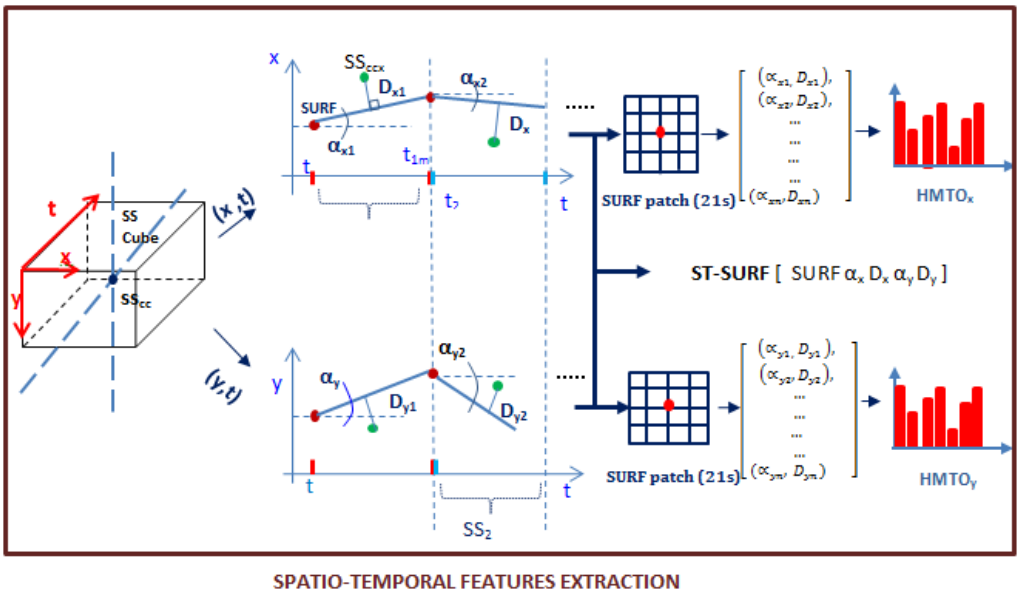
4.3 Trajectory based selective video segmentation

The underlying assumption behind a selective segmentation is that "Human action is accurate in a specific period of time and in a specific spatial position". The purpose of this work is to extract interest point of only moving objects/persons and then track them until the end of an action or a part of an action coined actionlet. Thus dense SURF are extracted to exploit the maximum spatial information in video frames [126]. Furthermore, we begin by a fine and dense SURFs detection in the first detected moving targets of the first frame. A group of SURF that covers significant moving human/objects parts is then defined. In this paper, we empirically set $G - SURF = 45$. Figure 4.2 illustrates the most relevant steps of the actionlets extraction and segmentation process.

The Trajectory based SS method introduced has many advantages. It not only allows to extract a SS covering significant Human/object actionlet. But also, it allows to track linear vectors of displacement to avoid additional trajectory shape feature computation.



(a)



(b)

Figure 4.1. The proposed framework. (a) discriminative segmentation process. SURF descriptors are densely extracted and a tracking process of the displacement of every group of interest points leads to selective snippets extraction. (b) every SS is considered a cubic volume. In this 3D volume SURF and their corresponding optical flow fields are extracted. Then $HMTO_x$, $HMTO_y$ are computed for every patch surrounding the selected SURF.

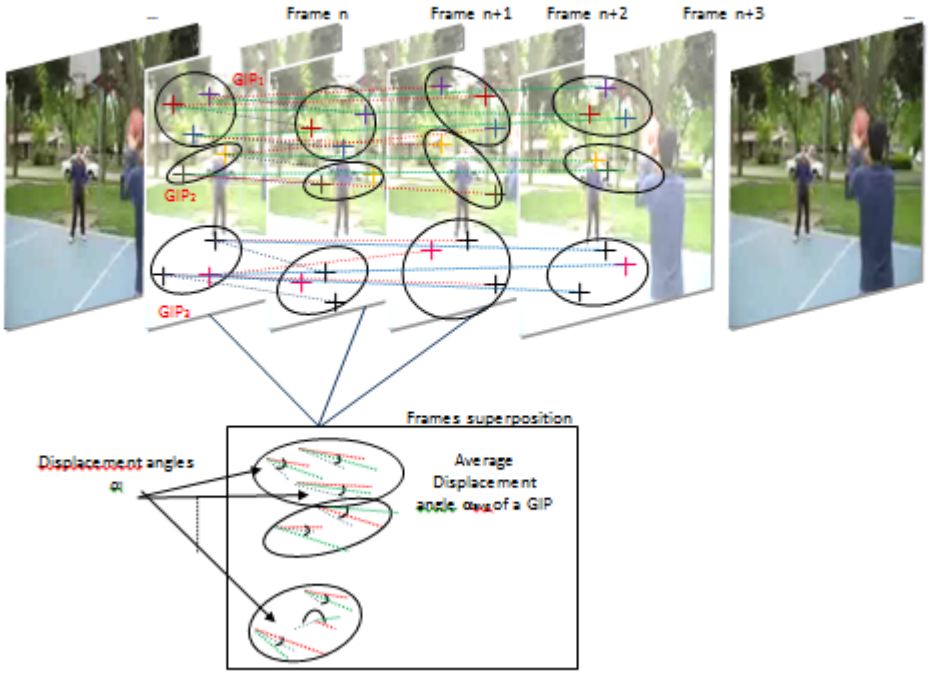


Figure 4.2. Actionlets extraction by selective segmentation. The densely detected SURF are divided into groups of 45 SURFs. Every Group trajectory is tracked. When significant motion is detected a SS is extracted.

4.3.1 Selective snippets (SS) and Group of SURF (G-SURF) segmentation

SSS One of the main objectives in the proposed method is to reduce computational load. This could be achieved by reducing the number of the video frames to be analyzed. For this purpose, we propose the use of concepts of selective snippets and the group of SURF (G-SURF) . Considering three successive frames $(n, n+1, n+2)$, a detected SURF in the frame $n-1$ can be detected in the same location in the following frame n. But it can simply disappears or can be detected in another spatial location. The first and second cases are not addressed in this work. Because, in the first case , no motion is detected . In the second case , the SURF can no longer be followed. In the third case the SURF is moved. This allows to determine a trajectory description to follow the motion of this point. Considering that α is the angle between the lines segments supporting the motion of a SURF from the couple of frames $(n, n+1)$ and $(n+1, n+2)$ (see Figure 4.3). We

compare α to α_{max} and α_{min} (α_{max} is a threshold empirically set) to segment a succession of frames (SS) in which each SURF has an α lower than α_{max} and grater than α_{min}. Let $D_{n,n+1}$ be the displacement vector of a given SURF from the frame (n) to the frame $(n+1)$. $D_{n,n+2}$ from the frame (n) to the frame $(n+2)$ (see Figure 4.3).

$$D_{n,n+1} = (Dx_{n,n+1}, Dy_{n,n+1}, Dt_{n,n+1}) \tag{4.1}$$

and

$$D_{n+1,n+2} = (Dx_{n+1,n+2}, Dy_{n+1,n+2}, Dt_{n+1,n+2}) \tag{4.2}$$

$$\alpha = \arccos \frac{D_{n,n+1} \cdot D_{n+1,n+2}}{\|D_{n,n+1}\| \times \|D_{n,n+2}\|} \tag{4.3}$$

Note that, within a SS, all SURF motions are less than α_{max}.

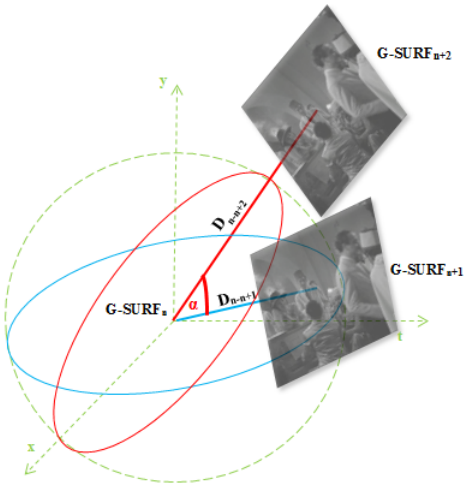


Figure 4.3. Displacement vectors between consecutive frames from KTH dataset

In order to avoid an oversized SS, we introduce, the concept of G-SURF. This is a parameter defining the number of grouped SURF empirically tuned. The grouping technique is then performed over successive detected SURF in a reference frame. By defining G-SURF, an average motion angle (α_{avg}) is computed and compared to α_{max}. The more SURF number is, the less the α_{avg} is sensitive to motion and the more the SS will have extended borders. The main steps of the proposed segmentation algorithm are given below (table **??**).

Table 4.1. Proposed algorithm.

Input : I - input video;
α_{min}, α_{max} - motion angles;
Algorithm :
step1 IP extraction from frames $\{f_1, f_2\}$;
step2 Groups of IPs defined;
step3 Compute the line supporting the motion;
Apply the above three steps to $\{f_2, f_3\}$;
Compute the angle between each motion line;
Extract α_{avg} for each GIP;
if $\alpha_{avg} \leq \alpha_{min}$;
then go to the next frame;
else Compare α_{avg} to α_{max};
end if
repeat previous steps;
until $\alpha_{avg} \geq \alpha_{max}$;
Output = f_n, t_{min}, t_{max} ;

4.4 Descriptor extraction

Action recognition is a very challenging computer vision task. As mentioned in the introduction, several descriptors have been proposed to achieve high quality action detection. In this section, we describe in details the main stages of the used descriptors. We also introduce a novel descriptor based on IPs trajectories to track interest points motions.

4.4.1 Motion trajectory extraction

The motion trajectory detection tracking and extraction are based on the following steps.

4.4.1.1 Optical flow extraction

Features tracking is performed by estimating optical flow. To increase optical flow estimation accuracy, several methods derived from the Horn and Schunck (HS) Optical flow formulation [106] have been proposed. In this thesis we employ Sun et al. [106] proposed algorithm. It approximates an optimized computationally tractable objective function, based on the original HS formulation. It is filtered using a bilateral weight that depends on the spatial and the color value distance of the pixels as done in bilateral filer. The initially computed optical flow serves in many blocks in the proposed framework. This

reduces feature extraction computational time. The complexity results will be shown and discussed in section 4.6.

4.4.1.2 Trajectory tracking

To every selective snippet corresponds a volume of frames in the 3D space called SS Volume (SS_v). This cubic volume is characterized by:

- The frame number(FN) varying from 1 to t_{max}.
- the frame surfaces dimensions (FS) varying from x to x_{max} in the x direction, and from y to y_{max} in the y direction.
- The SS cubic volume center (SS_cc) coordinates.

A given interest point $IP = (x, y, t)$ is defined by its spatial position (x, y) and its temporal cue t. In frame $(t+n)$, the IP undergoes a displacement u in the x direction, and v in the y direction and defined as, $IP(t+n) = (x+u, y+v, t+n)$. In all our experiments, unless mentioned otherwise, we consider only moving interest points when $u \neq 0, v \neq 0$. In every pre-defined SS_v, the 3D direction (u, v, n) is the direction of the IP motion. The motion vector is calculated by the Sun et al. [106] optical flow approach. Our main contributions consist on the use of motion trajectory orientation to describe IP displacement, instead of using directly the optical flow fields (u, v, n) and also to adapt the extracted features to the frame number of every SS. In-fact, the motion vector in the 3D space can be found by the intersection of two orthogonal planes to the plane (t, x) and the plane (t, y). To extract IP motion trajectory orientation, we project its motion vectors onto the planes (t, x) and (t, y) of the SS_v to define an angle for each projection of the first angle α_x between optical flow and the plane (t, x), the angle α_y between the plane (t, y) and the motion vector.

$$\propto_x = 90 - \frac{180}{\Pi}\arctan(\frac{u}{n}), \propto_y = 90 - \frac{180}{\Pi}\arctan(\frac{v}{n}). \tag{4.4}$$

The projection of each $SURF's$ motion vector on the planes (t, x) and (t, y) yields two lines L_x and L_y. The orthogonal projection of SS_{ccx} and SS_{ccy} onto the lines L_x and L_y allows computing the two distances D_x and D_y between the SS_v center and the lines supporting the motion vectors (L_x and L_y).
For an IP located at (x, y, t), the distances D_x and D_y are given by:

$$D_x = D_{xu} - D_{tv}, D_y = D_{yv} - D_{tu} \tag{4.5}$$

where

$$D_{xu} = (x - x_{max}/2)cos(180/\Pi arctan(\frac{u}{n})) \tag{4.6}$$

$$D_{tv} = (t - t_{max}/2)sin(180/\Pi arctan(\frac{v}{n})) \tag{4.7}$$

$$D_{yv} = (y - y_{max}/2)cos(180/\Pi arctan(\frac{v}{n})) \tag{4.8}$$

$$D_{tu} = (t - t_{max}/2)sin(180/\Pi arctan(\frac{u}{n})) \tag{4.9}$$

where t_{max}, x_{max} and y_{max} are the dimensions of the SS volume with t_{max} depending on the number of the frames contained within a segmented (SS_v). In the following, D_x and D_y describe the motion trajectory location in the 3D volume generated from the successive frames. Figure 3.14, is a graphical illustration of the cube center and its projection into the planes (t, x) and (t, y).

4.4.1.3 Histogram of motion trajectory orientation (HMTO)

A wide range of histograms have been proposed in the literature for action recognition description. Some of them focus on extracting motion cues such as [39] or (MBH) [81]. While other extract spatial information i.e., (HOG) descriptor [45]. In this paper, we introduce a novel descriptor called motion trajectory orientation histogram (HMTO). The most valuable property of this descriptor is that it is splitted in order to captures motion trajectory orientation patterns in both (x,t) and (y,t) directions. To gain more accuracy, we extract both $HMTO_x$ and $HMTO_y$ from a SURF centered patch. The patch is a square region with size $20s$ where s represent the current scale. Furthermore, for every pixel in the detected patch, we compute the optical flow. Then, we extract the direction parameters α_x and α_y. These are considered as the angular votes in $HMTO_x$ and $HMTO_y$. To use the trajectory cues to track actions, we propose to bin them based on the absolute motion distance. Finally we extract 8 bins histogram HTO_x and HTO_y. These histograms are finally L_2 normalized (see Figure 4.4).

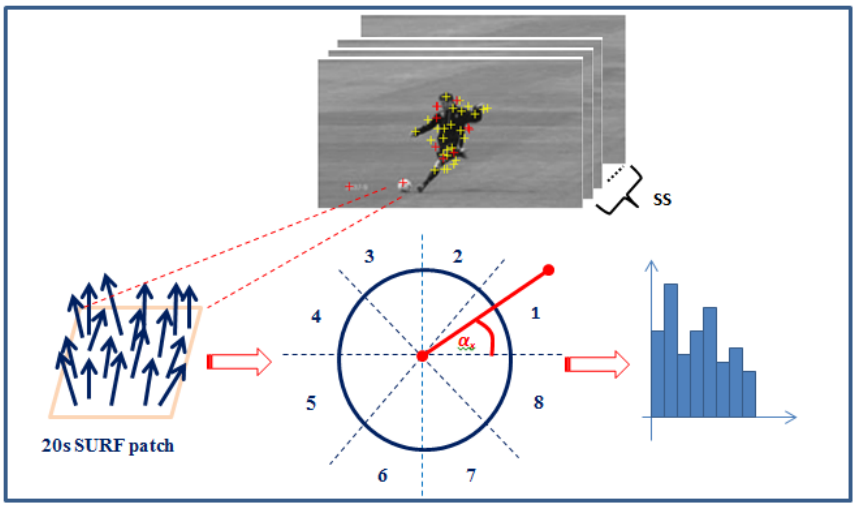


Figure 4.4. An overview of HMTO extraction.

4.4.1.4 Motion boundary histogram (MBH)

The motion boundary histogram (MBH) was introduced in [81] to detect action. MBH contains the distribution of the gradient of the optical flow fields in both x and in y directions. Hence, it captures salient optical flow changes while suppressing static motion usually derived from camera motion. The final MBH_x and MBH_y are 96D ($2 \times 2 \times 3 \times 8$) features set. In this work, we used MBH, not only for its aptitude of reducing camera motion, but also as a motion descriptor for its action recognition discriminative power attested in the state-of-the-art [81, 33].

4.4.1.5 Spatio-temporal SURF (ST-SURF)

ST-SURF was introduced by [43]. The main idea is to detect the trajectory of a SURF point by tracking its motion trajectory. The authors used Hessian Matrix to detect salient points. Then, they extract all The SURF in a given video. Finally they compute a 68D spatio-temporal SURF called ST-SURF. The results given by their proposed approach are encouraging but still below the state-of-the-art. In this thesis, we give an optimized ST-SURF extracted over a SS. This step is based on a dense SURF extraction, which boosts the information detection step. We combined ST-SURF with other descriptors to capture maximum spatial and temporal cues. We choose ST-SURF for many reasons. First, it contains spatial information driven by the SURF and temporal information driven by the

optical flow, the size of this descriptor and finally it provides localization information. The latter will add spatial information to the bag of words encoding step.

4.5 Experimental Setup

We conduct the proposed action recognition framework on three human action classification datasets, i.e., KTH, UCF sports, UCF11 (YouTube Action Data Set) (see Figure 4.5).

Experiments are carried on video characterized by various contexts, duration, viewpoints, occlusion, illuminations, actors, controlled ones and realistic others. Our experiments are carried on a total of 26 classes and 3759 video. To engage a fair comparison, we follows the settings provided by some methods of the state-of-the-art. The evaluation is given by the average accuracy result. In the following, we provide a brief overview of every dataset in the section 4.5.1, then the parameters setting for every block in the action recognition system are drawn in section 4.5.2.

4.5.1 Datasets

The proposed framework is validated on the KTH dataset [34], UCF sports dataset [102] and UCF11 [100]. Figure 4.5, depict major actions of these datasets.

4.5.1.1 KTH

The KTH is a controlled, commonly used public benchmark test dataset for human action recognition [13]. This dataset contains 6 actions classes (walking, running, jogging, boxing, hand waving, hand clapping). The actions are performed by 25 persons in 4 different scenarios (indoors, outdoor, different clothes outdoors, scale outdoors), with a total of 2391 video samples. The average length of videos in the KTH dataset is about 20 second long with homogenous and static background.

4.5.1.2 UCF sports

UCF sports dataset is a realistic and challenging data obtained from broadcast sport videos by Ahmed et al. [102]. The collection represents a natural pool of actions featured in a wide range of scenes and viewpoints. The publicly available part of this dataset contains nine actions namely Diving (16 videos), Golf swinging (25 videos), Kicking (25 videos), Lifting (15 videos), Horseback riding (14 videos), Running (15 videos), Skating

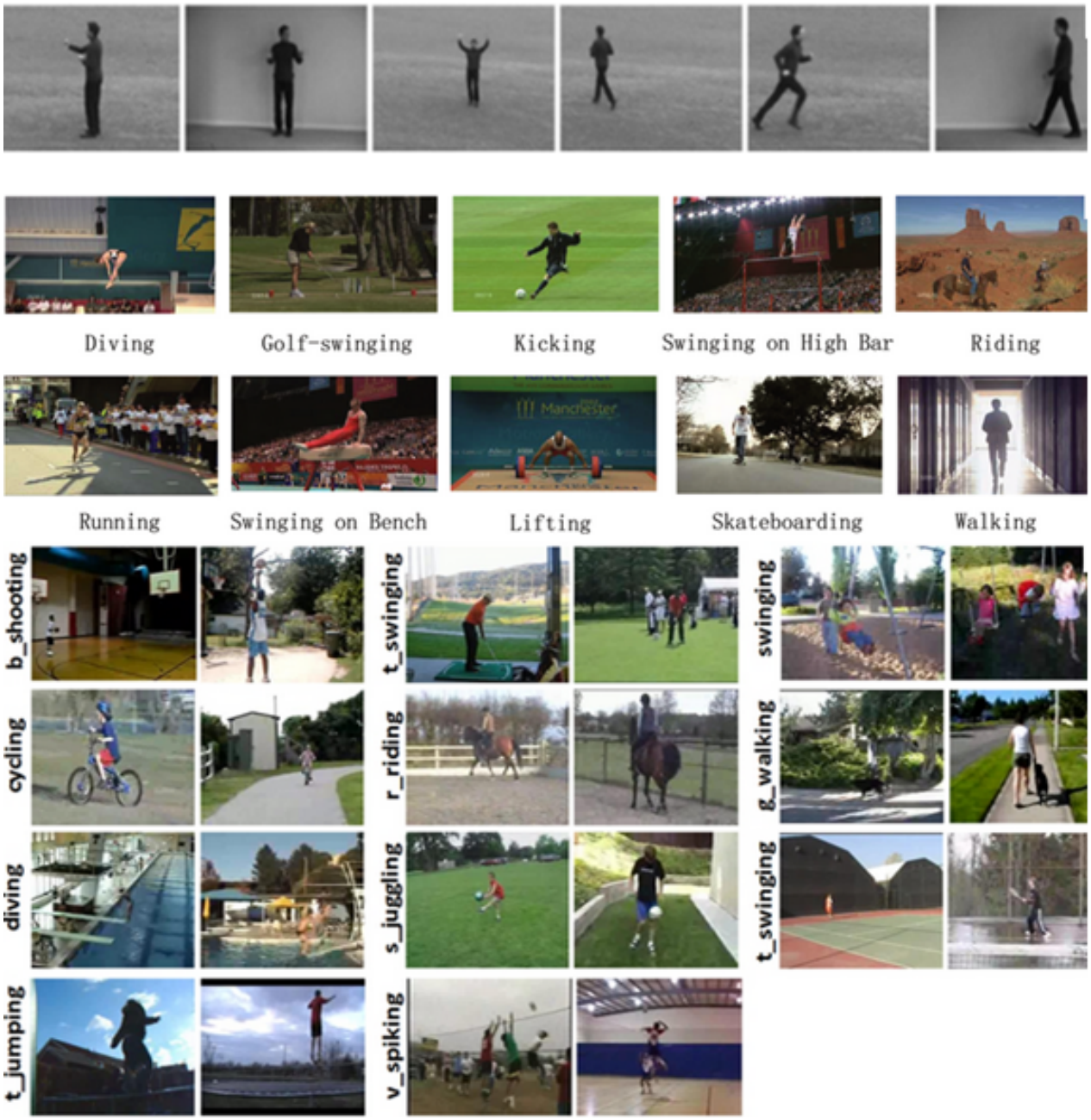


Figure 4.5. Selected frames from the evaluated benchmarks. KTH: 6 actions, UCF sport: 9 actions and YouTube: 11 actions.

(15 videos), Swinging (35 videos), Walking (22 videos) . This dataset contains close to 200 video sequences at a resolution of 720x480 [102].

4.5.1.3 UCF11 YouTube Action Data Set

UCF11 is the newest update of the YouTube action dataset, it contains 1168 videos and 11 action classes called basketball shooting, biking/cycling, diving, golf swinging, horse back riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog. For each class, the videos are grouped into 25 sub-groups with about 4 to 9 action clips in it. The video clips in the same sub-group share few common cues, such as the same actor, similar appearance, same background, similar viewpoint, etc. This data set is very challenging due to large intra-class variations, differences in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, illumination conditions, etc.

4.5.2 Extracted features

The extracted features in the proposed framework are SURF descriptor used as an appearance features. The motion trajectory orientation cues of the SURF are tracked using two approaches. The first consists in splitting optical flow fields, then an original projection of optical flow cues into the planes (x, t) and (y, t) leads to extract motion evolution through time. Followed by the extraction of the spati-temporal location of the trajectory in a 3D volume. As described before, The extracted ST-SURF is 68D vector (64D SURF, α_x, D_x, α_y, D_y). The second approach is based on the extraction of a square shape patches surrounding the detected SURF. The size of the detected patch is 20s. For every detected patch a novel histogram of motion trajectory orientation is computed in both planes (x, t) and (y, t). We Kept the same parameters settings used to design HOG and MBH. The extracted $HMTO_x$ and $HMTO_y$ are both 96D dimension. To reenforce our action recognition system we used motion boundary histogram MBH as a motion descriptor and also for its ability of removing camera motion. MBH_x and MBH_y are 96D histograms. We finally extract three descriptors ST-SURF (68D), $HMTO_x$ and $HMTO_y$ (96 + 96=192), MBH_x and MBH_y (192).

4.5.3 Features encoding: Bag of features

The classification step starts k-mean clustering applied on a set of 10^6 randomly selected features, to build a visual dictionary for every extracted descriptor type (ST-SURF, $HMTO_x$, $HMTO_y$, MBH_x, MBH_y). For every descriptor we construct 4000 visual words. The k-mean clustering is initialized 8 times and we kept the configuration with the lowest

error rate. The extracted histograms are L_2 normalized to ensure better visual quality. Finally, to classify the actions we use a non linear SVM with an RBF_{χ}^2 Kernels [39].

$$K(v_i, v_j) = \exp(-\sum \frac{1}{A^c} D(v_i^c, v_j^c)), \tag{4.10}$$

Where $D(v_i^c, v_j^c)$ is the χ^2 distance between video v_i and v_j of the channel c. A^c is the mean distance value of the training features.

4.6 Experimental results and discussion

In this section we report and discuss the action recognition results extracted from KTH, UCF sport and UCF11 datasets. The purpose of this discussion is to highlight the keys success and weaknesses of the proposed action recognition system. As described below, we use the same settings and evaluation metrics of the state-of-the-art.

4.6.1 Evaluation of the proposed approach

This section presents the results of a standard evaluation of the proposed system based on a selective snippet segmentation with $\alpha_{max} = 40$ in KTH dataset and $\alpha_{max} = 25$ for realistic datasets . The extracted features are then based on this segmentation extents. The results are reported for three datasets ie., a controlled dataset (KTH) and two realistic ones (UCF sport and YouTube). The evaluation results are given by the confusion matrices (see Tables 4.2, 4.3, 4.4). The overall performances on KTH, UCF sport and YouTube dataset are 94.9%, 90.3% and 90.44%.

With the KTH dataset, we follow the same protocols used in the methods of the state-of-the-art for learning and testing phases. A group of 24 actors is involved during the learning phase. One actor, for every action, is left for the test step. Our proposed approach achieved an accuracy rate of 94.9% outperforming various proposed approaches (see Table 4.2). KTH video sequences were captured using a static camera with 25 fps frame rate, with a spatial resolution of 160x120 pixels and a length of four seconds in average. The experiments are performed using an Intel core i5 computer. The code is paralellized and the number of cores was automatically selected by matlab software.

Figure 4.7, represents the computational complexity reported for the mean computing time for one video per subject, a total of 150 videos (25 actors- 6 actions), and with respect to the frame number into a SS. The selective snippets segmentation consumes only 2% of the whole processing time which highlight the efficiency of this segmentation technique.

The optical flow has the largest part of the processing time 48%. Since we reuse the pre-computed optical flow and the SURF descriptors, we save processing time for HMTO descriptor only 11%. The extraction time for descriptor, the training and the testing time are relatively not consuming, and about 10% are less than the optical flow extraction duration.

	boxing	handclap	handwave	jogging	walking	running
boxing	92.1	7.9	0	0	0	0
handclap	0	90.5	9.5	0	0	0
handwave	0	0	97.2	2.8	0	0
jogging	0	0	0	100	0	0
walking	0	0	0	0	96.3	3.7
running	0	0	0	0	6.3	93.7

Figure 4.6. Confusion matrix of the classification results for the KTH dataset for the proposed approach using the combination of HMTO, PCA-STSURF, MBH descriptors.

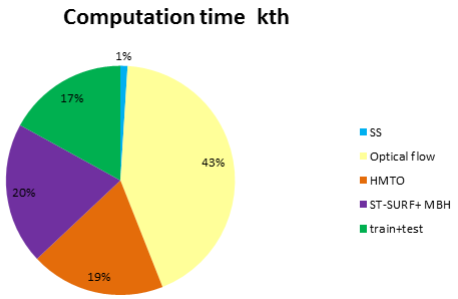


Figure 4.7. computationlal requirement for 150 videos in KTH dataset.

The results shown in Table 4.7, are reported from the original papers [34, 38, 39, 35, 21, 13, 43, 33, 127]. The performances are around 94.9% in KTH dataset, nearly 0.4% better than Spatio-temporal SURF [13]. We achieved 6-7 % improvement more than the ST-SURF [43]. This is due to many factors, such as the optimization of the selective segmentation based on dense SURF and the fusion of the ST-SURf with trajectory descriptors

(HMTO, MBH). It can be also noticed that MBH descriptor does not improve significantly the performances because KTH is a controlled dataset with minimum camera motion.

Table 4.2. Some state of the art recognition results over the KTH dataset.

Method	Year	Accuracy (%)
Shuldt et al. [34]	2004	71.7
Jhuang et al. [38]	2007	90.5
Laptev et al. [39]	2008	91.8
Niebles et al. [35]	2008	93.3
Lin et al. [21]	2009	95.8
Noguchi et al.[13]	2012	94.5
Megrhi et al. [43]	2013	88.2
Wang et al.[33]	2013	95.3
Virigkas et al. [127](CTW)	2014	93.8
HMTO+MBH+ST-SURF	2014	**94.9**

The UCF sport dataset is an uncontrolled video of sportive activities. The videos are captured from different camera under various conditions. The resolution is 720×480 pixels. For SS, we use the same settings as KHT dataset i.e. $\alpha_{max} = 25°$. The results of our approach are depicted in Figure 4.8. The proposed scheme achieves an accuracy of 90.3%, which is better than the state-of-the-art methods [102, 128, 23, 129, 43, 33, 127]. The increase in accuracy over the other methods is within the range $[0.3 - 4\%]$. This is due to many reasons related to our strategy. In fact, the selective segmentation allows to extract actionlets rather than randomly set a descriptor length. Note that the proposed approach gets better results on realistic video containing various motion from several sources. This is well handled by our approach, because in the SS, the segmentation ignores small angles and performs really well for detecting rotational motion. Second, the use of the MBH descriptor allows to capture significant motion while suppressing small ones.

Regarding the computational timing, we investigate running the proposed approach on a total of 90 videos (10 videos for each class randomly chosen). The SS step took less than 1% of the total processing time. While the biggest consumer is the optical flow extraction which represents 46% of the whole process. HMTO extraction process takes 18%, since it requires a dense optical flow extraction step. It is 2% less than both ST-SURF+MBH extraction step. This is explained by the reuse of the precomputed optical flow. Figure 4.9summarizes the obtained results.

	Diving	Golf	Kick	Lifting	Riding	Running	Skate	Swing	Walk
Diving	98.7	0.8	0	0	0	0	0	0.5	0
Golf	0	92.3	7.7	0	0	0	0	0	0
Kick	0	0	99.7	0.3	0	0	0	0	0
Lifting	10.5	0	0	89.5	0	0	0	0	0
Riding	0	0	4.2	0	95.8	0	0	0	0
Running	0	0	0	0	4.6	87.2	0	0	8.2
Skate	7.5	0	0	0	0	4.23	88.27	0	0
Swing	9.7	0	0	5.5	0	0	0	84.8	0
Walk	0	0	0	0	11	9.5	0	0	79.5

Figure 4.8. Confusion matrix of the classification results for the UCF sport dataset for the proposed approach using the combination of HMTO, ST-SURF, MBH descriptors.

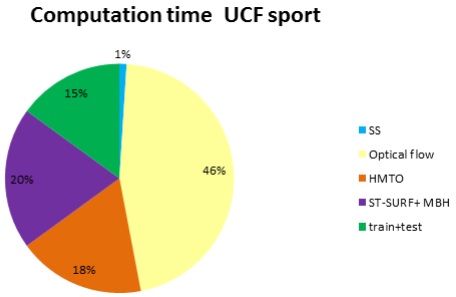


Figure 4.9. Computational requirement for UCF sport dataset.

Different background, view points, illumination, the existence of difference in scale, in actors, in appearances and poses and the existence of camera motion are the key-challenge of YouTube dataset. The latter is the more realistic benchmark we used in our experiments. The confusion matrix shown in Figure 4.10, reveals a performance of 90.44% outperforming the results published in the literature. We obtain an increase of performance around $[0.2 - 4\%]$ greater than the overall reported results and outperforming the lasted results [127] by 0.5%, (Table 4.4).

In order to evaluate the complexity, we conducted our experiments on 245 video (1 video from every groups in every action). According to Figure 4.11. the complexity is in

Table 4.3. Some state of the art recognition results over the UCF sport dataset.

Method	Year	Accuracy (%)
Rodriguez et al. [102]	2008	69.2
Kovaska and Grauman [128]	2010	87.3
Wang et al. [23]	2011	85.6
Le et al. [129]	2011	86.5
Megrhi et al. [43]	2013	88.2
Wang et al.[33]	2013	80.7
Virigkas et al (CTW)[127].	2014	90.1
HMTO+MBH+PCA-STSURF	2014	90.3

	shoot	bike	dive	golf	Riding	juggle	swing	tennis	jump	spike	w.dog
shoot	89.7	0.8	0	0	0	0	0	0.5	0	0	0
bike	0	99.7	7.7	0	0	0	0	0	0	0	0
dive	0	0	92.1	0.3	0	0	0	0	0	0	0
golf	10.5	0	0	88.5	0	0	0	0	0	0	0
Riding	0	0	0	0	89.3	0	0	0	0	0	0
juggle	0	0	0	0	4.6	86.5	0	0	8.2	0	0
swing	7.5	0	0	0	0	4.23	89.1	0	0	0	0
tennis	9.7	0	0	5.5	0	0	0	93.5	0	0	0
jump	0	0	0	0	11	9.5	0	0	95.4	0	
spike	0	0	0	0	11	9.5	0	0	79.5	92.4	0
w.dog	0	0	0	0	11	9.5	0	0	79.5	0	91.4

Figure 4.10. Confusion matrix of the classification results for the YouTube dataset for the proposed approach using the combination of HMTO, ST-SURF and MBH descriptors.

the same spirit of the previously reported results. In-fact, SS consumed 2% of the total time. These results prove the rapidity of the segmentation process even when facing realistic videos. The most expensive part is taken by the optical flow 55% about 9% more than in UCF sport experiment. This is the consequence of the complexity and the variable quality of the dataset and the density of details in a realistic benchmark. We note that the optical flow used in this work [106] is based on a pre-filtering step to optimize the computation of optical flow. HTMO computation takes about 21% as the optical flow is densely extracted from SURF based patches and contains many extra details and contexts in relation with the action.

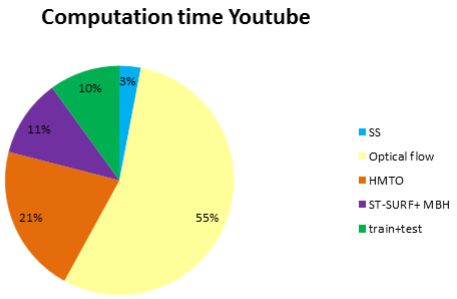


Figure 4.11. Computational requirement for 245 videos from YouTube dataset.

The proposed approach improves results on realistic datasets UCF sport and YouTube more than in the case of KTH dataset. In the latter our results are equivalent to those obtained with the state-of-the-art methods. The injection of the MBH is not really efficient in KTH since it has a static background. We achieved a slight improvement of 0.1% on UCF sport dataset. The best improvement of 0.5% is realized on YouTube dataset. These results confirm that the selective segmentation we proposed in this paper, (efficient more in rotation since it is based on an angular motion), contributes significantly in this improvement. It captures motion of several actors or objects in relation with the actors while ignoring small motion since we set a threshold on motion magnitude. We highlight also the use of MBH descriptor, associated with the proposed MHTO, strike a balance between motion description and camera motion reduction. Moreover, the reported results are based on settings that might be different in the code-book generation and the learning testing step.

Table 4.4. Some state of the art recognition results over the UCF YouTube dataset.

Method	Year	Accuracy (%)
Liu et al. [100]	2009	71.2
Ikizler-Cinbis and Sclaroff [130]	2010	75.2
Wang et al. [68]	2011	84.2
Le et al. [129]	2011	75.8
Wang et al.[33]	2013	85.4
Virigkas et al. (CTW) [127]	2014	90.1
HMTO+MBH+PCA-STSURF	2014	90.44

4.6.2 Comparison with other descriptors

In this section we compare the proposed features with a selection of descriptors from the literature. In [33], authors considered several approaches to evaluate their accuracy. We report some of their recent results for comparison. The reported results are given in the original papers [33, 127, 74]. To achieve fair comparison we conduct experiment using ST-SURF on YouTube dataset. We also fused MBH_x and MBH_y to extract MBH evaluation. $HMTO_x$ and $HMTO_y$ are also fused to allow evaluating the HMTO. Several observations could be drawn from the performances evolution depicted in Figure 4.12.

The distribution of the trajectory angles given by HMTO perform well in KTH dataset 90.1% outperforming dense trajectory 89.8%, KLT trajectory 89.4% and SIFT trajectory 44.6%. As HMTO allows the tracking of the trajectory of a moving patch, the temporal extents of the action are settled by selective segmentation into actionlets. Notice also that, MIEF32 reported 96.76%, this is due to the use of an optimized SVM learning approach. AMAR-CTW achieved 93.8% when they cluster motion curves using GMM in both learning/test steps.

In the case of realistic datasets UCF sport and YouTube, the performance of HMTO decreases to 80.3% and 76.4%. Hence, the more is the dataset realistic and contains spotty background the less is the performance. However, we still outperform the trajectory based descriptors ie. dense trajectory 75.4% and 67.5%, KLT trajectory 72.8% and 58.2% and SIFT trajectory 55.7% and 47.3%.

We can also observe that the performances of the proposed HMTO (90.1%, 80.3%, 76.4%) are better than MBH (90.0%, 79.6%, 74.8%) in the three datasets. This demonstrates the importance of the motion cues in detecting human actions. The results are all better in the case of KTH as it does not contain significant camera motion.

It could be also noticed that combined with ST-SURF and MBH, the HMTO gives best results in realistic and complicated video. This encourages the use of different features to achieve relevant action recognition. The results are improved by 4.9% on KTH dataset and, between 10-20% for realistic video. This is the consequence of the efficiency of the association of the actionlets extraction with MBH features to reduce video in realistic benchmarks. When combined with HOG, HOF and MBH, dense trajectory, KLT trajectory and SIFT trajectory shows a significant improvement. As the final descriptor globes spatial information from HOG, temporal information from trajectory and from HOF and the use of MBH which detects motion and suppresses small linear displacement. The use of different learning approaches influence significantly the final performances, this is shown

by MIL-F32 and TMAR-CTW on KTH dataset.

		Year	Dataset		
			KTH	UCF sport	Youtube
Proposed approach	HMTO	2014	90.1 %	80.3 %	76.4 %
	ST-SURF	2014	88.2 %	80.7 %	79.5 %
	MBH	2014	90.0 %	79.6 %	74.8 %
	Fused	2014	**94.9** %	**90.3** %	**90.44** %
D.T	Trajectory	2009	89.8 %	75.4 %	76.5 %
	Combined	2010	94.2 %	88.0 %	84.1 %
KLT	Trajectory	2009	89.4 %	72.8 %	58.2 %
	Combined	2010	93.4 %	82.1 %	79.5 %
SIFT	Trajectory	2009	44.6 %	55.7 %	47.3 %
	Combined	2010	84.9 %	77.9 %	73.2 %
ST-SV	Trajectory	2009	96.76 %	- %	84.52 ± 5.27 %
LCSS	Trajectory	2009	93.8 %	90.1 %	90.1 %

Figure 4.12. Various reported results of descriptor performances in action recognition.

4.6.3 Evaluation of the settings

In this section we evaluate some changes in both the segmentation and feature extraction processes. The first proposed experiment is based on controlling the selective snippets extents. The latter has a great impact on the action recognition results. In-fact, SS defines the actionlet content. The latter, combined orderly, describes human actions. The trajectory length depends on the motion extracted angles and the G-SURF. We apply the experiment on UCF sport dataset as a realistic benchmark and KTH dataset as a controlled one.

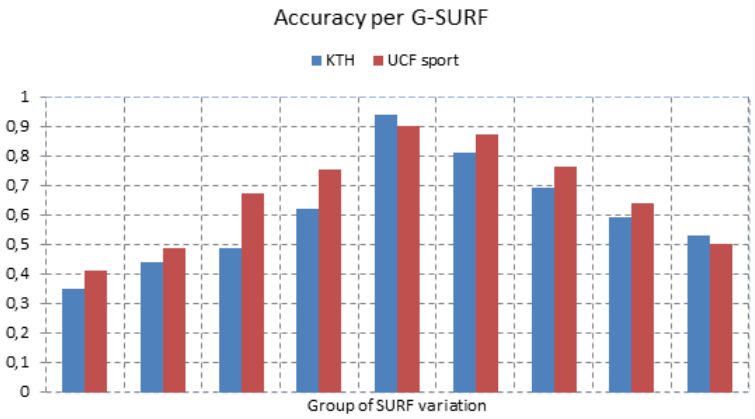


Figure 4.13. G-SURf variation.

Figure 4.13, reports the impact of the variation of the G-SURF. We can conclude that for a G-SURF between 35 to about 45 SURF, an improvement in performances is achieved with our approach. In-fact, below 35 SURF, the SS is not accurate since it captures very small motion. Greater than 45 SURF, the detected motion may ignore significant human action and results in a very long SS.

The second experiment reports the impact of the variation of the motion threshold α_{max} on the recognition of every action in KTH dataset. The results are depicted in Figure 4.14. The performance of the proposed approach decreases significantly with $\alpha_{max} = \pi/18$ and $\alpha_{max} = \pi/4$. However in this controlled dataset, with static background, $\alpha_{max} = \pi/4$ gives slightly better results than $\alpha_{max} = \pi/18$ because the dataset does not contain realistic scenes, it is also based on single action without contexts connected to the action (like ball for a footballer, or a glass for a drinker). So, increasing α_{max} increases the chance of detecting the whole body of the person performing the action. While decreasing this threshold, will not contribute to detect more information.

In Figure 4.15, we selected the first 4 SS extents to demonstrate α_{max} variations impact on the SS in KTH dataset. The tested video contain 360 frames. Indeed, every line, in Figure 4.15, represents the SS borders. With $\alpha_{max} = \pi/18$, we extracted 194 SS, which is the equivalent of 2-4 frames per SS. This is a huge number because the detected motion is concentrated on the actor performing one, almost linear, action. $\alpha_{max} = \pi/4$ we extracted 6 SS. We can clearly remark that with a small angle in a "boxing" action context a very small angle will report a big number of SS reporting visually the same action (see Figure 4.15, (a), line one and two). This yields to a longer processing time for redundant information extraction. Using $\alpha_{max} = \pi/4$, we can see from Figure 4.15.(b), that the results are not

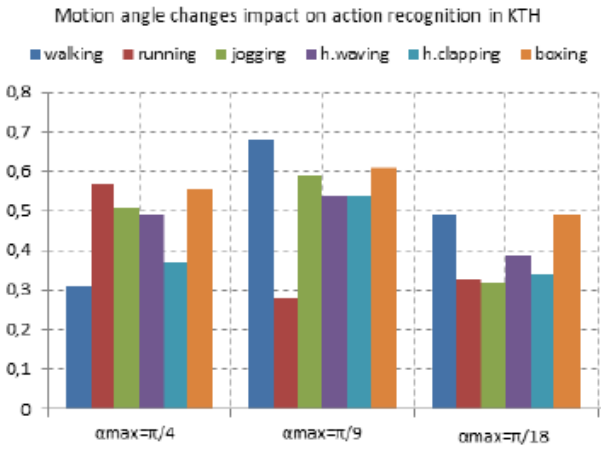
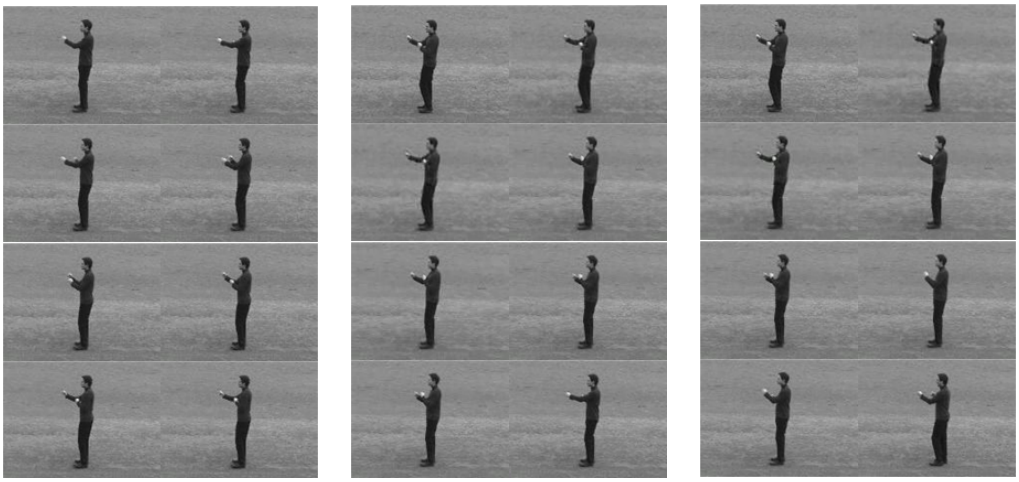


Figure 4.14. Motion angle variation on KTH

really different from $\alpha_{max} = \pi/9$, with less SS number. The emerged observation is that for non realistic benchmarks, we better use important thresholds to detect actions.

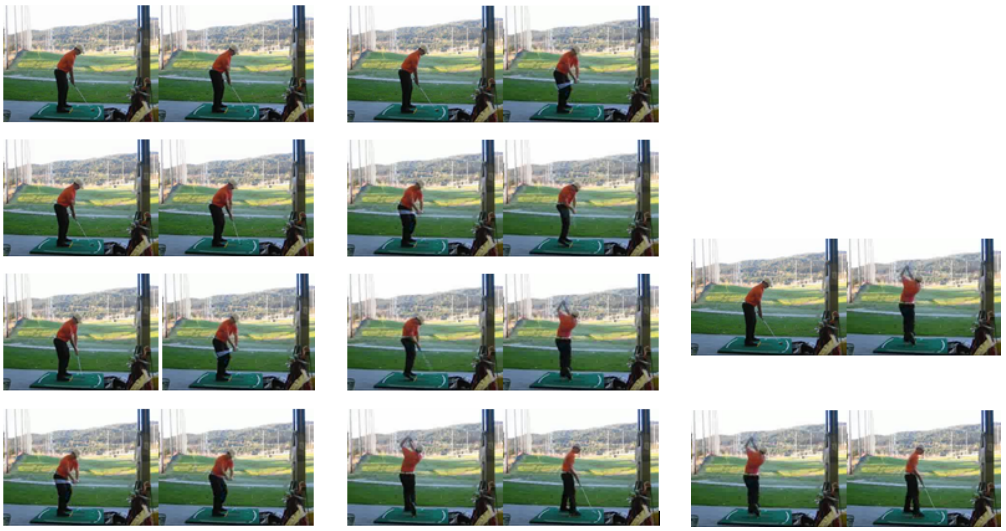
In the UCF sport Dataset evaluation experiment, the variation of the threshold has a great impact on the system performances. We used 171 frames video sequences. With significant background in a realistic context. The scene contains one person, performing "Golf" action. A realistic action contains an actor accompanied by several elements that are part and somehow in a relationship with the action (for instance a ball in the soccer kicking action, is a part of the action context). This case corresponds to a person, in a golf course, using a golf club. All the elements might contribute to detect the action. However, when hitting a draw the motion of the golf clubs is really fast. Surprisingly, the SS we extracted considers the golf club motion in addition to the human action. For $\alpha_{max} = \pi/18$, we extracted 38 SS, about 2-5 frames per SS. As we can see in Figure 4.16.(a), row 1, the SS contains the actionlet describing the small motion of the head of the actor. In row 2, we can see the small displacement of golf club despite the quality of the average resolution of the video and the non significant golf club motion.

With $\alpha_{max} = \pi/9$, the results are more convincing and confirm the excellent performances in term of tracking action. In-fact, in figure 4.16.(b), row 1, the extracted actionlet describes a significant motion. We detect the golf club motion despite the fastness of the action and the thinness of the the golf club. We extracted 19 SS, about 8-12 frames by SS. The actionlet perceptually describes a part of the action. We did not detect the ball motion, due to its size, color and the quality of the video and the fastness of its motion displacement after being hit. When using $\alpha_{max} = \pi/4$, two SS are extracted, which indeed,



(a) (b) (c)

Figure 4.15. Motion threshold changes in KTH dataset. (a) $\alpha_{max} = \pi/18$; (b) $\alpha_{max} = \pi/9$; (b) $\alpha_{max} = \pi/4$;



(a) (b) (c)

Figure 4.16. Motion threshold changes in UCF sport dataset. (a) $\alpha_{max} = \pi/18$; (b) $\alpha_{max} = \pi/9$; (b) $\alpha_{max} = \pi/4$;

gave bad results. We believe that, in realistic video, many elements can contribute in describing an action (ie. phone for call phone action). This is why the use of a small motion angle is desirable to extract relevant actionlets. We choose a threshold of $\alpha_{max} = 25°$ to detect small actions that are likely belonging to the action, but at the same time we need to avoid to capture small unnecessary movements.

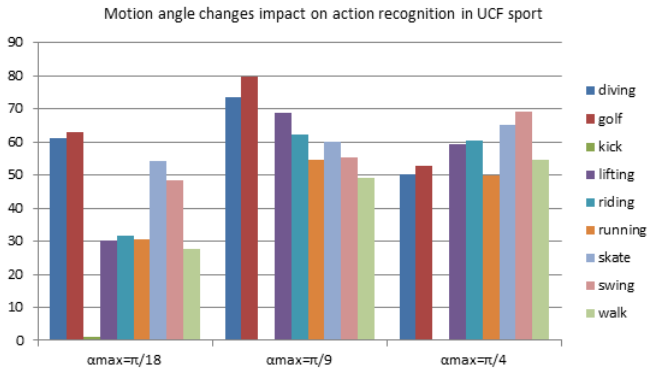


Figure 4.17. Motion angle variation on YouTube

The third experiment evaluates the resulting performance of the HMTO. In-fact, we changed the size of the neighborhood surrounding SURF form $0.5 \times s$ to $200 \times s$. The obtained results are shown in Figure 4.17.

4.7 Summary and conclusion

A novel technique for actionlets extraction from videos via a space-time selective video segmentation is introduced. The action recognition results in three challenging benchmarks. The obtained results prove the efficiency of the proposed approach. In fact adopting this method, there is no need to fix a trajectory length for all the extracted descriptors, or to set an equal sub-volume length for all the detected space-time descriptors. Rather, every significant motion is detected and compared to a threshold to evaluate the motion magnitude. Then every detected actionlet describes an action part that concatenated orderly with successive actionlets allows to construct a human action.
We developed an efficient descriptor called Histogram of motion trajectory orientation. The latter is based on the tracking along a trajectory of moving regions. The distribution of motion angles we extracted outperforms several proposed descriptors. We are convinced that our descriptor has many advantages. For instance, it extracts dense pre-filtered optical flow from local patches. The latter are based on moving SURF localization. Thus, we extract meaningful motion information while saving extracting dense optical flow from the whole frame. We inject localization cues into the proposed descriptor to avoid using extra computations to add spatial information ie. spatial pyramid. For every region of interest we extracted histogram on two channels based on the optical flow features. When we extracted the proposed descriptors in a selective snippet, we avoid computing trajec-

tory shape features and keep the motion angular trajectory cues. We qualitatively prove its sufficiency for action recognition.

One of the most important challenges human action recognition is the camera motion. To handle this issue we processed differently in the segmentation step and the descriptor extraction step. In-fact, in the SS process we set a threshold to avoid small linear motion, so it cannot influence significantly the segmentation quality. In the HMTO extraction process we first use a pre-filtering optical flow extraction to boost the accuracy of optical flow estimation. Then we coupled our descriptor with motion boundary histogram. The latter, is based on the gradient of optical, thus it suppresses small motion likely coming from video capture sources.

Despite we achieved encouraging results, there are several rooms of improvement. In the next chapter, we plan to focus on optimizing space-time selective segmentation for long actions. This is why we propose a more reliable and optimal action detection and segmentation. The proposed approaches are evaluated on some realistic big datasets.

CHAPTER 5

TRAJECTORY TRACKING FOR HUMAN ACTION DETECTION AND RECOGNITION

The whole is greater than the sum of its parts.
Metaphysica. Aristotle

Contents

5.1 Introduction

Currently recognizing human actions in videos is a challenging task. In fact, videos generally might contain complex actions with large intra-class variability, poor quality and camera motion. In this chapter, we focus on reducing camera motion effect in both action detection and video description tasks. The action detection allows to segment a video into a succession of patches containing significant human action. In the literature, videos are temporally segmented with different methods. Some, [131, 132, 57], are based on the trajectory of interest points. In the last decades, a wide variety of trajectory based descriptors has been proposed, [63, 58, 62, 61, 64]. Trajectories features can be extracted from optical flow [33, 43, 121, 92], or by matching the interest points in different frames [64, 33]. The number of frames involved in setting the trajectory length depends on the used approach. In [64], the trajectory length belongs to a fixed interval while in [68] it is chosen to be fix in order to extract a displacement vector.

Human motion segmentation acts as a pre-processing step for action recognition [133, 66]. Thus, the latter's performance is highly related to one of the motion segmentation algorithm. To assist action recognition, moving objects/humans in videos need to be first detected, then segmented. Pixel-wise techniques, namely background subtraction and temporal differencing [134], are the most straightforward methods for motion segmentation. However, they are only effective under the consideration of static cameras. When dealing with cameras in motion, these models are likely to fail as the background is continuously varying in addition to the target's motion. In the literature, many approaches considering camera motion are proposed.

A recent study [135], revealed that optical flow based methods [136, 137, 138] are one of the most employed techniques in motion segmentation. Horn and Schunck [139] and Lucas and Kanade (LK) [65] are the oldest yet most employed optical flow algorithms. Regarding their limitations toward accuracy and illumination changes, some improvements are presented. Our method is also based on optical flow computation. We apply the pyramidal implementation of the Lucas and Kanade algorithm [140] to estimate optical flows of the detected interest points (IP) in each frame. This method ensures detecting motion with different speeds.

In spatial domain, these interest points carry high information contents. The most employed interest points detectors are Scale Invariant Feature Transform (SIFT) [47] and SURF [24] descriptors. Jurie *et al.*[141] revealed that using a regular dense grid for sampling local image patches enhances the use of interest points. A recent evaluation of dense

sampling proposed by Uijlings et al. [126] proved that dense SIFT and dense SURF descriptors may be extracted more quickly with no loss of accuracy. Moreover, dense sampling has been shown to improve or produce comparable performance in different applications such as image classification [142][143]. Further, Wang *et al.* [23] evaluated the use of dense sampling at regular positions in space and time for action recognition

In practice, motion segmentation is quite difficult. The complexity of dynamic scenes is considered as the biggest challenge facing this task where objects' motion is combined with both the camera and background motions. In this case, camera motion compensation is crucial. Earlier approaches to camera motion compensation relied on estimating the camera motion as a 2D affine transform or homography [144, 78, 80]. Other methods performed motion compensation at trajectory level [79]. Uemura *et al.* [66] used a sophisticated and robust(RANSAC) estimation of camera motion. All these works support the potential of motion compensation. However, in some cases it is almost impossible to separate the foreground and the background when there are close up captures of the human activity.

We propose, in this chapter, a motion segmentation algorithm based on dense features which are comparable to state-of-the-arts. Optical flows of detected keypoints are then computed. We propose to compensate the camera motion by determining the camera flow direction using the k-Nearest Neighbor (KNN) clustering algorithm then applying the affine motion model. Finally, humans/objects are segmented using temporal differencing between two motion-compensated frames and a bounding box is drawn around each detected object. Thereafter, the discriminative video segmentation is performed based on the extracted bounding boxes (BB).

5.2 Action detection and motion segmentation

Motion is, with no doubt, the most reliable source of information for studying humans' behavior. Hence, human motion segmentation appears to be a good way to reduce the amount of data involved in the task of action recognition. Optical flow is one effective and commonly performed technique for motion segmentation [134]. This task gets more challenging when dealing with scenes captured by moving cameras. In this situation, the scene necessarily involves the background, the camera and/or humans and objects motion. At this level, camera motion compensation becomes compulsory. To do so, some attempts have been proposed. In [66], Uemure *et al.* combined color based image segmentation with dominant homographies estimated based on local extracted features. Cinbis *et al.* [130] applied video stabilization using homography-based motion compensation approach.

Nga *et al.* [145] subtracted the estimated camera flow multiplied by the camera direction ($d = \pm 1$) from the flow of each extracted spatio-temporal keypoint. However, in this work, only camera translation in both horizontal and vertical directions is considered. Different works [144, 78, 80] consider 2D polynomial affine motion models to compensate camera motion. In [144, 78], this model was employed to separate dominant motion, supposed to represent the camera motion, from residual motion in videos with dynamic scenes. More recently, Jain *et al.*[80] considered the same model. The compensated flow is computed in each point as the difference between the original flow vector and the affine flow vector of the same point. Thereby, each vector is compensated by its own affine flow and not by the one of the camera. To overcome these problems, we propose, for compensating the camera motion, to first determine the direction and magnitude of the dominant motion, then applying the affine motion model. Once this step is achieved, we obtain a situation similar to one where the camera is static.

5.3 Proposed method for motion segmentation

The proposed approach aims to detect and segment moving objects in a moving field of view. To reach this goal, first interest points are densely detected and extracted with a temporal step of N frames. Second, optical flows of detected keypoints between two frames are then computed by the iterative Lucas & Kanade optical flow using pyramids[140]. Then, the resulting vector field is passed to a flow clustering process which splits the list of flow vectors into clusters having similar flow direction within them and are different to each other. Based on the clustering results, camera motion is determined and compensated in order to extract foreground features. A schematic diagram of our algorithm is shown in Figure 5.1.

5.3.1 Computation of Optical Flow Vectors

In an image, some parts have almost the same color distribution such as the sky or the roof. These parts do not generally bring useful information and may add noise when computing the optical flow. In order to reduce the amount of data involved in motion analysis while preserving the most important structural features, we begin by detecting image edges using the canny edge detector [146], see Figure 5.2.

As follows, all the steps of the motion segmentation process will be applied on the edge frame. Once the set of interest points densely detected from the edge frame is defined, we

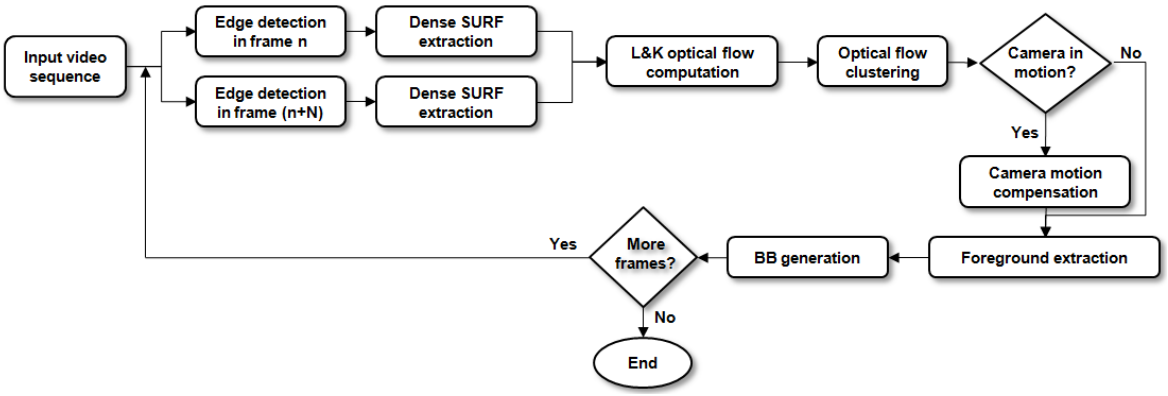


Figure 5.1. Proposed framework for motion segmentation.

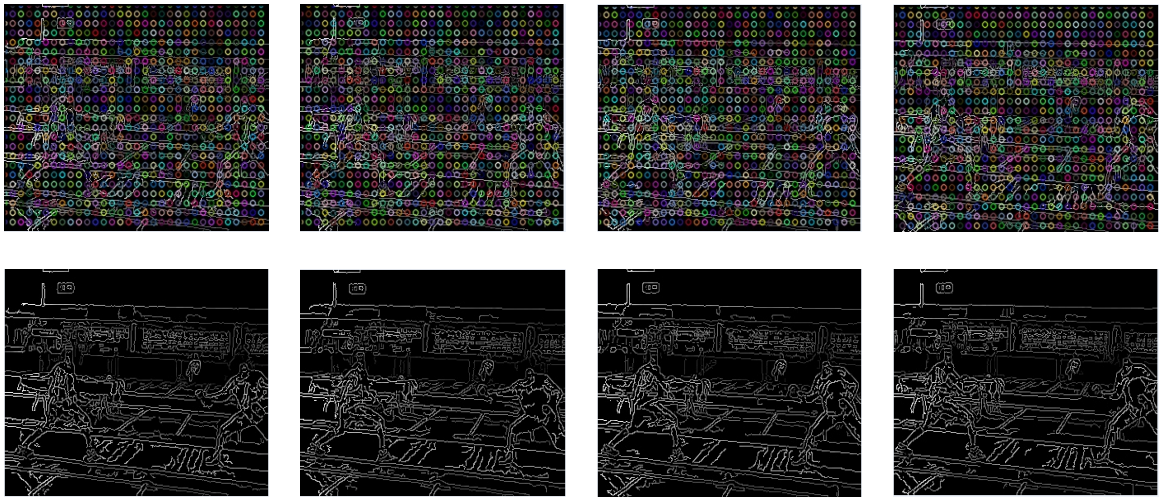


Figure 5.2. Edges detection for some classes from UCF101 dataset.

track them over the next one using the iterative Lucas & Kanade (LK) optical flow using pyramids. Figure 5.3 draws an example of LK optical flow before and after edge detection. We can easily notice in this figure, that after applying edge detetcion, many erroneous flow vectors were corrected.

The result of optical flow computation is a set of four-dimensional vectors V such as:

$$V = \{V_1 \cdots V_N | V_i = (x_i, y_i, a_i, m_i)\} \tag{5.1}$$

where x_i and y_i are the image coordinates of keypoint i, a_i and m_i are respectively the motion direction and magnitude of i. m_i corresponds to the distance between keypoint i in frame t and its corresponding feature in the next frame. Generally, optical flow is computed

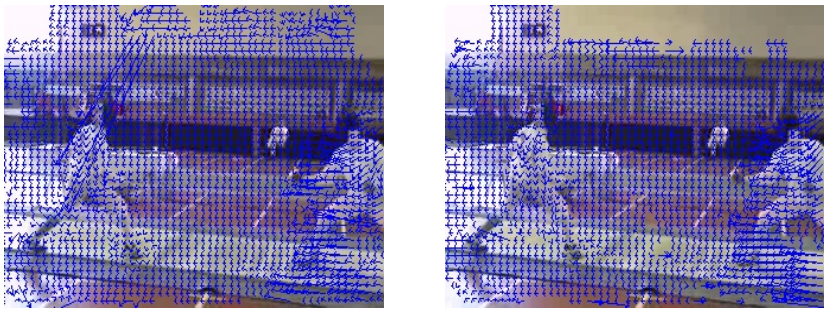


Figure 5.3. Results of LK optical flow computation before (left) and after (right) edge detection.

between two successive frames. However, the result may be unstable when objects either move too fast, too slow or stop between successive frames. In this thesis, we propose to extract keypoints and compute optical flow with a temporal step size of N frames. The choice of the temporal step's value varies according to the type of the video. For example, in sport videos such as running or swimming, motion is large. Hence, in order to obtain more information about the motion, it is better to choose small value of N. On the other hand, in every day activities videos such as talking or writing, motion is small. In such videos, N is chosen to be large.

The computation of optical flow vectors also allows the removal of static features. The latter are pixels that have optical flow component magnitudes lower than a threshold T in both x and y directions. They are also referred as "zero-motion" pixels. In our experiments, we set the minimum motion magnitude to 0.5 pixel per frame.

5.3.2 Detection and compensation of camera motion

Given a number of extracted dense keypoints and their associated motion vectors, we aim to separate local motions belonging to moving objects from the camera motion. In this work, we solve the problem of camera motion compensation by, first, confirming the existence of camera motion based on motion vectors. If detected, we determine the direction and magnitude of the camera motion before moving to the next step. Then, camera motion is compensated by applying affine transformations on the original frame.

Camera motion detection: In this step, we aim to find out how the camera moves at each frame basing on the assumption that if most points shift to the same direction, camera motion exists and it has the same direction as the moving points. This is derived from analysing the optical flows between two frames of a frame set. Therefore, we propose to

cluster optical flow vectors in order to eliminate outliers and to determine camera motion direction. In view of our real-time requirements, it is desirable to have a low number of clusters of similar optical flow vectors. Here, we don't seek to group motion vectors having the same magnitude or deviation. We are interested only on the direction of the motion. We define, as shown in figure 5.4, eight possible directions of the camera motion: six in

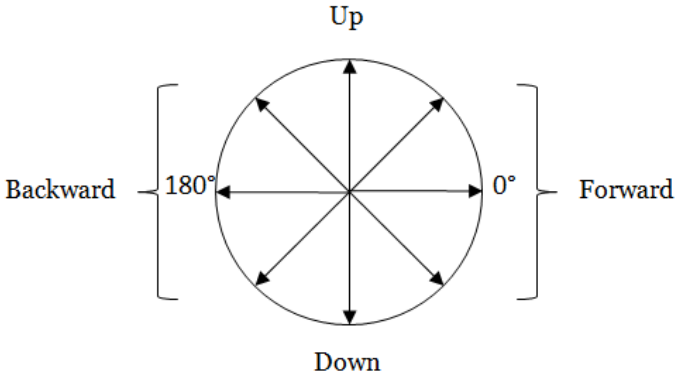


Figure 5.4. Possible directions of camera motion.

both horizontal direction, forward (up, down or right) or backward (up, down or left), and two in the vertical direction (up or down). In order to segment flow field into different groups, we employ the k-Nearest Neighbor (KNN) clustering algorithm.

After KNN clustering, some small clusters appear. These clusters do not belong to a dominant cluster and they are not relevant to the purposes of our work. Therefore, clusters with a size lower than a certain threshold are discarded. Figure 5.5 presents examples of optical flows clustering using KNN. Each of the eight directions of the camera is presented by a different color. In these images, it is easy to distinguish the moving objects from the background as well as determining the camera motion direction. In the first three images, the camera is moving in a different direction than the humans in the image. However in the last one, the man at the left has the same motion direction as the camera (presented in the same color) but their velocity is different. Hence the necessity of camera motion compensation.

Since we assumed that camera motion exists if most points move in the same direction, then, we seek to determine the size of each of the eight obtained clusters, get the largest one and compare its size to some threshold. Therefore, camera motion exists if equation (5.2) is satisfied:

$$\sup_{i \in \{1,..,8\}} \{s_i\} \geqslant k \tag{5.2}$$

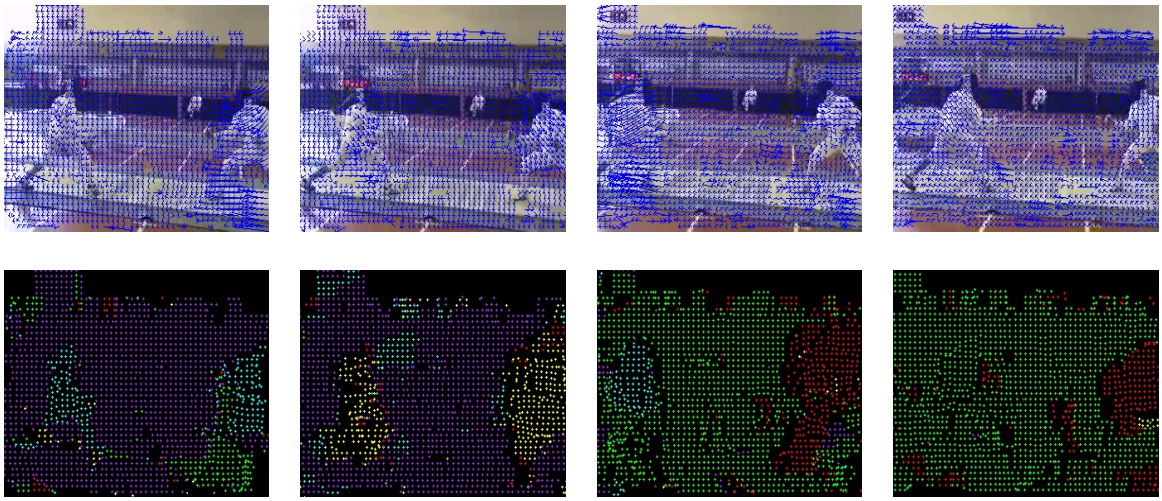


Figure 5.5. Optical flow clustering using KNN algorithm: the first row presents optical flows between two frames taken from a video sequence while the second row displays the results of KNN clustering. The keypoints are grouped into eight clusters with different colors depending on the flow direction: red (forward up), yellow (forward down), green (backward up), cyan (backward down), blue (forward to the right), purple (backward to the left), dark green (up) and orange (down).

Where s_i is the size of cluster i, i is the number of the cluster and k is a threshold representing the minimal required proportion of moving points. In our experiments, we set k as $\frac{N}{2}$ where N is the total number of detected points. As an example, in figure 5.5(first image), we can easily interpret that purple color is the dominant one. Hence, camera motion exits and the camera is moving horizontally to the left. The camera is supposed to be in rest if the above condition is not satisfied. If the camera is detected as in motion, then the camera motion magnitude and deviation are computed basing on the following equations:

$$m_m = mean \mid f_i \mid \tag{5.3}$$

$$\theta_m = mean(\theta_{f_i}) \tag{5.4}$$

Here, f_i and θ_{f_i} refer, respectively, to the flow and deviation of point i. m_m and θ_m refer, respectively, to the camera flow magnitude and deviation.

Camera motion Compensation: In Videos captured by a hand-held camera, camera motion is random. It is a combination of translation and rotation. In Nga *et al.*'s work[145],

only the camera translation is considered. Camera motion is compensated by subtracting the camera flow from the original flow of each SURF keypoints. So, the camera motion will not be correctly compensated if the motion is, for example, oblique. We propose to solve this problem by applying affine transformation on each frame in which camera motion is detected. The affine model [147] incorporates transformation such as translation, rotation and scaling (compressions or expansions). The transformation can be described as:

$$I' = D \times I + d \times T \tag{5.5}$$

Where I is the original frame, I' is the transformed frame, $D = \begin{bmatrix} s_x d_{xx} & s_y d_{xy} \\ s_x d_{yx} & s_y d_{yy} \end{bmatrix}$ is the deformation matrix accounting for rotation and scaling, d_{xx}, d_{xy}, d_{yx}, d_{yy} are the rotation parameters and s_x and s_y are the scaling ratios in the x and y directions. $T = \begin{bmatrix} d_x \\ d_y \end{bmatrix}$ is the translation vector.

In this work, we take under consideration only translation and rotation motions. Scaling is one of our future works. Therefore, the parameters s_x and s_y from the deformation matrix are equal to 1. Hence, I' from (5.5) becomes:

$$I' = \begin{bmatrix} \cos\theta_m & -\sin\theta_m \\ \sin\theta_m & \cos\theta_m \end{bmatrix} \times I + \begin{bmatrix} m_{mH} \\ m_{mV} \end{bmatrix} \tag{5.6}$$

Here m_{mH} (respectively m_{mV}) refers to the camera flow magnitude when the camera translates horizontally (respectively vertically). d equals 1 if the camera moves to positive direction or -1 if the camera moves to negative direction. In case of horizontal motion, $m_{mV} = 0$ and in case of vertical motion, $m_{mH} = 0$. Unlike in [80] and [145] where the motion of each flow vector is compensated independently, in our work, we apply the affine model on the whole image.

5.3.2.1 Motion segmentation

After compensating the camera motion, we reach a situation similar to one where the camera is static. Here, moving objects are segmented using a pixel-wise technique which is temporal differencing. It is the simplest method to extract moving objects and is robust to dynamic environments. It is similar to the background subtraction techniques. The only difference is that the background model is the previous frame. This algorithm classifies a new pixel as being a foreground pixel whenever $\| I(x,y) - I_{prev}(x,y) \| \geq T$ where T is a user

defined threshold. The obtained result is a binary image. However, due to camera noise and limitations of the background model, the foreground mask (binary image) typically contains numerous small "noise" clusters. These erroneous clusters can be removed by applying a noise filtering algorithm to the foreground mask. Removing these erroneous clusters in an early stage is desirable since they can interfere with later post-processing steps. In general, morphological operations are performed to remove noise and extract significant information from images. In our system, we used both morphological erosion and dilatation, respectively, to remove noise and unwanted objects. Erosion consists on convoluting an image A with some kernel B, calculating the local minima over the area of the kernel and replacing this value where the anchor of the kernel is located. After that, objects including many small holes and separated pixels may be connected into one cluster using the dilatation operation. Useless and small clusters are removed by setting limitation on their sizes. The remaining clusters represent the moving objects. Finally, a bounding box is drawn around each detected object. The aforementioned steps of our propose method for motion segmentation are applied on an input video with a temporal step with size N. Thus, we need to track the detected objects in the remaining frames (the frames between frame (n) and frame $(n+N)$). To accomplish this, we employ a template matching technique which is the normalized cross correlation [148].

Figure 5.6 emphasizes the effectiveness of our motion segmentation method. It can be observed that almost only local motions remain which are then employed, after filtering noise, to segment the motion. Our method succeeded to eliminate the motion induced by the camera leaving only humans/objects motion. However, in some cases, the process of camera motion compensation may have a reverse effect on motion segmentation. In fact, in some frames two or more dominant plans can coexist. Hence, the camera motion direction and deviation will not be determined correctly. For example in figure 5.6-fourth row, motion of two players can be easily detected before camera motion compensation. When applied, the latter adds some noise to the frame. At the end, we were able to solve this problem using morphological operations.

In case where no camera motion is detected, we admit that the detected flow belong to the objects/humans in motion. Hence, instead of applying, like we did previously, the temporal differencing technique, here, we propose to apply a second clustering of optical flow vectors based on the degree of similarity of their magnitudes, angles and closeness, under the assumption that optical flows of a single person/object have similar characteristics. We assume that two optical flow vectors, f_i and f_j, belong to the same

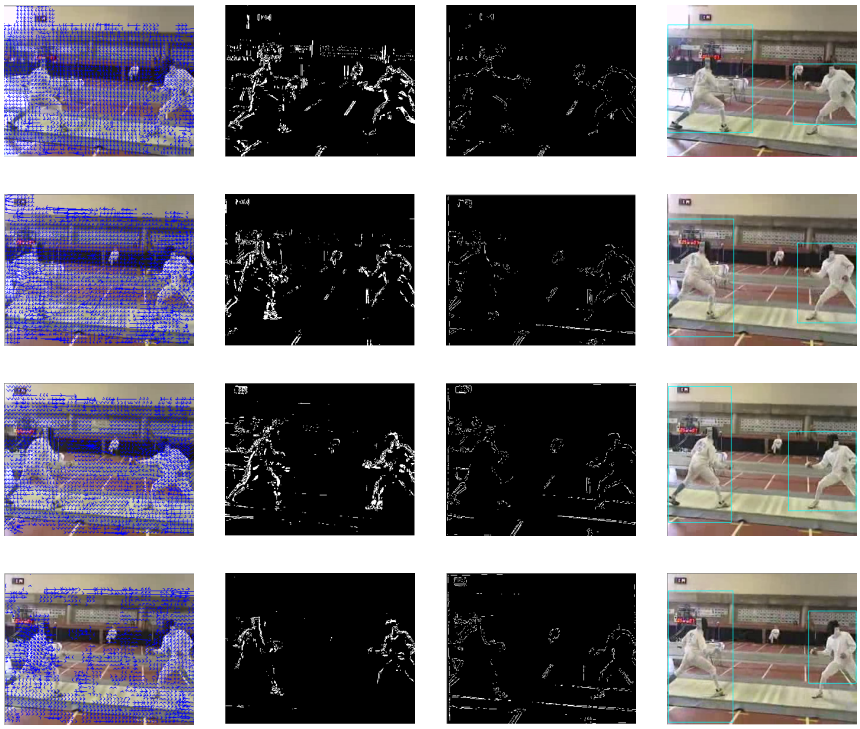


Figure 5.6. Results of our proposed method for motion segmentation. Camera motion exists in all the sequence. The first column presents a frame set of consecutive frames containing camera motion on which optical flow is drawn. The second column refers to the motion segmentation results before camera motion compensation. The third column shows the results of motion segmentation after camera motion compensation. Finally, the last column is the final segmentation after applying morphological operations.

cluster if the following assumptions are satisfied:

$$| l_i - l_j | \leq l_{th} \tag{5.7}$$

$$| \theta_i - \theta_j | \leq \theta_{th} \tag{5.8}$$

$$| posX_i - posX_j | \leq posX_{th} \tag{5.9}$$

$$| posY_i - posY_j | \leq posY_{th} \tag{5.10}$$

where l_i and l_j are the magnitudes of f_i and f_j. θ_i and θ_j are the deviations (angles) of f_i and f_j. (X_i, Y_i) and (X_j, Y_j) are the coordinates of optical flow vectors. Finally, l_{th},

θ_{th}, $posX_{th}$ and $posY_{th}$ are the thresholds for optical flow clustering. All detected flow vectors are compared two-by-two basing on these similarity comparisons leading to form a fixed number of clusters. In order to remove noisy and meaningless clusters, we discard ones with size smaller than a threshold. The remaining clusters belong to the foreground. A bounding Box is drawn around each one. Figure 5.7 presents the segmentation results derived from the optical flow clustering technique as well as the results of using frame differencing technique. The first technique (row 5) reached better segmentation results. It succeeds to capture the whole human motion while the second technique (rows 3 and 4) leads to loose information and only some parts of the motion were segmented.

5.4 Proposed framework for human action recognition

The ultimate goal of this work is to introduce an efficient method to achieve accurate and fast action recognition in big dataset. The overall architecture proposal of other spatial-temporal segmentation and the associated architecture are highlighted in Figure 5.8.

5.4.1 Selective temporal segmentation

In order to achieve a relevant spatial-temporal video segmentation, we extract interest points located within the detected Bounding box as it contains moving objects/persons. Here, dense SURFs are the most appropriate interest points to be employed in order to exploit the maximum spatial information in video frames [126]. This process begins by dense SURFs detection in the BBs extracted from the first frame. These descriptors trajectories are then tracked until the end of the video patch. In order to guarantee a fair comparison, we use the same settings as other chapters and as the state-of-the-art.
In this chapter, we briefly describe the theoretical details of all the phases involved in our human action recognition computation pipeline:

- A group of SURF that covers significant moving human/objects parts are defined as: G-SURF = 49.
- The concepts of selective snippets, previously described, and G-SURF are employed. In fact, for three successive frames $(n, n+1, n+2)$, trajectory features can be extracted for a moving SURF. Considering that α is the angle between the lines segments supporting the motion of a SURF from the couple of frames $(n, n+1)$ and $(n+1, n+2)$,

Figure 5.7. Results of motion segmentation in videos acquired by static camera. The first row presents a set of consecutive frames on which optical flow is drawn. The second row refers to optical flow clustering using KNN clustering. The third and forth rows show the results of temporal differencing technique. Finally, the last row is the result of motion segmentation after optical flow second clustering.

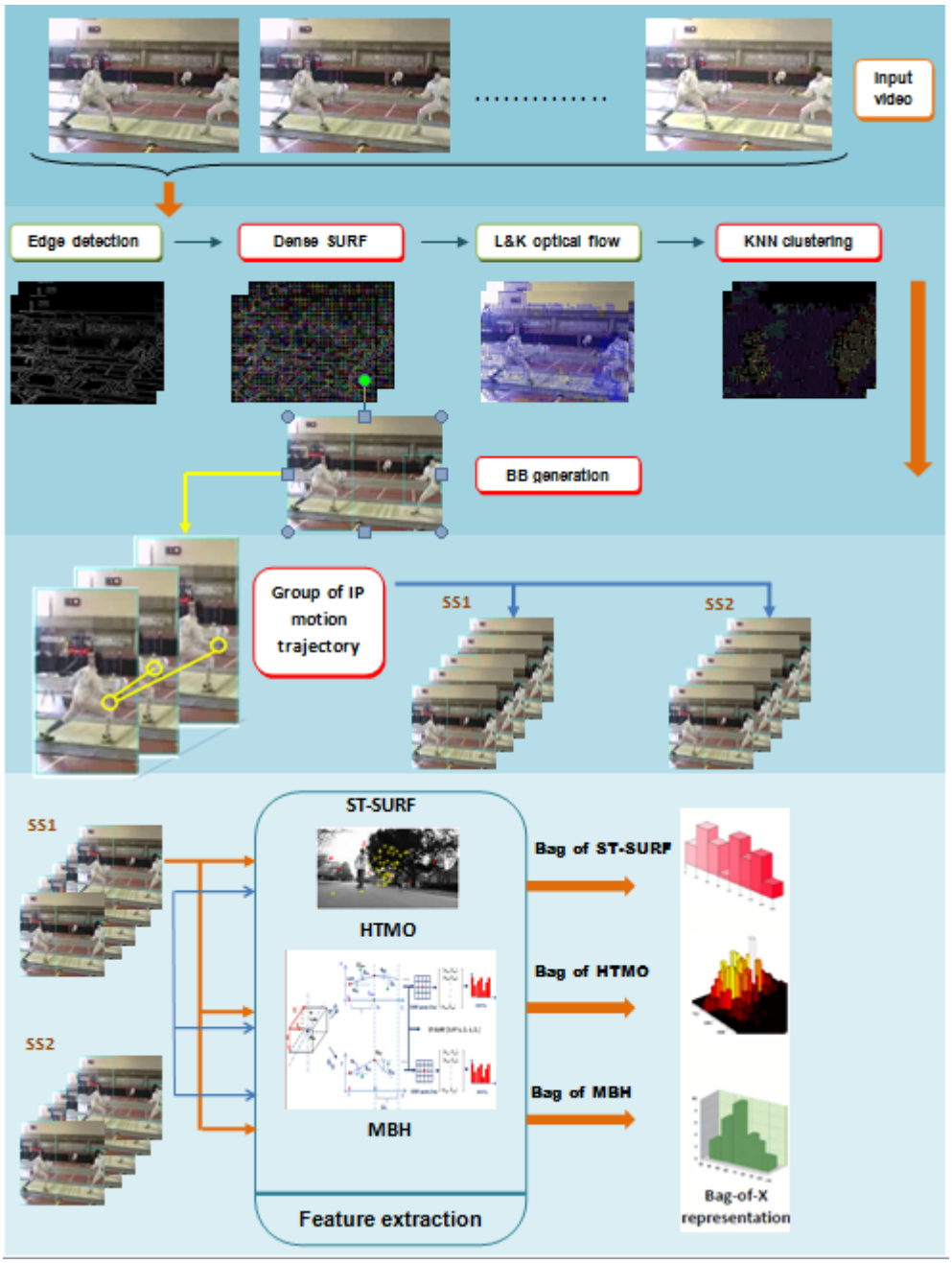


Figure 5.8. Human action "phone call"

a comparison between α to α_{max} and α_{min} (α_{max} and α_{min} are threshold empirically set) is performed in order to segment a succession of frames (SS) in which each SURF has an α lower than α_{max} and greater than α_{min}.

- A displacement vector $D_{n,n+1}$ is computed from the frame (n) to the frame $(n+1)$, and $D_{n,n+2}$ from the frame (n) to the frame $(n+2)$.

$$\alpha = \arccos \frac{D_{n,n+1} \cdot D_{n+1,n+2}}{\|D_{n,n+1}\| \times \|D_{n,n+2}\|} \tag{5.11}$$

5.4.2 Feature extraction

To increase the efficiency of the proposed camera motion compensation, we employ several local spatio-temporal descriptors. Based on our motion trajectory extraction framework and practical considerations, the overall extraction process can be summarized as follows:

- First, trajectories are extracted by optical flow detection. In this work, we employ the optical flow proposed by Sun et al. [106].

- Every bounding box based selective snippet corresponds to a volume of frames in the 3D space called SS Volume (SS_v). This cubic volume is characterized by the frame number (FN) varying from 1 to t_{max}. The frame surface dimensions (FS) vary from x to x_{max} in the x direction, and from y to y_{max} in the y direction.

- Descriptors computation: To extract IP motion trajectory orientation, we project its motion vectors onto the planes (t,x) and (t,y) of the SS_v to define an angle for each projection of the first angle α_x between optical flow and the plane (t,x), the angle α_y between the plane (t,y) and the motion vector.

$$\propto_x = 90 - \frac{180}{\Pi}\arctan(\frac{u}{n}), \propto_y = 90 - \frac{180}{\Pi}\arctan(\frac{v}{n}). \tag{5.12}$$

  The projection of each $SURF's$ motion vector on the planes (t,x) and (t,y) yields two lines L_x and L_y. The orthogonal projection of SS_{ccx} and SS_{ccy} onto the lines L_x and L_y allows computing the two distances D_x and D_y between the SS_v center and the lines supporting the motion vectors (L_x and L_y).
  For an IP located at (x,y,t), the distances D_x and D_y are given by:

$$D_x = D_{xu} - D_{tv}, D_y = D_{yv} - D_{tu} \tag{5.13}$$

- Histogram of motion trajectory orientation (HMTO) computation: Both $HMTO_x$ and $HMTO_y$ are extracted from a SURF centered patch. The patch is a square region with size $20s$ where s represent the current scale.

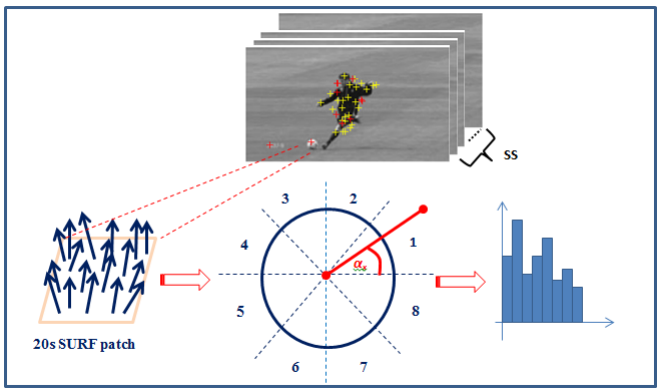


Figure 5.9. An overview of HMTO extraction.

- Motion boundary histogram (MBH)computation. The final MBH_x and MBH_y are 96D ($2 \times 2 \times 3 \times 8$) features set [81, 33].

- Spatio-temporal SURF (ST-SURF)computation: A 68D ST-SURF contains: spatial information driven by the SURF, temporal information driven by the optical flow and the size of this descriptor. It also provides localization information. The latter will add spatial information to the bag of words encoding step [43].

5.5 Experiments and results

5.5.1 Experimental settings

In earlier chapters, we introduced the overall approach for motion segmentation and action description. Our proposed technique for motion segmentation does not require no assumptions about the first frame nor initialization or training steps. We start the segmentation process with dense SURF features extraction with temporal step size of N frames. In our experiments, we fix N as 3 so that small motions will not be lost and fast motions will be captured with no errors. Then, we compute LK optical flow. The flow vectors are clustered to determine whether camera motion exists. If it does not, we conduct a second clustering of the flow vectors basing on the degree of similarity of their magnitudes, angles and closeness. We fixed the thresholds experimentally as follows: $l_{th} = 15$, $\theta_{th} = 2.0$,

$posX_{th} = 45$ and $posY_{th} = 305$.

The employed descriptors in the action recognition process provide a rich video representation in term of space and motion of moving interest points. From each clip, we extract local spatio-temporal features as ST-SURF. As described previously, the extracted ST-SURF is a 68D vector (64D SURF, α_x, D_x, α_y, D_y). We also extract square shape patches surrounding the detected SURFs. The size of each detected patch is 20s. For each one, a HTMO is computed in both planes (x, t) and (y, t). $HMTO_x$ and $HMTO_y$ are both 96D vectors. To reinforce our action recognition system, we used motion boundary histogram MBH as a motion descriptor and also for its ability to remove camera motion. MBH_x and MBH_y are 96D histograms.

We performed an experiment using the bag of words approach to provide baseline results on the UCF101 dataset. The classification step starts by k-mean clustering applied on a set of 10^6 randomly selected features to build a visual dictionary for every extracted descriptor type (ST-SURF, $HMTO_x$, $HMTO_y$, MBH_x, MBH_y). For each one, we construct 4000 visual words. The k-mean clustering is initialized 8 times and we kept the configuration with the lowest error rate. The extracted histograms are L_2 normalized to ensure better visual quality. Finally, to classify the actions, we use a non linear SVM with an RBF^2_χ Kernel [39].

$$K(v_i, v_j) = \exp(-\sum \frac{1}{A^c} D(v_i^c, v_j^c)), \tag{5.14}$$

Where $D(v_i^c, v_j^c)$ is the χ^2 distance between video v_i and v_j of the channel c. A^c is the mean distance value of the training features.

5.6 Dataset

In this thesis, the latest experiments are carried on a big realistic dataset called UCF101 [103]. It includes total number of 101 action classes which we have divided into five types:

- Human-Object Interaction.
- Body-Motion.
- Human-Human Interaction
- Playing Musical Instruments.
- Sports.

UCF101 is an extension of UCF50 which included the following 50 action classes, [104]: Baseball Pitch, Basketball, Shooting, Bench Press, Biking, Billiards Shot, Breaststroke, Clean and Jerk, Diving, Drumming, Fencing, Golf Swing, High Jump, Horse Race, Horse Riding, Hula Hoop, JavelinThrow, Juggling Balls, Jumping Jack, Jump Rope, Kayaking, Lunges, Military Parade, Mixing Batter, Nun chucks, Pizza Tossing, Playing Guitar, Playing Piano, Playing Tabla, Playing Violin, Pole Vault, Pommel Horse, Pull Ups, Punch, Push Ups, Rock Climbing Indoor, Rope Climbing, Rowing, Salsa Spins, Skate Boarding, Skiing, Skijet, Soccer Juggling, Swing, TaiChi, Tennis Swing, Throw Discus, Trampoline Jumping, Volleyball Spiking, Walking with a dog, Yo-Yo.

The following 51 new classes are introduced in UCF101: Apply Eye Makeup, Apply Lipstick, Archery, Baby Crawling,Balance Beam, Band Marching, Basketball Dunk, Blow Drying Hair, Blowing Candles, Body Weight Squats, Bowlng, Boxing-Punching Bag, Boxing-Speed Bag, Brushing Teeth, Cliff Diving, Cricket Bowling, Cricket Shot, Cutting In Kitchen, Field Hockey Penalty, Floor Gymnastics, Frisbee Catch, Front Crawl, Hair cut, Hammering, Hammer Throw, Handstand Pushups, Handstand Walking, Head Massage, Ice Dancing, Knitting, Long Jump, MoppingFloor, Parallel Bars, Playing Cello, Playing Daf, Playing Dhol, Playing Flute, Playing Sitar, Rafting, Shaving Beard, Shot put, Sky Diving, Soccer Penalty, Still Rings, SumoWrestling, Surfing, Table Tennis Shot, Typing, Uneven Bars, Wall Pushups, Writing On Board.
Clip Groups: The clips of one action class are divided into 25 groups which contain 4-7 clips each. The clips in one group share some common features, such as the background or actors. The colors on each bar illustrate the durations of different clips included in that class. The chart shown in Figure 5.10, illustrates the average clip length (green) and total duration of clips (blue) for each action class. The videos are downloaded from YouTube [10] and the irrelevant ones are manually removed. All clips have fixed frame rate and resolution of 25 FPS and 320×240 respectively.

5.7 Experimental results and discussion

In this section, we report and discuss the motion segmentation and action recognition results extracted from UCF101 datasets. The purpose of this discussion is to highlight the keys of success and weaknesses of the proposed action recognition system.

Figure 5.10. UCF 101 actions.

5.7.1 Motion segmentation

We, first, present the evaluation of the proposed motion segmentation process. The experiments are carried out on 125 randomly picked videos (25 videos from each of the five categories) from the UCF dataset. The latter is very complex. It represents different indoor and outdoor scenes with moving foreground, objects, complex background and camera motion. In fact, this dataset is dedicated mainly to the task of action recognition. Hence, as far as we know, there are no evaluations of proposed motion segmentation algorithms based on this dataset that we may compare our method to.
The system's performance is evaluated in terms of the average F-measure given by:

$$F = \frac{2 \times R_c \times P_r}{R_c + P_r} \tag{5.15}$$

where P_r is precision and R_c is the recall for bounding boxes annotations, for each video. These measures are assessed basing on some bounding boxes annotations provided in [32]. Our main purpose from segmenting motion, is to restrict the amount of data involved in studying human actions. Hence, we aim to detect a bounding box covering as much motion as possible. Table 5.1 reports the obtained results. For the Sports (74.50%), Playing Musical Instrument (88.75%),Human-Object Interaction (87.83%), Body-Motion Only(85.45%), Human-Human Interaction (84.53%).

In general, the camera motion segmentation process helps improving the accuracy of

Table 5.1. F-measure results over the UCF101 dataset.

Action class	F-measure (%)
Sports	74.50
Playing Musical Instrument	88.75
Human-Object Interaction	87.83
Body-Motion Only	85.45
Human-Human Interaction	84.53

motion segmentation. Also, for fixed scenes, employing a second clustering of motion flow vectors enhances moving objects extraction. We consider these results satisfying especially since we make no assumptions about the first frame and our process does not require any initialization or training steps. Sports actions are considered as the most challenging ones as they include important motions of humans along with camera motion. The majority of those videos were captured outdoors with the presence of trees and audience. Despite these effects, sports actions motion segmentation reached acceptable results. The performed

motion segmentation makes the action recognition task easier even with presence of camera motion.

5.7.2 Action Recognition

As described before, we use the same settings and evaluation metrics of the state-of-the-art. The accuracy rates reported for the predefined action types are shown in Table 5.2. For the Sports (87.23%), Playing Musical Instrument (79.4%), Human-Object Interaction (86.07%), Body-Motion Only(85.19%), Human-Human Interaction (88.61%).
We can notice that Human-Human Interaction actions achieve the highest accuracy since the spatio-temporal segmentation we introduced in this thesis highlight human bodies, thus the feature extraction is performed in the humans bounding boxes boosts significantly human detection. Performing sports action achieves a reasonable accuracy of 87.23%, this is due to two factors the first one is the temporal segmentation while the second one is the motion based extraction features. In fact, sports actions show important motion which is very well described in our proposed approaches. Despite that Human-objects and Body motion actions are not based on significant motion, the classification shows satisfactory results. We believe that pixel motion segmentation precision in detecting motion is a good cue to explore human action.

Table 5.2. Recognition results over the UCF101 dataset.

Action class	Accuracy (%)
Sports	87.23%
Playing Musical Instrument	83.4%
Human-Object Interaction	86.07%
Body-Motion Only	85.19%
Human-Human Interaction	88.61%

We present the results of our approach compared to trajectory and motion based video description approaches in Table 5.3. MBH descriptor is associated with several approaches to detect human actions since it is based on optical flow. This proves that combining MBH with different descriptors is a straightforward way to improve the results. The proposed approach which combines ST-SURF, HTMO and MBH gives an accuracy rate of 79.2% equivalent to the state-of-the-art trajectory based video description. As expected, the proposed spatio-temporal segmentation improves the proposed approach by 6.1% achieving 85.3% of accuracy in the challenging realistic big dataset UCF101. Compared with trajectory based descriptors, the proposed approach gives good performances.

Approach	Descriptor	UCF101: Accuracy %
Trajectory	TrajShape	47.1
Trajectory+ Loclal decriptors	TrajShape+MBH+HOG+HOF	72.8
Local descriptors	HOG3D+MBH+HOG+HOF	78.9
Trajectory	Dense trajectory	85.9
Trajectory	Dense trajectory+PSIFT	85.7
Trajectory	MBH	85.7
Trajectory	BB+ST-SURF+HTMO+MBH	85.3
Trajectory	ST-SURF+HTMO+MBH	79.2
Trajectory	BB+ST-SURF+HTMO+MBH	86.1

Table 5.3. Trajectory based descriptor performances over the UCF101 dataset.

5.7.3 Comparison with the state of the art

The results given by the-state-of-the-art are given in Table 5.4. They are reported from the original papers [149, 150, 32, 151, 104, 152, 153]. The overall performances on UCF101 dataset are 85.3%. These Results are significantly better than those reported in [104] using standard bag of words method with overall accuracy of 44.5%. In [149], authors used dense trajectory computed for fixed frame length L = 16 and L = 17. The overall performance rate is 47.1% using trajectory descriptor. Combined with MBH, HOG and HOF their trajectory based approach reaches 72.8%. These performances still lower than our results. We believe that this is due to the use of a fixed frame number. We also outperform the results given in [150]. Authors used multi-channel approach for Local Part Model LPM algorithm for efficient action recognition. Their approach was based on the fusion of HOG, HOF, HOG3D and MBH. They achieved 78.9% of average accuracy. Dense trajectory features were used in [32]. Author applied Fisher vector and spatio-temporal pyramids to embed structure information. Finally, a linear SVM combining everything gives the performance of 85.9% about 0.6% better than our results. This proves the importance of motion cues in detecting human actions, this also encourages us to investigate more learning approaches that the SVM we used in this thesis. Fishor vectors gives good results in [151]. In fact, authors extract features from both video and keyframe modalities. They used dense trajectory features associated with HOG and Motion Boundary Histogram (MBH), then they encode them as Fisher vectors. To represent action-specific scene context, we compute local SIFT pyramids on grayscale (P-SIFT) and opponent color keyframes (P-OSIFT) extracted as the central frame of each clip. improve accuracy by using L1-regularized logistic regression (L1LRS) for stacking classifier outputs 85.7%, 0.2% better than our

method. The results given in [152] are lower than ours. In fact, authors provide an extensive empirical evaluation of CNNs on large scale video classification 63.3%. However, in [153], authors investigate architectures of indiscriminately trained deep Convolutional Networks (ConvNets) for action recognition in video. This method achieves 87.6% which is the best result. This, also, highlights the importance of the classification task investigation, especially in term of deep classification.

Table 5.4. Some state of the art recognition results over the UCF101 dataset.

Method	Year	Accuracy (%)
Murthy et al. [149]	2013	72.8
Shi et al. [150]	2013	78.9
Wang et al. [32]	2013	85.9
Karaman et al. [151]	2013	85.7
Khuramm et al. [104]	2012	44.5
Karpathy et al.[152]	2014	63.3
Simonyan et al. [153]	2014	87.6

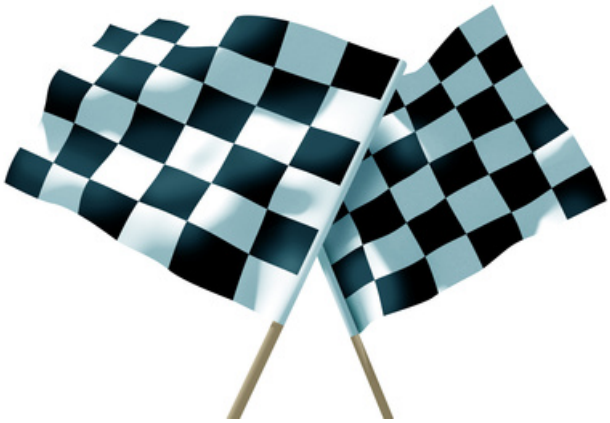
5.8 Summary and Conclusion

In this chapter, we presented an end-to-end framework for human action recognition in big dataset. As part of this effort, we introduced a new human motion segmentation process based on studying optical flows induced by human motion. The flows are clustered to determine the existence of camera motion. The latter is compensated by means of affine transformation. Pre-processing operations are performed in order to extract humans motion. The proposed approach achieves a reasonable trade-off between high accuracy and prohibitive computational cost.

The video segmentation task is followed by the video description process which includes different descriptors that shares the same purpose: extracting the maximum motion and appearance cues. To this end, we employed the Histogram of motion trajectory orientation. The latter is based on the tracking along a trajectory of moving regions. The distribution of motion angles we extracted has several advantages involved in different action recognition steps. For instances, it extracts meaningful local motion information in a dense sampling. It also have localization cues which avoid using extra computations to add spatial information ie. spatial pyramid.

CHAPTER 6

CONCLUSION AND PERSPECTIVES



Chaque enfant qu'on enseigne,
est un homme qu'on gagne. Victor Hugo

Contents

6.1 Conclusion

In this dissertation we presented a set of approaches aimed to improve the performance of both action detection and video description. The algorithms were specially designed to recognize human actions in video under a number of challenging conditions. These conditions included camera motion, low resolution and complex datasets.

In the realm of action detection, the first issue we addressed was that of spatio-temporal video segmentation. To accomplish this task, we used an original interest points tracking approach to detect moving humans/objects. This work is based on Speeded-up robust features (SURF) as interest points (IP). The proposed approach yields video segments containing significant action and does not rely on a fixed number of frames.

Secondly, we addressed the issue of camera motion compensation in the context of spatio-temporal video segmentation. In this thesis, we presented a method for human motion segmentation in dynamic scenes. The proposed process encompasses a collection of techniques enabling camera motion compensation as well as motion segmentation in complex videos. Camera motion compensation is achieved by clustering optical flow vectors of densely extracted key interest points. The largest cluster represents the camera motion, which is compensated for using affine motion models. Then, motion is segmented using temporal differencing between two frames. If the camera is static, we recommend segmenting motion by applying a second clustering of flow vectors based on their degree of similarity. The formed clusters belong to the foreground. Finally, a bounding box is drawn around each moving object.

We also focused on deploying these methods in video description frameworks. We enriched the state-of-the-art by proposing two different video descriptors. The first video descriptor is based on local interest point extension to the temporal domain. The second descriptor is another local descriptor based on the distribution of the trajectory of interest points in a predefined bounding box. Both descriptors' performances were tested using different schemes, such as single descriptors or a fusion of different descriptors.

In conclusion, the current research has *analyzed*, *developed* and *evaluated* different methods for not only action detection and video segmentation but also action recognition in video. A comparison of our experimental results with those of existing approaches has demonstrated the promise of our new methods.

6.2 Summary of Contributions

Our main contributions are summarized below

- *Trajectory based action detection:* Within the purview of the work proposed in this thesis, we addressed the spatio-temporal action detection. We proposed the detection of human action and action boundaries to localize action. The proposed technique relies on Interest Points (IP) detection. We chose to use the Speeded-Up Robust Feature, as it is a high performing and fast interest points detector. The detected interest point trajectory can then be tracked and the moving IP selected.

- *Dense trajectory based action detection:* In this part of the thesis, we focused on optimizing human action detection, since it has already been proven that real performances increase with respect to time consumption. We proposed in this context to detect moving human or objects in the scene. This is achieved first by detection of dense IP and then by optical flow computation, for which we designed bounding, boxes to surround the humans and/or objects in movement.

- *Spatio-temporal SURF (ST-SURF):* We proposed a new algorithm based on the extension of the speeded-up robust features (SURF) to the temporal domain. This feature captures the spatial information provided by the SURF and the motion information brought by the optical flow. In addition, ST-SURF contains localization information, which is not obtained by the Bag of Visual Words (BoVW) approach.

- *Histogram of trajectory motion orientation (HTMO):* We investigated a complementary source of information based on the tracking of the IP trajectory. HTMO captures joint cues between the distribution of motion and appearance of constituent IP. The motion trajectory is extracted by optical flow computation.

The descriptors we designed were tested in different scenarios. In order to evaluate our proposed approaches, we began by using simple yet challenging datasets. At the most basic level, we analyzed scenes in which one actor was performing one action without the influence of either camera motion or a challenging background. We then experimented with more complex datasets comprised of realistic videos captured by amateurs from real scenes.

6.3 Future work

In the course of the research carried out for this thesis, a number of possible directions for further research have been identified. Relating to the two main parts of this thesis, these research ideas can be divided into two separate sections. *First*, despite the extensive research already carried out in the field of human action detection, there is still room for improvement. In the current algorithms, the SURF detector is used in a dense sampling. Other interest points and methods based on more stable and effective dense sampling deserve careful consideration. An extension of our framework could be improved with the use of different descriptors.

Furthermore, an automatic segmentation could be investigated. For long video sequences a clustering could be performed on the extracted trajectories to accelerate the recognition process. We could also to apply trajectory features for video retrieval via a multimodal scheme.

Secondly, regarding the evaluation of video description, further work should be focused on more motion-based descriptors.

Conducting experiments using other classification methods, such as deep schemes or fisher vectors, could also significantly contribute to better human action recognition. Identifying these deficiencies would prepare the way for further improvements.

Experimenting with new datasets should also be considered for futur research. The main issue when dealing with realistic videos is that of a significant increase in running time. Thus, developing new algorithms that are able to segment low resolution videos would potentially increase the efficiency of action detection.

List of Tables

List of Figures

Bibliography

[1] Yu-Gang Jiang, Chong-Wah Ngo, and Jun Yang, "Towards optimal bag-of-features for object categorization and semantic video retrieval," in *Proceedings of the 6th ACM international conference on Image and video retrieval*, New York, NY, USA, 2007, CIVR '07, pp. 494–501, ACM.

[2] Weiming Hu, Tieniu Tan, Liang Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, pp. 334–352.

[3] Jake K Aggarwal and Quin Cai, "Human motion analysis: A review," in *Nonrigid and Articulated Motion Workshop*. IEEE, 1997, pp. 90–102.

[4] Dariu M Gavrila, "The visual analysis of human movement: A survey," *Computer vision and image understanding*, vol. 73, no. 1, pp. 82–98, 1999.

[5] Jessica JunLin Wang and Sameer Singh, "Video analysis of human dynamicsa survey," *Real-time imaging*, vol. 9, no. 5, pp. 321–346, 2003.

[6] Hilary Buxton, "Learning and understanding dynamic scene activity: a review," *Image and vision computing*, vol. 21, no. 1, pp. 125–136, 2003.

[7] Jake K Aggarwal and Sangho Park, "Human motion: Modeling and recognition of actions and interactions," in *3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on*. IEEE, 2004, pp. 640–647.

[8] Pavan Turaga, Rama Chellappa, Venkatramana S Subrahmanian, and Octavian Udrea, "Machine recognition of human activities: A survey," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 11, pp. 1473–1488, 2008.

[9] JK Aggarwal and Michael S Ryoo, "Human activity analysis: A review," *ACM Computing Surveys (CSUR)*, vol. 43, no. 3, pp. 16, 2011.

[10] Youtube, "Statistiques@ONLINE," June 2009.

[11] Tanaya Guha and Rabab K Ward, "Learning sparse representations for human action recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 8, pp. 1576–1588, 2012.

[12] Konrad Schindler and Luc Van Gool, "Action snippets: How many frames does human action recognition require?," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[13] Akitsugu Noguchi and Keiji Yanai, "A surf-based spatio-temporal feature for feature-fusion-based action recognition," in *Trends and Topics in Computer Vision*, pp. 153–167. Springer, 2012.

[14] Ivan Laptev and Patrick Pérez, "Retrieving actions in movies," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–8.

[15] Daniel DeMenthon and David Doermann, "Video retrieval using spatio-temporal descriptors," in *Proceedings of the eleventh ACM international conference on Multimedia*. ACM, 2003, pp. 508–517.

[16] Koen EA van de Sande, Jasper RR Uijlings, Theo Gevers, and Arnold WM Smeulders, "Segmentation as selective search for object recognition," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1879–1886.

[17] Jutta Willamowski, Damian Arregui, Gabriella Csurka, Christopher R Dance, and Lixin Fan, "Categorizing nine visual classes using local appearance descriptors," *illumination*, p. 21, 2004.

[18] Paul Viola and Michael Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition,CVPR*. IEEE, 2001, pp. I–511.

[19] Oded Maron and Tomás Lozano-Pérez, "A framework for multiple-instance learning," *Advances in neural information processing systems*, pp. 570–576, 1998.

[20] Tae-Kyun Kim, Shu-Fai Wong, and Roberto Cipolla, "Tensor canonical correlation analysis for action classification," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. IEEE, 2007, pp. 1–8.

[21] Zhe Lin, Zhuolin Jiang, and Larry S Davis, "Recognizing actions by shape-motion prototype trees," in *IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 444–451.

[22] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, Jiri Matas, Frederik Schaffalitzky, Timor Kadir, and Luc Van Gool, "A comparison of affine region detectors," *International journal of computer vision*, vol. 65, no. 1-2, pp. 43–72, 2005.

[23] Heng Wang, Muhammad Muneeb Ullah, Alexander Klaser, Ivan Laptev, Cordelia Schmid, et al., "Evaluation of local spatio-temporal features for action recognition," in *BMVC 2009-British Machine Vision Conference*, 2009.

[24] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, "Surf: Speeded up robust features," in *Computer Vision–ECCV 2006*, pp. 404–417. Springer, 2006.

[25] Moshe Blank, Lena Gorelick, Eli Shechtman, Michal Irani, and Ronen Basri, "Actions as space-time shapes," in *International Conference on Computer Vision*. IEEE, 2005, vol. 2, pp. 1395–1402.

[26] Aaron F. Bobick and James W. Davis, "The recognition of human movement using temporal templates," *Transactions of Pattern Analysis and Machine Intelligence*, vol. 23, no. 3, pp. 257–267, 2001.

[27] DM Gavrila, LS Davis, et al., "Towards 3-d model-based tracking and recognition of human movement: a multi-view approach," in *International workshop on automatic face-and gesture-recognition*. Citeseer, 1995, pp. 272–277.

[28] Hironobu Fujiyoshi, Alan J Lipton, and Takeo Kanade, "Real-time human motion analysis by image skeletonization," *IEICE Transactions on Information and Systems*, vol. 87, no. 1, pp. 113–120, 2004.

[29] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele, "Pictorial structures revisited: People detection and articulated pose estimation," in *Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 1014–1021.

[30] Vasu Parameswaran and Rama Chellappa, "View invariance for human action recognition," *International Journal of Computer Vision*, vol. 66, no. 1, pp. 83–101, 2006.

[31] Fengjun Lv and Ramakant Nevatia, "Single view human action recognition using key pose matching and viterbi path searching," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8.

[32] Heng Wang and Cordelia Schmid, "Lear-inria submission for the thumos workshop," in *ICCV Workshop on Action Recognition with a Large Number of Classes*, 2013.

[33] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International Journal of Computer Vision*, pp. 1–20, 2013.

[34] Christian Schuldt, Ivan Laptev, and Barbara Caputo, "Recognizing human actions: a local svm approach," in *Proceedings of the 17th International Conference on Pattern Recognition, ICPR*. IEEE, 2004, pp. 32–36.

[35] Juan Carlos Niebles, Hongcheng Wang, and Li Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," *International Journal of Computer Vision*, , no. 3, pp. 299–318, 2008.

[36] Piotr Dollár, Vincent Rabaud, Garrison Cottrell, and Serge Belongie, "Behavior recognition via sparse spatio-temporal features," in *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*. IEEE, 2005, pp. 65–72.

[37] Paul Scovanner, Saad Ali, and Mubarak Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *Proceedings of the 15th international conference on Multimedia*. ACM, 2007, pp. 357–360.

[38] Hueihan Jhuang, Thomas Serre, Lior Wolf, and Tomaso Poggio, "A biologically inspired system for action recognition," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–8.

[39] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld, "Learning realistic human actions from movies," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[40] Geert Willems, Tinne Tuytelaars, and Luc Van Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *Computer Vision–ECCV 2008*, pp. 650–663. Springer, 2008.

[41] Alexander Klaser and Marcin Marszalek, "A spatio-temporal descriptor based on 3d-gradients," 2008.

[42] Lahav Yeffet and Lior Wolf, "Local trinary patterns for human action recognition," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 492–497.

[43] Sameh Megrhi, Wided Souidene, and Azeddine Beghdadi, "Spatio-temporal salient feature extraction for perceptual content based video retrieval," in *Colour and Visual Computing Symposium (CVCS), 2013*. IEEE, 2013, pp. 1–7.

[44] Ivan Laptev, "On space-time interest points," *International Journal of Computer Vision*, vol. 64, no. 2-3, pp. 107–123, 2005.

[45] Navneet Dalal and Bill Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005.* IEEE, 2005, vol. 1, pp. 886–893.

[46] Ivan Laptev and Tony Lindeberg, "Local descriptors for spatio-temporal recognition," in *Spatial Coherence for Visual Motion Analysis*, pp. 91–103. Springer, 2006.

[47] David G Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[48] Ramesh Jain and H-H Nagel, "On the analysis of accumulative difference pictures from image sequences of real world scenes," *Transactions on Pattern Analysis and Machine Intelligence*, , no. 2, pp. 206–214, 1979.

[49] Alessandro Neri, Stefania Colonnese, Giuseppe Russo, and Paolo Talone, "Automatic moving object and background separation," *Signal Processing*, vol. 66, no. 2, pp. 219–232, 1998.

[50] Christopher Richard Wren, Ali Azarbayejani, Trevor Darrell, and Alex Paul Pentland, "Pfinder: Real-time tracking of the human body," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780–785, 1997.

[51] Chris Stauffer and W. Eric L. Grimson, "Learning patterns of activity using real-time tracking," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747–757, 2000.

[52] Jens Rittscher, Jien Kato, Sébastien Joga, and Andrew Blake, "A probabilistic background model for tracking," in *Computer VisionECCV 2000*, pp. 336–350. Springer, 2000.

[53] Antoine Monnet, Anurag Mittal, Nikos Paragios, and Visvanathan Ramesh, "Background modeling and subtraction of dynamic scenes," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003, pp. 1305–1312.

[54] Jing Zhong and Stan Sclaroff, "Segmenting foreground objects from a dynamic textured background via a robust kalman filter," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003, pp. 44–50.

[55] Juan Carlos Niebles, Chih-Wei Chen, and Li Fei-Fei, "Modeling temporal structure of decomposable motion segments for activity classification," in *Computer Vision–ECCV 2010*, pp. 392–405. Springer, 2010.

[56] Pyry Matikainen, Martial Hebert, and Rahul Sukthankar, "Trajectons: Action recognition through the motion analysis of tracked features," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 514–521.

[57] Ross Messing, Chris Pal, and Henry Kautz, "Activity recognition using the velocity histories of tracked keypoints," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 104–111.

[58] Claudio Fanti, Lihi Zelnik-Manor, and Pietro Perona, "Hybrid models for human motion recognition," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. IEEE, 2005, vol. 1, pp. 1166–1173.

[59] Omar Oreifej and Zicheng Liu, "Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 716–723.

[60] Nazli Ikizler, Ramazan Gokberk Cinbis, and Pinar Duygulu, "Human action recognition with line and flow histograms," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE, 2008, pp. 1–4.

[61] Yang Song, Luis Goncalves, and Pietro Perona, "Unsupervised learning of human motion models," *Advances in Neural Information Processing Systems*, vol. 14, 2003.

[62] Cen Rao, Alper Yilmaz, and Mubarak Shah, "View-invariant representation and recognition of actions," *International Journal of Computer Vision*, vol. 50, no. 2, pp. 203–226, 2002.

[63] Naresh P Cuntoor and Rama Chellappa, "Epitomic representation of human activities," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8.

[64] Ju Sun, Xiao Wu, Shuicheng Yan, Loong-Fah Cheong, T-S Chua, and Jintao Li, "Hierarchical spatio-temporal context modeling for action recognition," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 2004–2011.

[65] Bruce D Lucas, Takeo Kanade, et al., "An iterative image registration technique with an application to stereo vision.," in *IJCAI*, 1981, vol. 81, pp. 674–679.

[66] Hirofumi Uemura, Seiji Ishikawa, and Krystian Mikolajczyk, "Feature tracking and motion compensation for action recognition.," in *BMVC*, 2008, pp. 1–10.

[67] Ju Sun, Yadong Mu, Shuicheng Yan, and Loong-Fah Cheong, "Activity recognition using dense long-duration trajectories," in *Multimedia and Expo (ICME), 2010 IEEE International Conference on*. IEEE, 2010, pp. 322–327.

[68] Heng Wang, Alexander Klaser, Cordelia Schmid, and Cheng-Lin Liu, "Action recognition by dense trajectories," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 3169–3176.

[69] Neil Johnson and David Hogg, "Learning the distribution of object trajectories for event recognition," *Image and vision computing*, vol. 14, no. 8, pp. 609–615, 1996.

[70] Wang-Chou Lu, Y-CF Wang, and Chu-Song Chen, "Learning dense optical-flow trajectory patterns for video object extraction," in *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*. IEEE, 2010, pp. 315–322.

[71] Alexander Kläser, Marcin Marszałek, Cordelia Schmid, and Andrew Zisserman, "Human focused action localization in video," in *Trends and Topics in Computer Vision*, pp. 219–233. Springer, 2012.

[72] Nadeem Anjum and Andrea Cavallaro, "Multifeature object trajectory clustering for video analysis," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 11, pp. 1555–1564, 2008.

[73] Alexandre Hervieu, Patrick Bouthemy, and J-P Le Cadre, "A statistical video content recognition method using invariant features on object trajectories," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 11, pp. 1533–1543, 2008.

[74] Michael Sapienza, Fabio Cuzzolin, and Philip Torr, "Learning discriminative space-time actions from weakly labelled videos," 2012.

[75] David J. Fleet and Allan D. Jepson, "Stability of phase information," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 15, no. 12, pp. 1253–1268, 1993.

[76] Paul Viola, Michael J Jones, and Daniel Snow, "Detecting pedestrians using patterns of motion and appearance," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003, pp. 734–741.

[77] Orit Kliper-Gross, Yaron Gurovich, Tal Hassner, and Lior Wolf, "Motion interchange patterns for action recognition in unconstrained videos," in *Computer Vision–ECCV 2012*, pp. 256–269. Springer, 2012.

[78] Gwenaëlle Piriou, Patrick Bouthemy, and Jian-Feng Yao, "Recognition of dynamic video contents with global probabilistic models of visual motion," *Image Processing, IEEE Transactions on*, vol. 15, no. 11, pp. 3417–3430, 2006.

[79] Shandong Wu, Omar Oreifej, and Mubarak Shah, "Action recognition in videos acquired by a moving camera using motion decomposition of lagrangian particle trajectories," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1419–1426.

[80] Mihir Jain, Hervé Jégou, and Patrick Bouthemy, "Better exploiting motion for better action recognition," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 2555–2562.

[81] Navneet Dalal, Bill Triggs, and Cordelia Schmid, "Human detection using oriented histograms of flow and appearance," in *Computer Vision–ECCV 2006*, pp. 428–441. Springer, 2006.

[82] David Cunado, Mark S Nixon, and John N Carter, "Automatic extraction and description of human gait models for recognition purposes," *Computer Vision and Image Understanding*, vol. 90, no. 1, pp. 1–41, 2003.

[83] Hae Jong Seo and Peyman Milanfar, "Action recognition from one example," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 5, pp. 867–882, 2011.

[84] Jingen Liu, Saad Ali, and Mubarak Shah, "Recognizing human actions using multiple features," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[85] Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt, "Sequential deep learning for human action recognition," in *Human Behavior Understanding*, pp. 29–39. Springer, 2011.

[86] Oscar Perez Concha, Richard Yi Da Xu, and Massimo Piccardi, "Compressive sensing of time series for human action recognition," in *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. IEEE, 2010, pp. 454–461.

[87] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman, "Discriminative learned dictionaries for local image analysis," in *Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8.

[88] Xue Mei and Haibin Ling, "Robust visual tracking using &# x2113; 1 minimization," in *International Conference on Computer Vision*. IEEE, 2009, pp. 1436–1443.

[89] Rodrigo Fernández and Emmanuel Viennet, "Face identification using support vector machines," 1999.

[90] Minh Hoai, Zhen-Zhong Lan, and Fernando De la Torre, "Joint segmentation and classification of human actions in video," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 3265–3272.

[91] Tianzhu Zhang, Jing Liu, Si Liu, Yi Ouyang, and Hanqing Lu, "Boosted exemplar learning for human action recognition," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 538–545.

[92] Alireza Fathi and Greg Mori, "Action recognition by learning mid-level motion features," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[93] Mohiuddin Ahmad, Irine Parvin, and Seong-Whan Lee, "Silhouette history and energy image information for human movement recognition," *Journal of Multimedia*, vol. 5, no. 1, pp. 12–21, 2010.

[94] Frank Moosmann, Bill Triggs, Frederic Jurie, et al., "Fast discriminative visual codebooks using randomized clustering forests," in *NIPS*, 2006, vol. 2, p. 4.

[95] Tsz-Ho Yu, Tae-Kyun Kim, and Roberto Cipolla, "Real-time action recognition by spatiotemporal semantic and structural forests," in *Proceedings of the British machine vision conference*, 2010, p. 56.

[96] Khai N Tran, Ioannis A Kakadiaris, and Shishir K Shah, "Modeling motion of body parts for action recognition.," in *BMVC*. Citeseer, 2011, pp. 1–12.

[97] Adrien Gaidon, Zaid Harchaoui, and Cordelia Schmid, "Actom sequence models for efficient action detection," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 3201–3208.

[98] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray, "Visual categorization with bags of keypoints," in *Workshop on statistical learning in computer vision, ECCV*, 2004, p. 22.

[99] Ana Paula Brandão Lopes, Eduardo Alves do Valle Jr, Jussara Marques de Almeida, and Arnaldo Albuquerque de Araújo, "Action recognition in videos: from motion capture labs to the web," *arXiv preprint arXiv:1006.3506*, 2010.

[100] Jingen Liu, Jiebo Luo, and Mubarak Shah, "Recognizing realistic actions from videos in the wild," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1996–2003.

[101] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. IEEE, 2006, vol. 2, pp. 2169–2178.

[102] Mikel D Rodriguez, Javed Ahmed, and Mubarak Shah, "Action mach a spatio-temporal maximum average correlation height filter for action recognition," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[103] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[104] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *CoRR*, vol. abs/1212.0402, 2012.

[105] Mousa Mojarrad, Mashallah Abbasi Dezfouli, and Amir Masoud Rahmani, "Feature extraction of human body composition in images by segmentation method," *World Academy of Science, Engineering and Technology*, 2008.

[106] Deqing Sun, Stefan Roth, and Michael J Black, "Secrets of optical flow estimation and their principles," in *IEEE Conference onComputer Vision and Pattern Recognition CVPR*. IEEE, 2010, pp. 2432–2439.

[107] Jan J Koenderink, "The structure of images," *Biological cybernetics*, vol. 50, no. 5, pp. 363–370, 1984.

[108] Paul R Beaudet, "Rotationally invariant image operators," in *Proceedings of the International Joint Conference on Pattern Recognition*, 1978, pp. 579–583.

[109] Ruxandra Tapu and Titus Zaharia, "High level video temporal segmentation," in *Advances in Visual Computing*, pp. 224–235. Springer, 2011.

[110] Daniel Dementhon and David Doermann, "Video retrieval of near-duplicates using κ-nearest neighbor retrieval of spatio-temporal descriptors," *Multimedia Tools and Applications*, , no. 3, pp. 229–253, 2006.

[111] Ronald Poppe, "A survey on vision-based human action recognition," *Image and vision computing*, vol. 28, no. 6, pp. 976–990, 2010.

[112] Tong Yubing, Faouzi Alaya Cheikh, Fahad Fazal Elahi Guraya, Hubert Konik, and Alain Trémeau, "A spatiotemporal saliency model for video surveillance," *Cognitive Computation*, vol. 3, no. 1, pp. 241–263, 2011.

[113] Zhuolin Jiang, Zhe Lin, and Larry S Davis, "Recognizing human actions by learning and matching shape-motion prototype trees," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 3, pp. 533–547, 2012.

[114] Eleonora Vig, Michael Dorr, and David Cox, "Space-variant descriptor sampling for action recognition based on saliency and eye movements," in *Computer Vision–ECCV 2012*, pp. 84–97. Springer, 2012.

[115] Mikel D. Rodriguez, Javed Ahmed, and Mubarak Shah, "Action mach: a spatio-temporal maximum average correlation height filter for action recognition," in *In Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2008.

[116] Marcin Marszalek, Ivan Laptev, and Cordelia Schmid, "Actions in context," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on.* IEEE, 2009, pp. 2929–2936.

[117] Wanqing Li, Zhengyou Zhang, and Zicheng Liu, "Expandable data-driven graphical modeling of human actions based on salient postures," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 11, pp. 1499–1510, 2008.

[118] Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani, and Ronen Basri, "Actions as space-time shapes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 12, pp. 2247–2253, 2007.

[119] Fabio Martínez, Antoine Manzanera, and Eduardo Romero, "A motion descriptor based on statistics of optical flow orientations for action classification in video-surveillance," in *Multimedia and Signal Processing*, pp. 267–274. Springer, 2012.

[120] Yan Ke, Rahul Sukthankar, and Martial Hebert, "Efficient visual event detection using volumetric features," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on.* IEEE, 2005, vol. 1, pp. 166–173.

[121] Alexei A Efros, Alexander C Berg, Greg Mori, and Jitendra Malik, "Recognizing action at a distance," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003, pp. 726–733.

[122] Shandong Wu, YF Li, and Jianwei Zhang, "A hierarchical motion trajectory signature descriptor," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. IEEE, 2008, pp. 3070–3075.

[123] Jianyu Yang, YF Li, and Keyi Wang, "A new descriptor for 3d trajectory recognition via modified cdtw," in *Automation and Logistics (ICAL), 2010 IEEE International Conference on*. IEEE, 2010, pp. 37–42.

[124] Josef Sivic and Andrew Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Ninth IEEE International Conference on Computer Vision*. IEEE, 2003, pp. 1470–1477.

[125] Jianguo Zhang, Marcin Marszałek, Svetlana Lazebnik, and Cordelia Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study," *International journal of computer vision*, vol. 73, no. 2, pp. 213–238, 2007.

[126] Jasper RR Uijlings, Arnold WM Smeulders, and Remko JH Scha, "Real-time visual concept classification," *Multimedia, IEEE Transactions on*, vol. 12, no. 7, pp. 665–681, 2010.

[127] Michalis Vrigkas, Vasileios Karavasilis, Christophoros Nikou, and Ioannis A Kakadiaris, "Matching mixtures of curves for human action recognition," *Computer Vision and Image Understanding*, vol. 119, pp. 27–40, 2014.

[128] Adriana Kovashka and Kristen Grauman, "Learning a hierarchy of discriminative space-time neighborhood features for human action recognition," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2046–2053.

[129] Quoc V Le, Will Y Zou, Serena Y Yeung, and Andrew Y Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 3361–3368.

[130] Nazli Ikizler-Cinbis and Stan Sclaroff, "Object, scene and actions: Combining multiple features for human action recognition," in *Computer Vision–ECCV 2010*, pp. 494–507. Springer, 2010.

[131] Thomas Brox and Jitendra Malik, "Object segmentation by long term analysis of point trajectories," in *Computer Vision–ECCV 2010*, pp. 282–295. Springer, 2010.

[132] Adrien Gaidon, Zaid Harchaoui, Cordelia Schmid, et al., "Recognizing activities with cluster-trees of tracklets," in *BMVC*, 2012.

[133] Ling Shao, Ling Ji, Yan Liu, and Jianguo Zhang, "Human action segmentation and recognition via motion and shape analysis," *Pattern Recognition Letters*, vol. 33, no. 4, pp. 438–445, 2012.

[134] Luca Zappella, Xavier Lladó, and Joaquim Salvi, "Motion segmentation: A review," in *Proceedings of the 2008 conference on Artificial Intelligence Research and Development: Proceedings of the 11th International Conference of the Catalan Association for Artificial Intelligence*. IOS Press, 2008, pp. 398–407.

[135] Luca Zappella, Xavier Lladó, and Joaquim Salvi, "New trends in motion segmentation," *Pattern Recognition*, pp. 31–46, 2009.

[136] Saad Ali and Mubarak Shah, "A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–6.

[137] Daniel Cremers and Stefano Soatto, "Motion competition: A variational approach to piecewise parametric motion segmentation," *International Journal of Computer Vision*, vol. 62, no. 3, pp. 249–265, 2005.

[138] Thomas Brox, Mikaël Rousson, Rachid Deriche, and Joachim Weickert, "Colour, texture, and motion in level set based segmentation and tracking," *Image and Vision Computing*, vol. 28, no. 3, pp. 376–390, 2010.

[139] Berthold K Horn and Brian G Schunck, "Determining optical flow," in *1981 Technical Symposium East*. International Society for Optics and Photonics, 1981, pp. 319–331.

[140] Jean-Yves Bouguet, "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm," *Intel Corporation*, vol. 5, 2001.

[141] Frederic Jurie and Bill Triggs, "Creating efficient codebooks for visual recognition," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*. IEEE, 2005, vol. 1, pp. 604–610.

[142] Li Fei-Fei and Pietro Perona, "A bayesian hierarchical model for learning natural scene categories," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. IEEE, 2005, vol. 2, pp. 524–531.

[143] Eric Nowak, Frédéric Jurie, and Bill Triggs, "Sampling strategies for bag-of-features image classification," in *Computer Vision–ECCV 2006*, pp. 490–503. Springer, 2006.

[144] Jean-Marc Odobez and Patrick Bouthemy, "Robust multiresolution estimation of parametric motion models," *Journal of visual communication and image representation*, vol. 6, no. 4, pp. 348–365, 1995.

[145] Do Hang Nga and Keiji Yanai, "A spatio-temporal feature based on triangulation of dense surf," in *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*. IEEE, 2013, pp. 420–427.

[146] John Canny, "A computational approach to edge detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, , no. 6, pp. 679–698, 1986.

[147] James D Foley, Andries Van Dam, Steven K Feiner, John F Hughes, and Richard L Phillips, *Introduction to computer graphics*, vol. 55, Addison-Wesley Reading, 1994.

[148] Roberto Brunelli, "Template matching techniques in computer vision," 2008.

[149] OV Murthy and Roland Goecke, "Ordered trajectories for large scale human action recognition," in *International Conference on Computer Vision Workshops (ICCVW)*. IEEE, 2013, pp. 412–419.

[150] Feng Shi, Robert Laganiere, Emil Petriu, and Haiyu Zhen, "Lpm for fast action recognition with large number of classes," in *THUMOS: ICCV Workshop on Action Recognition with a Large Number of Classes. Notebook paper*, 2013.

[151] Svebor Karaman, Lorenzo Seidenari, Andrew D Bagdanov, and Alberto Del Bimbo, "L1-regularized logistic regression stacking and transductive crf smoothing for action recognition in video," in *ICCV Workshop on Action Recognition with a Large Number of Classes*, 2013.

[152] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[153] Karen Simonyan and Andrew Zisserman, "Two-stream convolutional networks for action recognition in videos," *arXiv preprint arXiv:1406.2199*, 2014.