Learning human actions in video

Alexander Klaser

UNIVERSITE DE GRENOBLE

N° attribué par la bibliothèque

THÈSE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITE DE GRENOBLE

Spécialité : Mathématiques et Informatique

préparée au Laboratoire Jean Kuntzmann

dans le cadre de l'École Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique

présentée et soutenue publiquement

par

Alexander Kläser

le 31 juillet 2010

Apprentissage pour la reconnaissance d'actions humaines en vidéo

Learning human actions in videos

Directeur de thèse : Cordelia Schmid

JURY

M. James. L. Crowley	Président
M. Martial Hebert	Rapporteur
M. François Brémond	Rapporteur
Mme. Cordelia Schmid	Examinateur
M. Ivan Laptev	Examinateur

Trust only movement.
Life happens at the level of events, not of words.
Trust movement.

Alfred Adler, Austrian psychologist

Contents

Introduction

Au cours des dix dernières années, les ordinateurs et l'Internet ont influencé nos vies d'une manière fondamentale. Les ordinateurs effectuent des calculs intenses et répétitifs sur des bases de données très larges. De cette manière ils ont étendu nos possibilités de travailler et communiquer. Avec les progrès technologiques récents, les données vidéo sont devenues de plus en plus accessibles et jouent un rôle de plus en plus important dans notre vie quotidienne. Aujourd'hui, même des matériels éléctroniques couramment utilisés, tels que les ordinateurs portables, les téléphones mobiles et les appareils photo numériques, permettent de créer des vidéos. Simultanément, un accès plus rapide à l'Internet et des capacités de stockage de plus en plus élevées permettent de publier et partager des vidéos de manière instantanée. Par exemple, 36 millions d'internautes allemands (44% de la population) ont regardé plus de 6 milliards de vidéos en ligne en août 2009. Par rapport à août 2008, cela représente une augmentation de 38%. Un autre example est le nombre de vidéo téléchargé sur YouTube par minute qui est passé de six heures en 2007 à 20 heures en 2009—soit une augmentation d'environ 330% sur deux ans.

Cependant, malgré l'importance croissante des données vidéo, les possibilités de les analyser d'une façon automatisée sont plutôt limitées. Les systèmes de vision par ordinateur sont loin d'être à l'hauteur de la vision humaine. Par exemple, la recherche de vidéos dans les archives de bases de données à grande échelle est actuellement uniquement possible grace à l'annotation manuelle par des humains. Des moteurs de recherche pour vidéo, tels que YoutTube, reposent essentiellement sur des données textuelles, telles que la description ou des étiquettes, afin de récupérer des vidéos pertinentes. Un autre exemple est le domaine de la vidéo-surveillance. Jusqu'à aujourd'hui, la ville de Londres a installé environ 1 million de caméras vidéo. En ce qui concerne un rapport interne, il a été souligné que "les caméras de surveillance conduisent à des dépenses massives avec une efficacité minimale".

Ces exemples montrent qu'il existe une forte demande pour des systèmes de vision par ordinateur afin de pouvoir traiter des données vidéo d'une manière automatisée. Ces technologies de vision par ordinateur auront vraisemblablement un fort impact sur notre avenir.

Enoncé du problème

Cette dissertation se concentre sur le problème de la reconnaissance d'actions simples et génériques dans des vidéos réalistes, tels que les films, les vidéos sur Internet et les vidéos de surveillance. La figure 1.1 illustre différentes actions dans des films, et la figure 1.2 montre des détections d'actions examplaires que nous sommes en mesure de localiser dans des films réalistes.

Contributions

La première partie de notre travail se base sur des primitives locales pour la classification d'action. Pour cela, les approches existantes sont étudiées et de nouvelles méthodes élaborées. La deuxième partie présente une nouvelle méthode pour la localisation d'action dans des vidéos. Ci-dessous, nous résumons nos contributions:

- Nous introduisons un nouveau descipteur local pour des séquences d'images basé sur les histogrammes d'orientations de gradients spatio-temporels (HOG3D). Nous proposons une approche efficace afin de calculer des gradients 3D à des échelles arbitraires et nous dévéloppons un algorithme pour la quantification d'orientations 3D basé sur des polyèdres réguliers. Les paramètres de notre descripteur sont évalués en profondeur et ils sont optimisés pour la reconnaissance d'actions en utilisant la représentation sac-de-mots. Ce travail est présenté dans le chapitre 3. Le travail a été effectué en collaboration avec Marcin Marszałek et il a été publié dans Kläser et al. [2008].
- Nous évaluons et comparons plusieurs méthodes existantes de détection et description de characteristiques locales pour la reconnaissance d'action dans des vidéos. En total, quatre détecteurs et six descripteurs sont étudiés en utilisant une approche standard par sac-de-mots avec une machine à vecteurs de support (SVM) comme classifieur. Nous évaluons la performance sur un total de 25 classes d'action réparties sur trois bases de données avec différents niveaux de difficulté. Cette contribution est discuté en détail dans le chapitre 4. Elle a été publiée dans [Wang et al., 2009] en collaboration avec Wang Heng et Mohammed Muneeb Ullah.
- Nous développons un nouveau descripteur pour la reconnaissance d'action basé sur des trajectoires de points locaux pertinents. Contrairement aux méthodes existantes, nous étendons la description d'une trajectoire avec l'information sur l'apparence et le mouvement de son entourage. Pour cela, nous introduisons également un nouveau descripteur basée sur des histogrammes de frontière de mouvement. Les paramètres de ce descripteur sont étudiés et optimisées pour la tâche de reconnaissance d'action dans des vidéos réalistes. Ce travail a été effectué en collaboration avec Heng Wang et il est détaillée dans le chapitre 5.
- Nous étudions la combinaison de la représentation par sac-de-mots avec la localisation de personnes et nous quantifions ses améliorations pour la reconnaissance d'action. Pour ce faire, nous évaluons d'abord le gain en performance par la réduction de l'attention uniquement sur des personnes dans les vidéos. Puis, nous montrons comment intégrer des contraintes spatiales dans le modèle sac-de-mots pour améliorer la classification. Ce travail est détaillé dans le chapitre 6.
- Nous proposons une nouvelle approche afin de détecter et localiser des actions humaines dans des films. Pour cela, nous développons un détecteur de personnes adapté à ce type de données et étant en mesure de faire face à un large éventail de postures, articulations, mouvements et points de vue de caméra. Pour la représentation d'action, nous introduisons un descripteur spatio-temporel qui est adapté à la détection de personne. Des résultats sont montrés pour les actions "boire", "fumer", "téléphoner" et "se lever". Cette contribution est présentée dans le chapitre 7. Elle a été un travail en collaboration avec Marcin Marszałek et elle a été publiée dans [Kläser et al., 2010].

1

Introduction

Contents

Over the past decade, computers and world-wide networks have influenced our lives tremendously. Computers perform repetitive and data intensive computational tasks and extend fundamentally our possibilities to communicate. Along with recent technological advances of computers in general, video data has become more and more accessible and plays an increasingly important role in our everyday life. Today, even commonly used consumer hardware, such as notebooks, mobile phones, and digital photo cameras, allow to create videos. At the same time, faster internet access and growing storage capacities enable to direcly publish and share videos with others. For example, 36 million German internet users (44% of the population) watched more than 6 billion videos online in August 2009 [1]. Compared to August 2008, this is an increase of 38%. The amount of video uploaded to YouTube every minute increased from six hours in mid-2007 to 20 hours in May 2009 [2], i.e., an increase of about 330% over two years.

However, despite the increasing importance of video data, the possibilities to analyze it in an automated fashion are rather limited. Computer vision systems are far behind the capabilities of human vision. For instance, video search in large scale databases archives is currently only feasible with costly manual annotation. Web search engines commonly rely mainly on textual data, such as descriptions or tags, in order to retrieve relevant videos.

Another example are surveillance applications. Up to today, the city of London has installed about 1 million closed-circuit television (CCTV) cameras at the cost of approximately 200 million British pounds. However, in 2008, surveillance cameras helped to solve only one crime per 1,000 cameras [3]. With respect to an internal report, it has been pointed out that "CCTV leads to massive expense and minimum effectiveness" [4]. Research

1. Source: http://www.comscore.com
2. Source: http://youtube-global.blogspot.com
3. Source: http://news.bbc.co.uk
4. Source: http://www.telegraph.co.uk

Figure 1.1: Sample actions in videos.

commissioned by the Home Office [5] concluded that CCTV virtually has not helped cutting down crime, it showed to be most effective for preventing vehicle crimes in car parks. In fact, given the vast amount of video data, one major bottleneck is the necessity to acquire the data (analog cameras store data on video tapes) and analyze it manually.

A further application area is computer games, for which video analysis has gained a lot of attention as sophisticated human-computer interface. One on-going project is Microsoft's Project Natal [6]. The project's framework allows for full-body 3D motion capture, facial recognition, voice recognition, and acoustic source localization. This is achieved by combining information from several sensors: a video camera, a depth sensor (based on infrared patterns), and a multi-array microphone. This allows users to play video games without controller devices and to interact in a virtual world using their full bodies in a natural way.

Motion capturing of human actors has evolved to a de facto standard for character animation in computer animated movies as well as for movie special effects [7]. Otherwise, human motion analysis can also play an important role in medical applications (e.g., rehabilitation, medical examination) as well as in the analysis and optimization of movements of sport athletics or in dance choreography.

5. The Home Office is the United Kingdom government department responsible for immigration control, security and order, thus also including the police.

6. http://www.xbox.com/projectnatal/

7. http://www.motioncapturesociety.com

Figure 1.2: Sample detections of particular actions in common movies (*cf.* chapter 7).

These examples show that there is a large demand for computer vision systems to understand and process video in an automated fashion. They also illustrate that computer vision technologies have a high potential to influence our future.

1.1 Problem statement

This dissertation focuses on the problem of *action recognition* in realistic video material, such as movies, internet and surveillance videos. Figure 1.1 illustrates various actions in movies, and figure 1.2 shows sample detections of actions that we are able localize in challenging movie material (*cf.* chapter 7). In order to be more precise about our goal, we clarify the meaning of *action* and *action recognition* by an analogy to languages.

Human language is composed of sentences which are themselves structured with *subjects*, *verbs*, and *objects*. In order to describe the visual content of a video in an automatic fashion, a structure similar to that of a language is necessary. From an algorithmic point of view, this translates to the detection of (a) *subjects* (or *actors*) which most commonly are humans; (b) *objects* which can be other humans, they can be objects, and they also include environments in which the *subject* is operating; (c) *verbs* which describe *actions* of the subject as well as *interactions* between subjects and objects.

In this sense, an *action* can be precisely localized in a short interval in time, yet it can also refer to an event that lasts for a rather long time period. For clarification, an action taxonomy can be defined as in [Moeslund et al., 2006]: *action primitive* (or *movement*), *action*, and *activity*. An *action primitive* describes a basic and atomic motion entity out of which *actions* are built. An *activity* is a set of several *actions*. *Activities* can be understood as larger scale events that often depend on the context and the environment in which the action happens.

Figure 1.3: Motion capture for movie production in a studio (courtesy of Sony Pictures Imageworks).

Considering the example of playing tennis, "Playing tennis" itself can be seen as an *activity*. It involves several *actions*, such as "serving", "returning ball", or "taking a break". "Serving" could be split into the *action primitives* "throwing the ball up", "swinging racket back", and "hitting the ball". A different *activity* such as "drinking coffee" might involve *actions* including, e.g., "drinking", "filling cup", "taking cup", "putting back cup", and maybe also other *actions* like "smoking", "reading", "talking". "Drinking" could be decomposed into *action primitives* such as "raising cup to mouth", "drinking from cup", "lowering cup".

Interestingly, some *action primitives* are intrinsically linked to an object. Only "raising arm" alone is not sufficient to be part of the *action* "drinking". Instead of drinking, one can also raise the arm towards the mouth in order to smoke. Therefore, "raising cup with arm" is a more appropriate term as *action primitive* for "drinking".

Apart from *actions* and *action primitives* that are closely related to a particular *activity* (e.g., "returning ball" for "playing tennis"), there is a set of rather *generic actions* or *action primitives* which are independent of the context. Entities of this set include "walking", "running", "jumping", "standing up", "sitting down", "shaking hands", "hugging a person", "drinking", "smoking" etc.

In this dissertation, we focus on the detection of visible low-level *action primitives* and *actions* of a rather *generic* type. Figure 1.1 gives some examples. In the remainder of this work, we will refer to this task as *action recognition*.

1.2 Context

Numerous works and methods have been proposed in the past within the field of action and activity recognition. Since recognizing actions in videos is a challenging problem, a lot of approaches have considered simplified settings. For a broader view, we discuss existing works according to the type of video data that they employ. For this, we distinguish the categories "controlled video data", "constrained video data", "uncontrolled video data".

Controlled video data. Controlled video data is acquired in a way to facilitate its automated processing. For instances, markers can be attached on human actors for detecting joints and limbs, e.g., Medina-Carnicer et al. [2009], Li et al. [2008] (see figure 1.3);

Figure 1.4: Action recognition in a multi-camera setup (courtesy of Weinland et al. [2007]).

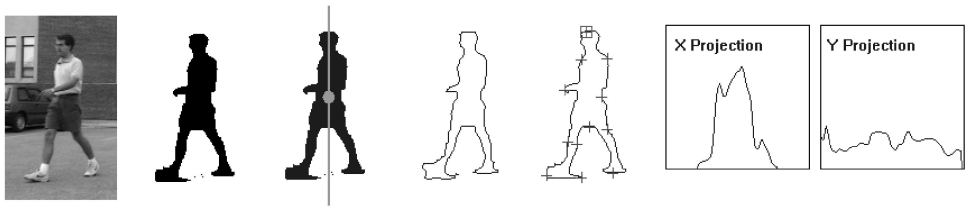


Figure 1.5: Analysis of shape masks obtained via background substraction for a video surveillance system (courtesy of Haritaoglu et al. [2000]).

lighting conditions can be controlled to better detect markers and human bodies; multiple cameras can be placed in order to cover a necessary range of view points for 3D reconstruction, e.g., Fleuret et al. [2008], Weinland et al. [2007] (see figure 1.4). A prominent example are commercial high-end motion capture systems for film productions. These use extensively optical markers and a large set of cameras to record motion up to the level of facial gestures and finger movements, e.g., [Havaldar, 2006].

Constrained video data. Applications that operate on constrained video data are able to influence environmental parameters to a limited degree. This is the case for commercial video game platforms based on visual interfaces, such as the Project Natal [Microsoft, 2009]; certain assumptions can be made, e.g., a single person fully visible or favorable lighting conditions. However, a certain robustness is necessary with respect to other visual conditions (e.g., varying size of humans, different clothing, motion variability) that cannot be influenced.

Another very common application area is video surveillance [Hu et al., 2004, Senior, 2009] for which camera placement and parameters are fixed and known, e.g., Fleuret et al. [2008]. Since cameras are in general static, techniques such as background substraction are commonly applied to compute human shape masks. These masks are then further analyzed to recognize human behavior and actions [Haritaoglu et al., 2000] (see figure 1.5). Nevertheless, certain aspects cannot be controlled: the clothes that humans wear, the way they move, or weather and lighting conditions.

In this sense, we also consider an environment with a rather limited set of expected actions – such as dancing, ballet, or sports [Urtasun et al., 2006, Ramasso et al., 2009] – to belong to this category of constrained data.

Uncontrolled video data. Uncontrolled video data is recorded under conditions which cannot be influenced. This is the case for, e.g., TV and cinema style movie data, sports broadcasts, music videos, or personal amateur clips. Only very few assumptions, if any, of a rather general nature can be made, such as humans are present and relative well visible. The main challenges for this more realistic data include changes of viewpoint, scale, and lighting conditions, partial occlusion of humans and objects, cluttered backgrounds, abrupt movement etc.

Earlier work on human action recognition in video [Bobick and Davis, 2001, Blank et al., 2005, Efros et al., 2003, Dollár et al., 2005, Niebles et al., 2006, Jhuang et al., 2007, Wong and Cipolla, 2007, Scovanner et al., 2007, Schindler and van Gool, 2008, Weinland and Boyer, 2008, Willems et al., 2008] employed image data with mainly static cameras, simple and homogeneous backgrounds, and humans fully visible. The most popular datasets are the *KTH* [Schüldt et al., 2004] and the *Weizmann* [Blank et al., 2005] actions dataset, *cf.* sections 2.2.2 and 2.2.1, respectively. This enabled to explore classifiers with variations in actors and actions. However, it did not take into account added complexity for more realistic data, such as movies, music videos, or personal amateur clips.

With recently published action datasets based on generic movie data [Laptev and Perez, 2007, Laptev et al., 2008, Marszałek et al., 2009], YouTube video sequences [Liu et al., 2009], or sports broadcasts [Rodriguez et al., 2008], the field of action recognition has in general moved towards less controlled and much more challenging type of data. For this task, methods that use local features [Laptev et al., 2008, Mikolajczyk and Hirofumi, 2008, Marszałek et al., 2009, Liu et al., 2009, Willems et al., 2009, Gilbert et al., 2009, Han et al., 2009] have shown excellent results. A common representation used in the literature is bag-of-features (*cf.* section 2.1.3) in which video sequences are represented as occurrence histograms of quantized local features.

Some approaches [Laptev and Perez, 2007, Ke et al., 2007a, Hu et al., 2009, Willems et al., 2009] have also addressed the problem of localizing actions spatially as well as temporally in more realistic video settings. As opposed to action classification where sequences of pre-defined temporal extent are classified as belonging to one of n action classes, action localization is a much more difficult task.

1.3 Main contributions

The goal of this dissertation is the recognition of rather simple, low-level actions in uncontrolled, realistic video data. The first part of our work is based on local features which are employed for action classification. For this, existing approaches to describe local information in videos are investigated and new methods are developed. The second part of this work introduces a new method for action localization in videos. To this end, we develop a human detection system as well as a method to describe and localize actions in Hollywood-style movies.

To summarize, we provide the following main contributions:

- We introduce a novel local desciptor for image sequences based on histograms of spatio-temporal gradient orientations (HOG3D). Our approach is based on a memory-efficient algorithm to compute 3D gradients for arbitrary scales and a generic 3D orientation quantization based on regular polyhedrons. Descriptor parameters are evaluated in depth and optimized for action recogition using bag-of-features representation. This joint work with Marcin Marszałek was published in [Kläser et al., 2008] and is presented in chapter 3.
- We evaluate and compare several existing local space-time features for action recognition. In total, four different feature detectors and six local feature descriptors are investigated using a standard bag-of-features SVM approach. We investigate their performance on a total of 25 action classes distributed over three datasets with varying difficulty. This contribution was published in [Wang et al., 2009] in collaboration with Heng Wang and Muhammad Muneeb Ullah. It is discussed in detail in chapter 4.
- We develop a novel descriptor for action recognition based on local feature trajectories. Contrary to existing methods, we extend the trajectory descriptor with appearance and motion information in the local neighborhood of the trajectory. For this, we also introduce a new descriptor based on motion boundary histograms. Descriptor parameters are studied and optimized for the task of action recognition in realistic video settings. This joint work with Heng Wang is detailed in chapter 5.
- We investigate combining bag-of-features models with person localization and quantify the improvements for action recognition. For this, first, we evaluate the gain in performance by narrowing down the attention to human actors. Second, we show how to incorporate spatial constraints in BoF models to improve accuracy for action recognition. This work is detailed in chapter 6.
- We propose a novel human-centric approach to detect and localize human actions in Hollywood-style movie data. To achieve this, we develop a human upper-body detector and tracker for movie data which is able to cope with a wide range of postures, articulations, motions and camera viewpoints. For the action representation, we introduce a spatio-temporal HOG3D based descriptor adapted to human tracks. Results are included for the actions “drinking”, “smoking”, “phoning”, and “standing-up”. This contribution was joint work with Marcin Marszałek. It was published in [Kläser et al., 2010] and is presented in chapter 7.

État de l'art et base de données

Ce chapitre passe en revue l'état de l'art des méthodes de reconnaissance d'action dans des vidéos réalistes. Nous répartissons les travaux existants en trois catégories:

- Les *méthodes basées sur un modèle du corps humain* (section 2.1.1) emploient un modèle 3D (ou 2D) sur les parties du corps humain. La reconnaissance d'action s'effectue alors en utilisant des informations sur le positionnement et le mouvement des parties du corps.
- Les *méthodes holistiques* (section 2.1.2) utilisent la connaissance sur la localisation des personnes dans la vidéo. Par conséquent, elles apprennent un modèle d'action à partir des mouvements caractéristiques du corps entier sans aucune notion de parties du corps.
- Les *méthodes basées sur des caractéristiques locales* (section 2.1.3) utilisent uniquement des descripteurs locaux de vidéo. Aucune connaissance préalable sur le positionnement des personnes dans la vidéo ou sur celui de leurs membres n'est utilisée.

En outre, nous présentons dans ce chapitre des bases de données pour la reconnaissance d'action utilisées dans cette thèse (sections 2.2.1-2.2.5). Au-delà de leur description, nous comparons également les meilleurs résultats qui ont été publiés dans la littérature.

2

Related work and datasets

Contents

2.1 Related work

This section reviews the state-of-the-art methods for action recognition in realistic, uncontrolled video data. To this end, we structure existing works into three categories:

- *Human model based methods* (section 2.1.1) employ a full 3D (or 2D) model of human body parts, and action recognition is done using information on body part positioning as well as movements.
- *Holistic methods* (section 2.1.2) use knowledge about the localization of humans in video and consequently learn an action model that captures characteristic, global body movements without any notion of body parts.
- *Local feature methods* (section 2.1.3) are entirely based on descriptors of local regions in a video, no prior knowledge about human positioning nor of any of its limbs is given.

Surveys on generic action and activity recognition as well as motion analysis and body tracking include Weinland et al. [2010], Poppe [2010], Moeslund et al. [2006], Buxton [2003], Moeslund and Granum [2001], Gavrila [1999], Aggarwal and Cai [1999]. Furthermore, Hu et al. [2004] present a survey for video surveillance, and Turaga et al. [2008] review the state-of-the-art for high level activity analysis. Most relevant in our context are the surveys by Weinland et al. [2010] and Poppe [2010] which focus on the recognition of actions and action primitives.

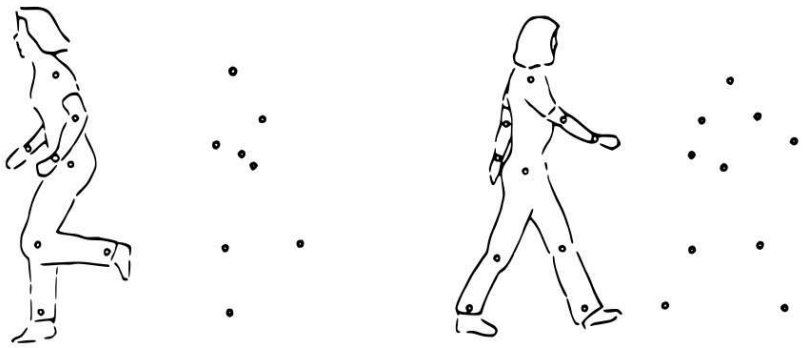


Figure 2.1: Examples of motions with a few moving light displays (MLD) attached to the human body (courtesy of Johansson [1973]).

2.1.1 Human model based methods

Human model based methods recognize actions by employing information such as body part positions and movements. A significant amount of research [Moeslund et al., 2006] is devoted to action recognition using trajectories of joint positions, body parts, or landmark points on the human body with or without a prior model of human kinematics, e.g., [Ali et al., 2007, Parameswaran and Chellappa, 2006, Yilmaz and Shah, 2005b]. Approaches in this field can be related to psychophysical work on visual interpretation of biological motion [Johansson, 1973] which shows that humans are able to recognize actions soley from the motion of a few moving light displays (MLD) attached to the human body (see figure 2.1).

The localization of body parts in movies has been investigated in the past (e.g., Ramanan et al. [2007], Ferrari et al. [2008]) and some works have shown impressive results. However, the detection of body parts is a difficult problem in itself, and results especially for the case of realistic and less constrained video data remain limited in their applicability. Some recent approaches that are able to provide more robust results (e.g., Agarwal and Triggs [2006], Urtasun et al. [2006]), use strong prior knowledge by assuming particular motion patterns in order to improve tracking of body parts. However, this also limits their application to action recognition.

2.1.2 Holistic methods

Holistic methods do not require the localization of body parts. Instead, global body structure and dynamics are used to represent human actions. Polana and Nelson [1994] refered to this approach as "getting your man without finding his body parts". The key idea is that, given a region of interest centered on the human body, global dynamics are discriminative enough to characterize human actions.

Compared to approaches that explicitely use a kinematic model or information about body parts, holistic representations are much simpler since they only model global motion

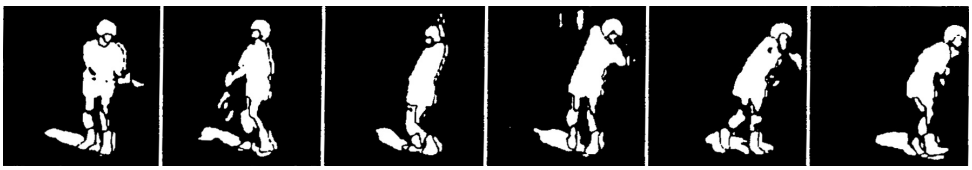


Figure 2.2: Shape masks for recognizing tennis actions (courtesy of Yamato et al. [1992]).

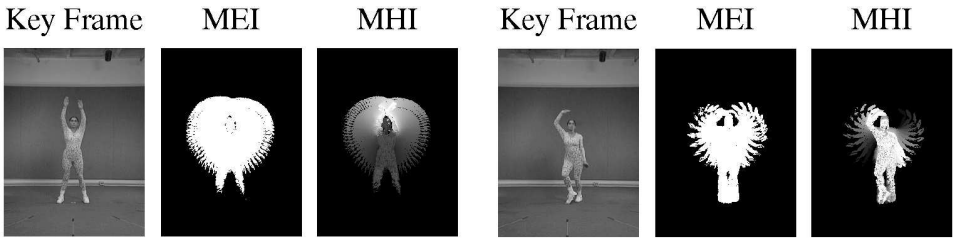


Figure 2.3: Shape masks from difference images for computing motion history images (MHI) and motion energy images (MEI) (courtesy of Bobick and Davis [2001]).

and appearance information. Therefore their computation is in general more efficient as well as robust. This aspect is especially important for realistic videos in which background clutter, camera ego-motion, and occlusion render the localization of body parts particularly difficult.

In general, holistic approaches can be roughly divided into two categories. The first category employs shape masks or silhouette information, stemming from background substraction or difference images, to represent actions. The second category is mainly based on shape and optical flow information.

Shape mask and silhouette based methods

Several approaches for action recognition use human *shape masks* and *silhouette* information to represent the human body and its dynamics. Yamato et al. [1992] are among the first to propose silhouette images (*cf.* figure 2.2). Their representation computes a grid over the silhouette and computes for each cell the ratio of foreground to background pixels. The grid representations are quantized into a vocabulary, and tennis actions are then learned as sequences of "words" using hidden Markov models (HMM) [Rabiner, 1989].

Bobick and Davis [2001] use shape masks from difference images to detect human actions. As action representation, the authors employ so-called *motion energy images* (MEI) and *motion history images* (MHI), as illustrated in figure 2.3. More precisely, MEIs are binary masks that indicate regions of motion, and MHIs weight these regions according to the point in time when they occurred (the more recent, the higher the weight). This approach is the first to introduce the idea of temporal templates for action recognition.

Sullivan and Carlsson [2002] detect tennis forehand strokes by matching a set of hand-drawn key postures together with annotated body joint positions to edge information in a video sequence. Positions of joints are then tracked between the keyframes using silhouette

Figure 2.4: Space-time volumes for action recognition based on silhouette information (courtesy of Blank et al. [2005]).

information of the tennis player. This approach allows to infer positions of body parts which can be applied to, e.g., 3D animation.

An action model based on space-time shapes from silhouette information is introduced by Blank et al. [2005], Gorelick et al. [2007]. Silhouette information is computed using background substraction. Figure 2.4 illustrates some examples of space-time shapes. The authors use properties of the solution to the Poisson equation to extract features such as local saliency, action dynamics, shape structure and orientation. Chunks of 10 frames length are then described by a high-dimensional feature vector. During classification, these chunks are matched in a sliding window fashion to space-time shapes in test sequences.

Another work that uses space-time shapes of humans, is proposed by Yilmaz and Shah [2005a]. Spatio-temporal shapes are obtained from contour information using background substraction, similar to Blank et al. [2005]. For a robust representation, actions are then represented by sets of characteristic points (such as saddle, valley, ridge, peak, pit points) on the surface of the shape. In order to recognize actions, the authors propose to match spatio-temporal shapes by computing a homography using point-to-point correspondences.

Weinland and Boyer [2008] introduce an orderless representation for action recognition using a set of silhouette exemplars. Action sequences are represented as vectors of minimum distance between silhouettes in the set of exemplars and in the sequence. Final classification is done using Bayes classifier with Gaussians to model action classes. In addition to silhouette information, the authors also employ the Chamfer distance measure to match silhouette exemplars directly to edge information in test sequences.

Foreground shape masks based on motion information in chunks of video data are employed by Zhang et al. [2008], *cf.* figure 2.5. A Motion Context descriptor is computed over consistent regions of motion by using a polar grid. Each cell in the grid is described with a histogram over quantized SIFT [Lowe, 2004] features. The final descriptor for a sequence is a sum over all chunk descriptors. For classification, support vector machines (SVM) [Burges, 1998] and different models for probabilistic latent semantic analysis (PLSA) [Hofmann, 1999] are employed.

Silhouettes are also a popular representation for surveillance applications [Haritaoglu et al., 2000, Hu et al., 2004, Senior, 2009]. Since cameras are in general static, background substraction techniques can be employed to compute silhouette information. As illustrated

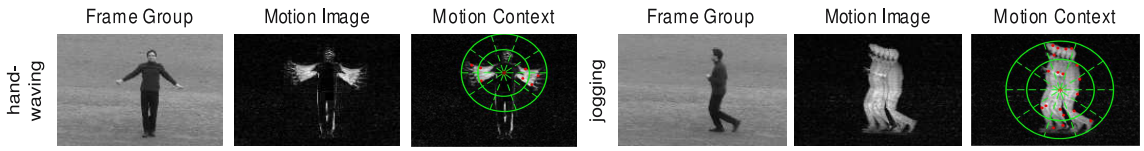


Figure 2.5: Illustration of the Motion Context descriptor for the actions hand waving and jogging: motion images are computed over groups of images; the Motion Context descriptor is computed over consistent regions of motion (courtesy of Zhang et al. [2008]).

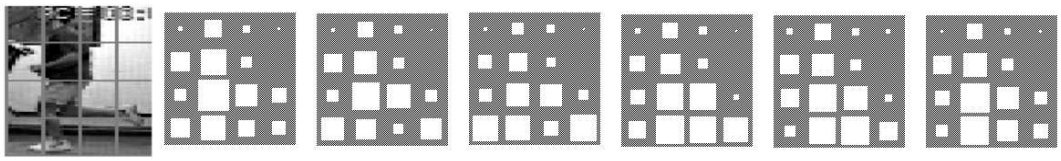


Figure 2.6: A human centric grid of optical flow magnitudes to describe actions (courtesy of Polana and Nelson [1994]).

in figure 1.5, silhouettes can be analyzed to recognize running and walking actions, but also people carrying backpacks or heavy objects. In order to cope with more challenging video data and camera motion, Ramasso et al. [2009] employ a human tracker and camera motion estimation to compute shape information. However, to deal with noisy and imprecise segmentation information, a more robust classification method is used as well.

Another way to match space-time shape models to cluttered image data with heterogeneous background is demonstrated by Ke et al. [2007b]. The authors oversegment video sequences using color information. Volumetric and optical flow features are then matched to action templates in form of space-time shapes. To account for occlusion and actor variability, Ke et al. extend their template to an action part model using pictorial structures.

Silhouettes provide strong cues for action recognition. Nevertheless, they are difficult to compute in the presence of clutter and camera motion. Furthermore, they only describe the outer contours of a person and thus lack discriminative power for actions that include self-occlusions.

Optical flow and shape based methods

Human-centric approaches based on *optical flow* and generic *shape* information form another sub-class of holistic methods. As one of the first works in this direction, Polana and Nelson [1994] propose a human tracking framework along with an action representation using spatio-temporal grids of optical flow magnitudes as shown in figure 2.6. The action descriptor is computed for periodic motion patterns. By matching against reference motion templates of known periodic actions (e.g., walking, running, swimming, skiing) the final action can be determined.

In another approach purely based on optical flow, Efros et al. [2003] track soccer players in videos and compute a descriptor on the stabilized tracks using blurred optical flow.

Their descriptor separates x and y flow as well as positive and negative components into four different channels, as can be seen in figure 2.7. For classification, a test sequence is frame-wise aligned to a database of stored, annotated actions. Further experiments include tennis and ballet sequences as well as synthesis experiments.

The same human-centric representation based on optical flow and human tracks for action recognition is employed by Fathi and Mori [2008]. As classification framework, the authors use a two-layered AdaBoost [Freund and Schapire, 1999] variant. In a first step, intermediate features are learned by selecting discriminative pixel flow values in small spatio-temporal blocks. The final classifier is then learned from a all previously aggregated intermediate features. Evaluations are carried out on four datasets: *KTH*, *Weizmann*, a soccer, and a ballet dataset.

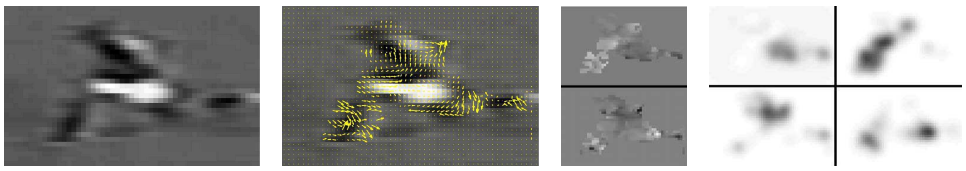
Rodriguez et al. [2008] propose an approach using flow features in a template matching framework. Spatio-temporal regularity flow information is used as feature type. Regularity flow shows improvement over optical flow since it globally minimizes the overall sum of gradients in the sequence. Rodriguez et al. learn cuboid templates by aligning training samples via correlation. For classification, test sequences are correlated with the learned template via generalized Fourier transform that allows for vectorial values. Results are demonstrated on the *KTH* dataset, for facial expressions, as well as on custom movie and sports actions.

To localize humans performing actions such as sit down, stand up, grab cup and close laptop, Ke et al. [2005] use a forward features selection framework and learn a classifier based on optical flow features. Spatio-temporal Haar features on optical flow components are efficiently computed using an integral video structure. During learning, a discriminative set of features are greedily chosen to optimally classify actions which are represented as spatio-temporal cuboidal regions. For classification, the authors perform a sliding window approach and classify each position as containing a particular action or not.

A method purely based on shape information is presented in [Lu and Little, 2006]. In their experiments, Lu and Little track soccer or ice-hockey players and represent each frame by a descriptor using histograms of oriented gradients. They then employ principal component analysis (PCA) [Pearson, 1901] to reduce dimensionality. An HMM with a few states models actions such as running/skating left, right etc.

Hybrid representations combine optical flow with appearance information. Schindler and van Gool [2008] use optical flow information and Gabor filter responses in a human-centric framework. For each frame, both types of information are weighted and concatenated. PCA over all pixel values is applied to learn the most discriminative feature information. Majority voting yields a final class label for a full sequence in multi-class experiments. Results are carried out on the *KTH* and *Weizmann* dataset.

Another recent hybrid representation yields promising results on more realistic video data. Laptev and Perez [2007] demonstrate the localization of drinking actions in movies by learning a cuboid classifier that combines a set of appearance (histograms of oriented gradients) and motion features (histograms of optical flow) as illustrated in figure 2.8. To avoid an exhaustive spatio-temporal search and to improve performance for localizing



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

Figure 2.7: Motion descriptor using optical flow: (a) Original image, (b) Optical flow, (c) Separating the x and y components of optical flow vectors, (d) Half-wave rectification and smoothing of each component (courtesy of Efros et al. [2003]).

actions, the authors propose to pre-filter possible action localizations with a human key-pose detector trained on keyframes of the action.

Human centric approaches necessitate a method for localizing humans, therefore they rely intrinsically on the quality of human detections. To cope with imperfect localizations from weakly labeled training data and an automatic human tracker, Hu et al. [2009] introduce an approach based on multiple instance learning. In the neighborhood around an annotated action or a human detection, a bag of possible action localization hypotheses (i.e., instances) is generated. An initial classifier is learned on all positive and negative instances. Iteratively, instances in bags are relabeled using the previously learned classifier and the classifier is retrained on the new data. Hu et al. apply a simulated annealing strategy to ensure convergence. Feature types that are used are histograms of oriented gradients, foreground segmentation, and motion history images [Bobick and Davis, 2001]. Results are presented on simple actions in crowded sequences as well as in more challenging data recorded in a shopping mall.

Albeit holistic approaches have been shown suitable for action recognition in more realistic video data, certain points are important to note. Holistic representations are in general not invariant to camera view direction. This needs to be accounted for, either by learning different models for particular views (frontal, lateral, rear), or by providing a sufficiently large amount of training data. Additionally, humans can appear at different scales (distant view, close-up view) such that certain parts of the body might not be visible in the image. However, human localizations reduce the computational complexity of detecting actions in time substantially.

2.1.3 Local feature methods

Local space-time features capture characteristic shape and motion information for a local region in video. They provide a relatively independent representation of events with respect to their spatio-temporal shifts and scales as well as background clutter and multiple motions in the scene. Such features are usually extracted directly from video and therefore avoid possible failures of other pre-processing methods such as motion segmentation or human detection.

In the following, we first discuss existing space-time feature detectors and feature descriptors. Methods based on feature trajectories are presented separately since their conception

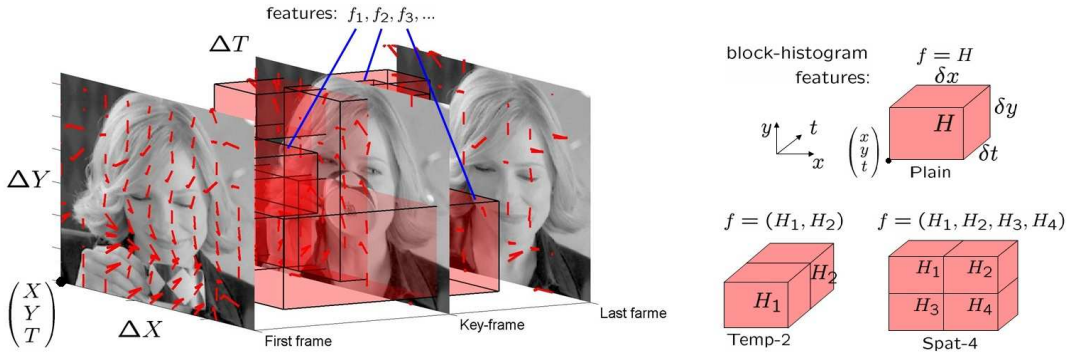


Figure 2.8: (left) A drinking action represented by a set of basic motion and appearance features with varying position and size; (right) each basic feature can have different spatial and temporal layouts internally (courtesy of Laptev and Perez [2007]).

differs from space-time point detectors. We then review approaches which employ the orderless bag-of-features representation and which build spatio-temporal action models based on local features. Finally, methods for localizing actions in videos are discussed.

Feature detectors

Feature detectors usually select characteristic spatio-temporal locations and scales in videos by maximizing specific saliency functions. Laptev and Lindeberg [2003], Laptev [2005] are the first to propose a feature detector based on a spatio-temporal extension of the Harris cornerness criterion [Harris and Stephens, 1988]. The cornerness criterion is based on the eigenvalues of a spatio-temporal second-moment matrix at each video point. Local maxima indicate points of interest. The authors note the importance of using separate spatial and temporal scale values since spatial and temporal extent of events are in general independent. Results of detecting Harris interest points in an outdoor image sequence of a person walking is illustrated in figure 2.9.

Dollár et al. [2005] argue that in certain cases, true spatio-temporal corner points (according to the Harris criterion) are relatively rare, while enough characteristic motion is still present. Therefore, they design their interest point detector to yield denser coverage in videos. Their method employs spatial Gaussian kernels and temporal Gabor filters. As for 3D Harris, local maxima give final interesting positions.

A space-time extension of a salient region detector using entropy, is introduced by Oikonomopoulos et al. [2006]. Entropy is computed in a cylindric neighborhood around a given space-time position for the temporal derivative of a video sequence. To obtain a sparse representation and more stable interest points, local maxima candidates are thresholded and clustered.

The Hessian3D detector is proposed by Willems et al. [2008] as spatio-temporal extension of the Hessian saliency measure applied for blob detection in images [Beaudet, 1978]. The authors aim at a rather dense, scale-invariant, and computationally efficient interest

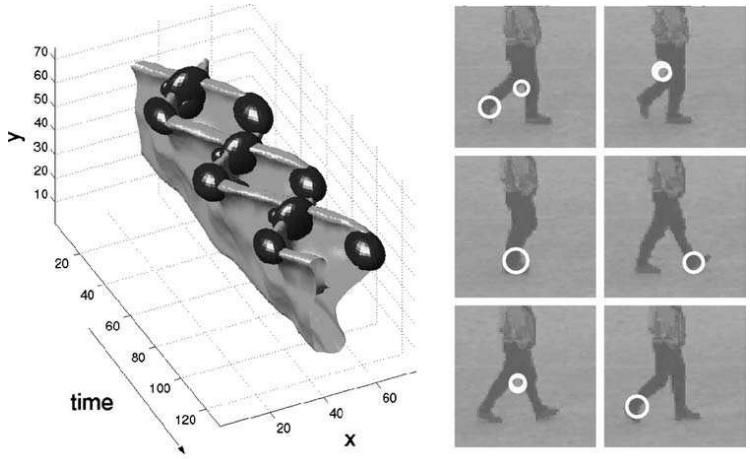


Figure 2.9: Spatio-temporal interest points from the motion of the legs of a walking person; (left) 3D plot of a leg pattern (upside down) and the detected local interest points; (right) interest points overlaid on single frames in the original sequence (courtesy of Laptev [2005]).

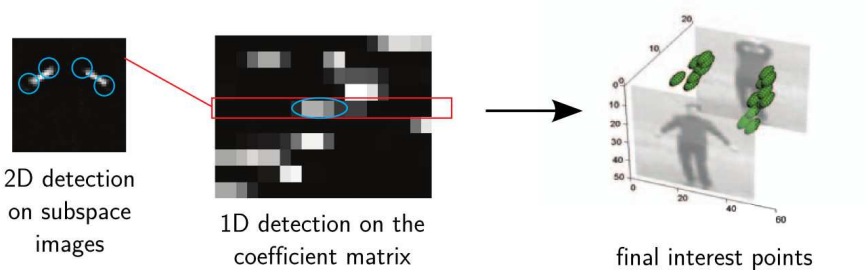


Figure 2.10: Feature detection with global information; (left) spatial feature positions are given by 2D detections in subspace images, (middle) the temporal position is given by maxima in the coefficient matrix; (right) final positions in a waving sequence. (courtesy of Wong and Cipolla [2007]).

point detector. Their detector measures saliency using the determinant of the 3D Hessian matrix. An integral video structure allows to speed up computations by approximating derivatives with box-filter operations. A non-maximum suppression algorithm selects joint extrema over space, time and different scales.

Most feature detectors determine the saliency of a point with respect to its local neighborhood. Wong and Cipolla [2007] suggest to determine salient features by considering global information. For this, video sequences are represented as dynamic texture with a latent representation and a dynamic generation model. This allows to synthesize motion, but also to identify important regions in motion. The dynamic model is approximated as linear transformation. A sub-space representation is computed via non-negative matrix factorization. Local 2D interest in the sub-space images and temporal maxima in their coefficient matrix indicate localizations of globally salient positions, as illustrated in figure 2.10.

Feature descriptors

Feature descriptors capture shape and motion information in a local neighborhood surrounding interest points. Among the first works on local descriptors for videos, Laptev and Lindeberg [2004] develop and compare different descriptor types: single- and multi-scale higher-order derivatives (local jets), histograms of optical flow, and histograms of spatio-temporal gradients. Histograms for optical flow and gradient components are computed for each cell of a $M \times M \times M$ grid layout describing the local neighborhood of an interest point. A different variant describes the surrounding of a given position by applying PCA to concatenated optical flow or gradient components of each pixel. The resulting descriptor uses the dimensions with the most significant eigenvalues. In their experiments, Laptev and Lindeberg report best results for descriptors based on histograms of optical flow and spatio-temporal gradients.

In a similar work, Dollár et al. [2005] evaluate different local space-time descriptors based on brightness, gradient, and optical flow information. They investigate different descriptor variants: simple concatenation of pixel values, a grid of local histograms, and a single global histogram. Finally, PCA reduces the dimensionality of each descriptor variant. Overall, concatenated gradient information yields best performance.

HOG and HOF descriptors are introduced by Laptev et al. [2008]. To characterize local motion and appearance, the authors combine histograms of oriented spatial gradients (HOG) and histograms of optical flow (HOF) in a late fusion approach. The histograms are accumulated in the space-time neighborhood of detected interest points. Each local region is subdivided into a $N \times N \times M$ grid of cells; for each cell, 4-bin HOG histograms and a 5-bin HOF histogram are computed. The normalized cell histograms are concatenated into the final HOG and HOF descriptors.

An extension of the image SIFT descriptor [Lowe, 2004] to 3D was proposed by Scovanner et al. [2007]. For a set of randomly sampled positions, spatio-temporal gradients are computed in the local neighborhood of each position. Each pixel in the neighborhood is weighted by a Gaussian centered on the given position and votes into a $M \times M \times M$ grid of histograms of oriented gradients. For orientation quantization, the authors represent gradients in spherical coordinates ϕ, ψ that are divided into a 8×4 histogram. To be rotation-invariant, the axis corresponding to $\phi = \psi = 0$ is aligned with the dominant orientation of the local neighborhood.

Willems et al. [2008] propose the extended SURF (ESURF) descriptor which extends the image SURF descriptor [Bay et al., 2006] to videos. Like in previous approaches, the authors divide 3D patches into a grid of local $M \times M \times M$ histograms. Each cell is represented by a vector of weighted sums of uniformly sampled responses of Haar-wavelets along the three axes.

Feature trajectories

Feature trajectories are based on spatial interest points tracked in time—as opposed to spatio-temporal interest points. Trajectory shapes encode information about local motion

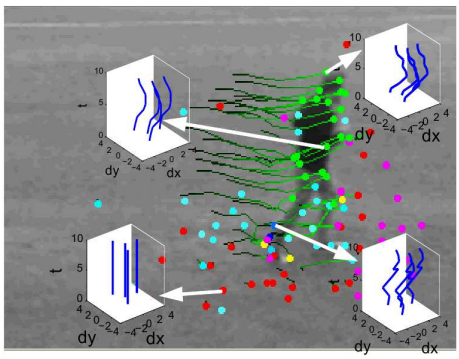


Figure 2.11: Matikainen et al. [2009] obtain feature trajectories by detecting and tracking spatial interest points. Trajectories are quantized to a library of trajectons which are used for action classification (courtesy of Matikainen et al. [2009]).

patterns and can thus be directly used as local feature. Messing et al. [2009] represent feature trajectories of varying length as sequences of log-polar quantized velocities. Activities are modeled using a generative mixture of Markov chain models.

In a different approach, Matikainen et al. [2009, 2010] employ feature trajectories of a fixed length in a bag-of-features framework for action classification (*cf.* figure 2.11). Trajectories of a video are clustered together, and for each cluster center an affine transformation matrix is computed. In addition to displacement vectors, the final trajectory descriptor contains elements of the affine transformation matrix for its assigned cluster center.

Bag of features

A popular representation based on local features is the bag-of-features (BoF) model. It originates from document retrieval applications where orderless methods are a popular choice for representing textual data. The bag-of-words model describes text documents as frequency distributions over words and has been applied extensively in this domain [Salton, 1968].

For visual recognition tasks, Cula and Dana [2001], Sivic and Zisserman [2003], Csurka et al. [2004], Sivic et al. [2005] are among the first authors to extend this concept to visual classification with applications for texture classification, object/scene retrieval, image categorization, and object localization, respectively. Schüldt et al. [2004], Dollár et al. [2005], Niebles et al. [2006] propose the first extensions to action recognition.

For the BoF representation in videos, feature detectors determine a set of salient positions in the sequences. Feature descriptors compute a vector representation for the local neighborhood of a given position. The visual vocabulary (or codebook) is then computed by applying a clustering algorithm (e.g., k-means) on feature descriptors obtained from training sequences; each cluster is referred to as visual word. Descriptors are quantized by assignment to their closest visual word, and video sequences are represented as occurrence histogram of visual words. A non-linear SVM with χ^2 kernel is a popular classifier that

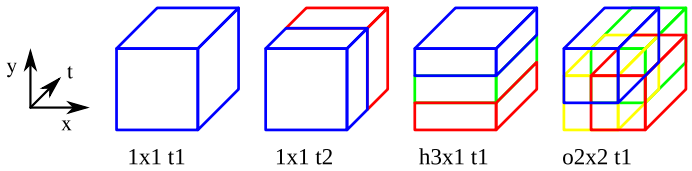


Figure 2.12: Laptev et al. [2008] incorporate weak geometric information in the bag-of-features model by introducing rough spatio-temporal grids overlayed on video sequences (courtesy of Laptev et al. [2008]).

is used throughout different works, e.g., Schüldt et al. [2004], Dollár et al. [2005], Laptev et al. [2008], Willems et al. [2008]. Such histograms only contain global statistics about the type of descriptors that are present in the video sequence. Any information of temporal or spatial relations between the descriptors is ignored.

Spatio-temporal action models

Since the BoF model does not incorporate any geometrical information between features, recent works propose methods to build stronger action models based on local features. For instance, Laptev et al. [2008] include weak geometric information by introducing rough spatio-temporal grids overlayed on video sequences as shown in figure 2.12. Grid layouts as well as shape and motion descriptors are combined by kernel fusion using a non-linear SVM. A greedy optimization strategy learns the best combination of grids and feature types per action class. The authors demonstrate the effectiveness of their approach on the *KTH* dataset and a large set of sample actions obtained from Hollywood movies.

Han et al. [2009] combine different local features with varying layouts and types (histograms of oriented gradients, histograms of optical flow, histograms of oriented spatio-temporal gradients) by fusing multiple kernels using Gaussian processes. By employing various object detectors (for full person, upper body, chairs, cars), they additionally include information about the absence or presence of objects in the sequences. Results on different datasets (*KTH*, *Hollywood1*, *Hollywood2*) demonstrate state-of-the-art classification results.

A hierarchical approach based on SIFT feature trajectories is suggested by Sun et al. [2009]. The authors introduce different levels of context information: the local spatial neighborhood of a trajectory is represented with an averaged SIFT descriptor; a series of state transitions related to quantized orientation and magnitude bins encodes trajectory information; a cuboidal neighborhood captures the relation among adjacent trajectories. In order to capture dynamics of the different levels, Sun et al. use stationary Markov distribution vector. Multiple kernel learning (MKL) [Bach et al., 2004] is employed to combine the different levels of information.

Gilbert et al. [2009] introduce a hierarchical combination of features along with an efficient data mining technique to recognize actions. First, Harris corner points are detected on $(x, y), (x, t), (y, t)$ planes. Detected points are described by their scale and dominant

Figure 2.13: Examples of object-action category detections using an approach based on local features (courtesy of Mikolajczyk and Hirofumi [2008]).

gradient orientation. Then, frequent feature combinations that occurr in a local spatio-temporal neighborhood are learned. These features are combined again in a hierarchical manner. Gilbert et al. propose also a voting scheme to localize actions in video sequences.

Action localization by voting

Combined with a voting scheme, local features can also be employed to spatially as well as temporally localize actions in videos. For instance, Niebles et al. [2006] perform a latent topic discovery and model the posterior probability of each quantized feature for a given action class. In order to localize actions, features are spatially clustered in each frame using k-means.

Mikolajczyk and Hirofumi [2008] propose a voting approach to localize objects that perform a particular action. The authors use a forest of tree classifiers for fast feature quantization. The GLOH image descriptor [Mikolajczyk and Schmid, 2005] together with its dominant motion orientation is used as local descriptor type. Features in motion cast intial hypotheses for position and scale of objects performing an action. Maxima in the voting space indicate detections, and static features refine their intial localization. For the final pose estimation, the object's global orientation is computed from the orientation of voting features. Figure 2.13 illustrates results of object/action detections.

In order to localize actions in YouTube video sequences, Liu et al. [2009] propose an approach based on pruning local features. First, spatio-temporal features are detected and their mean position over a range of neighboring frames is computed. Features that are too far away from the center position are pruned. Second, static features are computed over all frames. By applying the PageRank algorithm over a graph for feature matches in a video sequence, the authors are able to identify discriminative features. For this, similar background features are assumed to by less frequently visible than foreground features. Finally, static and motion features are combined with an AdaBoost classifier. Action localization is carried out with a temporal sliding window over spatio-temporal candidate regions defined by the center and the second moments of motion as well as static features.

Willems et al. [2009] model actions as space-time cubes. They localize drinking actions in movies by casting localization hypotheses for the strongest visual codebook entries of an action. Weak hypotheses are pruned, and a non-linear χ^2 SVM evaluates the BoF representations of remaining ones. Local maxima in the voting space indicated the final action positions. Different action hypotheses and the final detection are shown in figure 2.14.

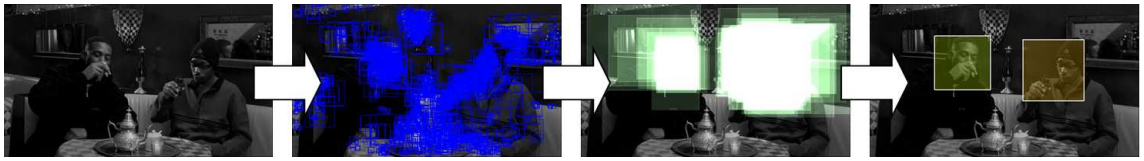


Figure 2.14: Localization of drinking actions based on local features and hypotheses casting (courtesy of Willems et al. [2009]).

A related approach by Yuan et al. [2009] employs the branch-and-bounds algorithm to localize actions in video sequences. Actions are, again, represented as cuboid volumes. The volumes themselves are scored based on mutual information and a Gaussian kernel for density estimation. For a more efficient density estimation, the authors introduce an approximated nearest neighbor search based on local sensitive hashing. Experimental results are shown for the *KTH* and the *CMU* actions dataset.

A key advantage of local features based approaches is their flexibility with respect to the type of video data. They can be applied to videos for which the localization of humans or their body parts is not feasible. More recent works demonstrate its successful application to real world video data, such as Hollywood movies and YouTube video sequences [Laptev et al., 2008, Mikolajczyk and Hirofumi, 2008, Marszałek et al., 2009, Liu et al., 2009].

2.2 Datasets

We present in this section some of the state-of-the-art action recognition datasets that are used in the following. Along with dataset descriptions, we also compare the best results that have been published so far. For this, we distinguish between results for BoF frameworks and overall best results, regardless of the method used.

Subsections 2.2.1 and 2.2.2 describe the *Weizmann* and *KTH* actions dataset, respectively. Both datasets have been used extensively in research, however both represent only a set of rather artificial actions with a homogeneous background. Additionally, the *Weizmann* dataset is about one order of magnitude smaller than *KTH*. The *UCF* sports dataset (subsection 2.2.3) is a collection of TV sport events. It offers a large variety of action classes while being limited in its size. The most challenging and extensive datasets that have been published in the literature are the *YouTube* and *Hollywood2* datasets which are presented in subsections 2.2.4 and 2.2.5. They offer an extensive amount of video sequences in realistic setups: YouTube videos and Hollywood movies, respectively.

2.2.1 Weizmann actions

The Weizmann actions dataset [Blank et al., 2005] [1] consists of ten different types of action classes: bending downwards, running, walking, skipping, jumping-jack, jumping

1. Available at http://www.wisdom.weizmann.ac.il/~vision/SpaceTimeActions.html

Figure 2.15: Sample frames from the Weizmann actions dataset.

	Reference	Method	Accuracy
BoF	chapter 3	Harris3D + HOG3D	90.7%
	Niebles et al. [2008]	Gabor filters + gradients, PLSA	90.0%
	Liu et al. [2008]	Spin + ST Features	90.4%
	Kläser et al. [2008]	Harris3D + HOG3D	84.3%
	Scovanner et al. [2007]	3D-SIFT	82.6%
	Niebles and Fei-Fei [2007]	Shape Context + Gradients + PCA	72.8%
others	Fathi and Mori [2008]	smoothed optical-flow + silhouettes + human tracks + AdaBoost	100%
	Weinland and Boyer [2008]	exemplar-based embedding + silhouettes	100%
	Schindler and van Gool [2008]	Gabor filters + optical flow + human tracks	100%
	Gorelick et al. [2007]	Poisson equation + silhouettes	97.8%
	Wang and Suter [2007]	kernel PCA + factorial CRFs + silhouettes	97.8%
	Zhang et al. [2008]	motion context + foreground segmentation	92.9%
	Ali et al. [2007]	chaotic invariants + silhouettes	92.6%

Table 2.1: State-of-the-art results on *Weizmann* actions reported as avg. class accuracy.

forward, jumping in place, galloping sideways, waving with two hands, and waving with one hand (cf. fig. 2.15). Each action class is performed once (sometimes twice) by 9 subjects resulting in 93 video sequences in total. The background in the videos is homogeneous and static. Blank et al. advocate to test using leave-one-out cross-fold validation, i.e., testing is performed for one sequence at a time while training is executed on all remaining sequences. Performance is given in terms of average accuracy (error rate).

Table 2.1 summarizes current state-of-the-art results on the *Weizmann* dataset. In the literature, several authors report 100% performance for this dataset [Fathi and Mori, 2008, Weinland and Boyer, 2008, Schindler and van Gool, 2008], all employing silhouette information obtained via background substraction. Works based on BoF representations that do not use foreground segmentation have reported results at about 90% [Niebles et al., 2008, Liu et al., 2008]. With our spatio-temporal HOG descriptor, we achieve in a BoF setup comparable results (*cf.* chapter 3).

2.2.2 KTH actions

The *KTH* actions dataset [2] has been introduced by Schüldt et al. [2004]. It consists of six different human action classes: walking, jogging, running, boxing, waving, and clapping (cf. fig. 2.16). Each action class is performed several times by 25 subjects. The sequences were recorded in four different scenarios: outdoors, outdoors with scale variation, outdoors with different clothes, and indoors. The background is homogeneous and static in most sequences. In total, the data consists of 2391 video samples. In the original experimental setup of the authors, samples are divided into a test set (9 subjects: 2, 3, 5, 6, 7, 8, 9, 10, and 22) and training set (the remaining 16 subjects). Evaluation on this dataset is done via multi-class classification. Classification performance is evaluated as average accuracy over all classes.

Current state-of-the-art approaches are given in table 2.2. Gilbert et al. [2009], Han et al. [2009] achieve overall best performance with about 94%. Wong and Cipolla [2007] obtain best results among BoF approaches. With features based on local trajectories (*cf.* chapter 5), we are able to improve significantly over the state-of-the-art for BoF methods and are on par with the overall best results reported in the literature.

There is other work that uses the *KTH* datasets for evaluation, e.g., Fathi and Mori [2008], Jhuang et al. [2007], Wong et al. [2007], Schindler and van Gool [2008], Kim et al. [2007], Uemura et al. [2008], Bregonzio et al. [2009], Liu and Shah [2008], Liu et al. [2008]. However, we cannot compare to them since their results are based on non-standard setups. They reported results either using more training data or splitting the problem into simpler tasks.

2. Available at http://www.nada.kth.se/cvap/actions/

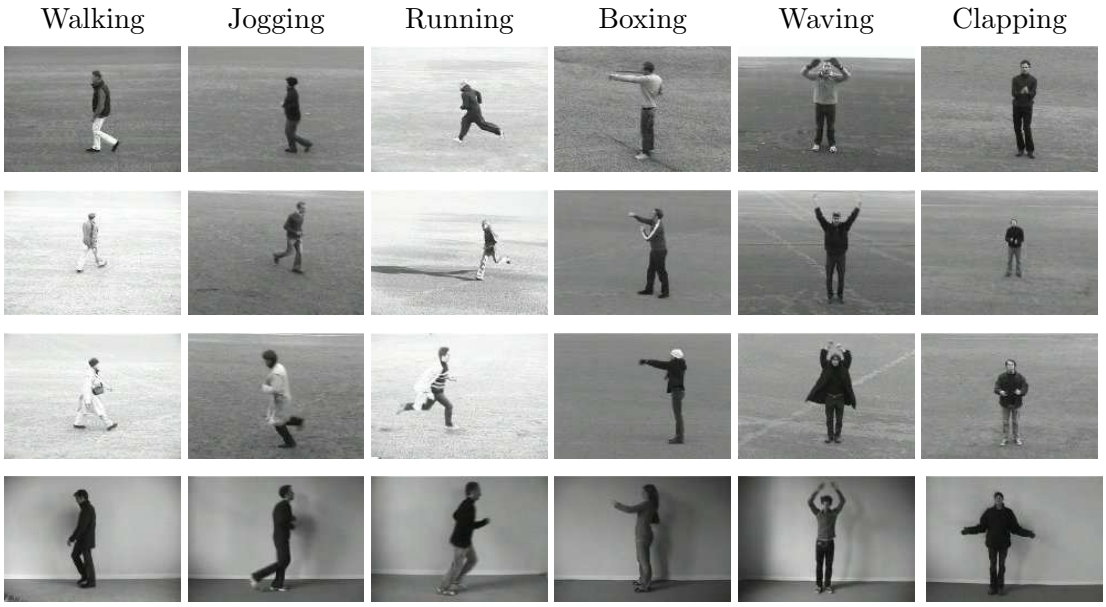


Figure 2.16: Sample frames for all different action classes (columns) in the different scenarios (rows) from the KTH actions dataset.

	Reference	Method	Accuracy
BoF	chapter 5	feature trajectories + HOG-HOF-MBH	94.2%
	chapter 3	Harris3D + HOG3D	92.6%
	chapter 4, Wang et al. [2009]	Harris3D + HOF	92.1%
	chapter 4, Wang et al. [2009]	Harris3D + HOG-HOF	91.8%
	Kläser et al. [2008]	Harris3D + HOG3D	91.4%
	Wong and Cipolla [2007]	non-negative matrix factorization + gradients	86.7%
	Willems et al. [2008]	Hessian3D + extended SURF	84.3%
	Niebles et al. [2008]	Gabor filters + gradients, PLSA	83.3%
	Dollár et al. [2005]	Gabor filters + gradients	81.2%
	Schüldt et al. [2004]	Harris3D + local jets	71.7%
others	Gilbert et al. [2009]	hierarchical data mining	94.5%
	Han et al. [2009]	different local features + grid layouts + object detectors	94.1%
	Yuan et al. [2009]	mutual information for sets of unquantized local features	93.3%
	chapter 6	dense + HOG3D + human tracks	92.1%
	Laptev et al. [2008]	Harris3D + HOG + HOF + grid layouts	91.8%

Table 2.2: State-of-the-art results on the *KTH* dataset reported as average class accuracy.

	Reference	Method	Accuracy
BoF	chapter 4	Gabor + HOG3D	85.0%
others	chapter 6	dense + HOG3D + human tracks	90.1%
	Rodriguez et al. [2008]	MACH template matching	69.2%

Table 2.3: State-of-the-art results on the *UCF* dataset reported as average class accuracy.

2.2.3 UCF sport actions

The UCF sport actions dataset [Rodriguez et al., 2008] [3] contains ten different types of human actions: swinging (on the pommel horse and on the floor), diving, kicking (a ball), weight-lifting, horse-riding, running, skateboarding, swinging (at the high bar), golf swinging and walking (*cf.* figure 2.17). The dataset consists of 150 video samples which show a large intra-class variability. The performance criterion for the multi-class task is the average accuracy over all classes. The original setup employs leave-one-out for testing.

The only published results that are known to use for *UCF* are given by Rodriguez et al. [2008]. They report 69.2% accuracy which we outperform significantly in our experiments (chapters 4 and 6).

2.2.4 YouTube actions

The *YouTube* dataset has been introduced by Liu et al. [2009] [4] and contains 11 action categories: basketball shooting, biking/cycling, diving, golf swinging, horse back riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog (in Figure 2.18). This dataset is challenging due to large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, illumination conditions etc. The dataset contains a total of 1600 sequences. In the original setting, the evaluation is carried out using cross validation for a set of 25 folds that is defined by the authors. Average accuracy over all classes is used as performance measure.

To the best of our knowledge, Liu et al. [2009] are the only authors so far to evaluate on this dataset. They obtain 71.2% which is slightly better than we obtain with our spatio-temporal HOG descriptor (chapter 3). With local feature trajectories (chapter 5), we yield a significant improvement of over 8.5%.

2.2.5 Hollywood actions

There exist two versions of the *Hollywood* actions dataset: *Hollywood1* [Laptev et al., 2008] and *Hollywood2* [Marszałek et al., 2009]. To avoid exhaustive manual annotation of several hundreds of hours of movie data, the authors use in both cases movie scripts which provide textual description of the movie content, such as scenes, characters, transcribed dialogues,

3. Available at http://www.cs.ucf.edu/vision/public_html/
4. Available at http://www.cs.ucf.edu/~liujg/YouTube_Action_dataset.html

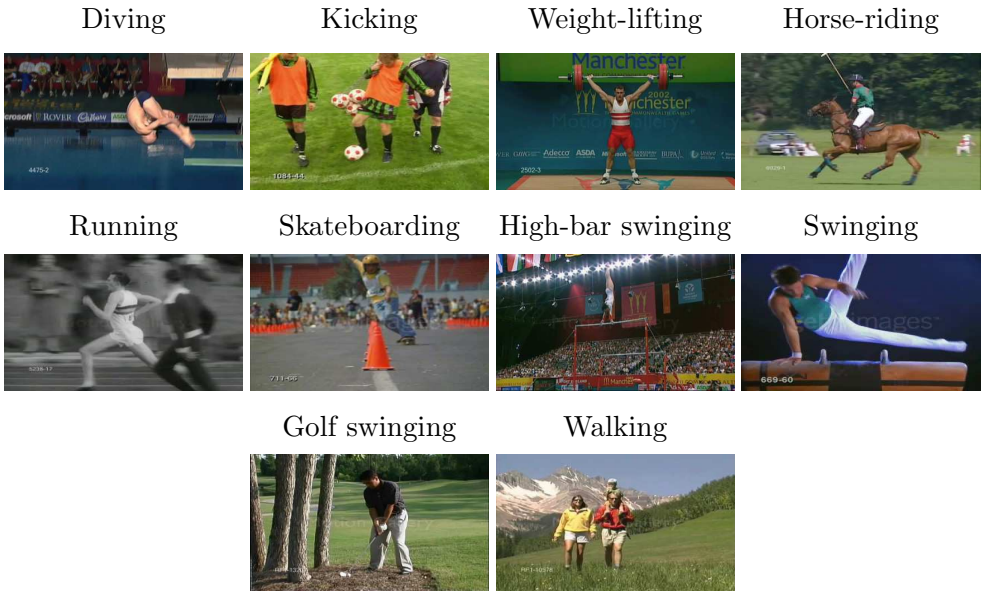


Figure 2.17: Sample frames for all action classes of the UCF sport action datasets.

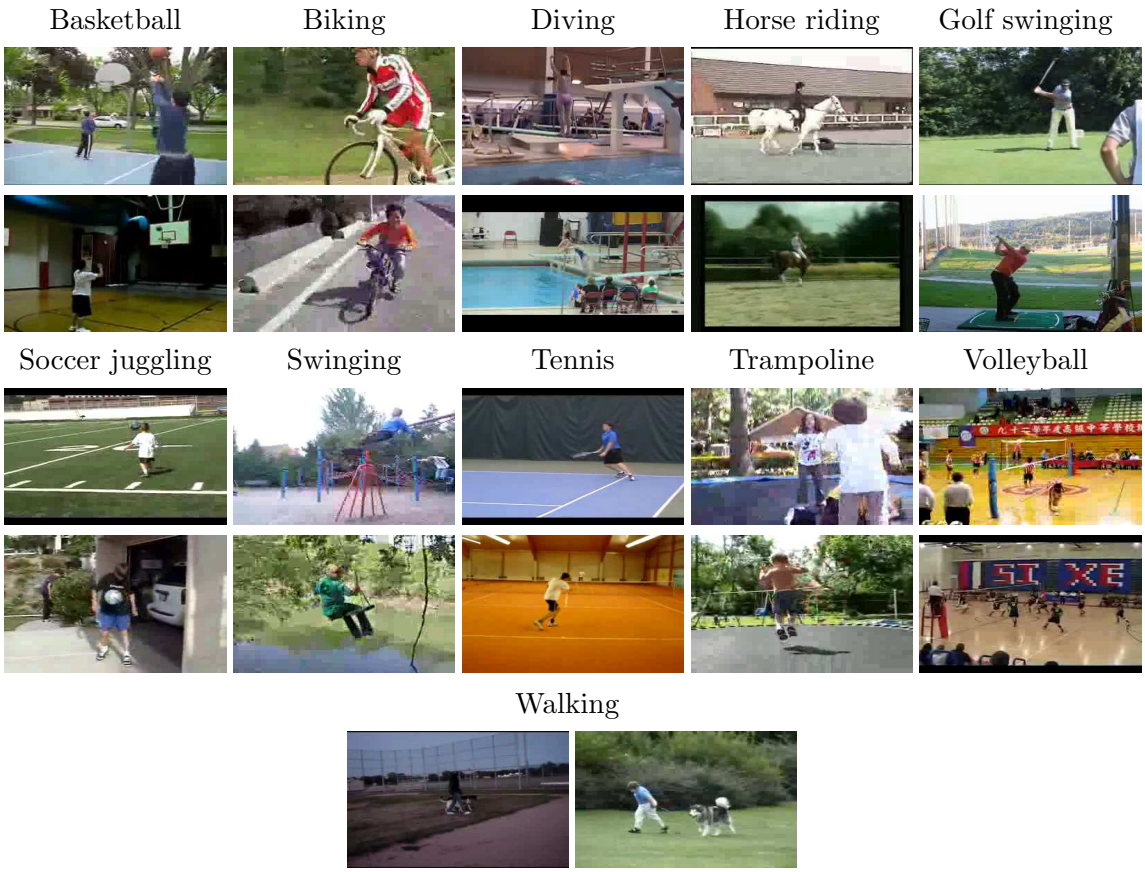


Figure 2.18: Sample frames from the *YouTube* action dataset; two samples are given for each of the eleven action classes.

	Reference	Method	Accuracy
BoF	chapter 5	feature trajectories + HOG-HOF-MBH	79.8%
	chapter 3	Harris3D + HOG3D	68.3%
others	Liu et al. [2009]	motion/static features + pruning + grouping + AdaBoost	71.2%

Table 2.4: State-of-the-art results on the *YouTube* dataset reported as avg. class accuracy.

	Reference	Method	AP
BoF	Willems et al. [2009]	Hessian3D + HOG3D variant	29.6%
	Laptev et al. [2008]	Harris3D + HOG	27.0%
	Kläser et al. [2008]	Harris3D + HOG3D	24.7%
	Laptev et al. [2008]	Harris3D + HOF	21.5%
others	Gilbert et al. [2009]	hierarchical data mining	53.5%
	Han et al. [2009]	different local features + grid layouts + object detectors	47.5%
	Sun et al. [2009]	hierarchical context model, feature trajectories	47.1%
	Laptev et al. [2008]	Harris3D + HOG + HOF + grid layouts	38.4%
	chapter 6	dense + HOG3D + human tracks	36.4%

Table 2.5: State-of-the-art results on the *Hollywood1* dataset reported as mean AP.

and human actions. In a first step, scripts are aligned to movie subtitles since they usually come without time information. In a second step, classifiers are trained on a bag-of-words representation of the scene description for different action classes. Several features are used: bag-of-words over single words, over adjacent pairs of words, as well as over pairs of words in a small neighborhood. This allows to cope with significant variations in the text and to retrieve action samples. The authors manually ensure the visual integrity of annotations in the train and test set and additionally provide a noisy training set.

The first version, *Hollywood1*, has been published by Laptev et al. [2008] [5]. It contains eight different action classes: answering the phone, getting out of the car, hand shaking, hugging, kissing, sitting down, sitting up, and standing up. Action samples have been collected from in total 32 different Hollywood movies. The full dataset contains 663 video samples, divided into a clean training set (219 sequences) and a clean test set (211 sequences), where training and test sequences were obtained from different movies. The additional noisy training set consists of 233 sequences.

Hollywood2 is the extended version introduced by Marszałek et al. [2009] [6]. In total it consists of samples from 69 different Hollywood movies. The initial eight action classes

5. Available at http://www.irisa.fr/vista/actions/
6. Available at http://www.irisa.fr/vista/actions/hollywood2

Figure 2.19: Sample frames from the *Hollywood2* action dataset; two samples are given for each of the twelve action classes.

	Reference	Method	AP
BoF	chapter 5	feature-trajectories + HOG-HOF-MBH	52.5%
	chapter 3	Harris3D + HOG3D	48.6%
	chapter 4, Wang et al. [2009]	Harris3D + HOG/HOF	47.6%
others	Gilbert et al. [2009]⋆	hierarchical data mining	50.9%
	Han et al. [2009]	different local features + grid layouts + object detectors	42.1%

⋆ *Unpublished results, personal communication with the authors.*

Table 2.6: State-of-the-art results on the *Hollywood2* dataset reported as mean AP.

were extended by adding four additional ones: driving car, eating, fighting, and running. Action samples for all classes are illustrated in figure 2.19. In total, there are 2517 action samples split into a manually cleaned training set (823 sequences) and a test set (884 sequences). The noisy training set contains 810 sequences. Train and test sequences are obtained from different movies.

The performance for both, *Hollywood1* and *Hollywood2*, is evaluated by computing the average precision (AP) for each of the action classes and reporting the mean AP over all classes (mAP). Note that this follows the evaluation procedure which has been established by the Pascal Visual Object Class Challenge [Everingham et al., 2008].

For both variants of this dataset, Gilbert et al. [2009] yield current state-of-the-art results: 53.5% on *Hollywood1* and 50.9% on *Hollywood2*. In chapter 5, we show that we outperform their results by 1.6% (i.e., 52.5%) with local descriptors based on feature trajectories. We cannot compare to Marszałek et al. [2009], since they only report results for classifiers trained on the noisy dataset.

Un descripteur basé sur des gradients spatio-temporels

En suivant l'évolution récente de la reconnaissance visuelle des images statiques, de nombreux concepts ont été étendus et appliqués à des séquences vidéo Par exemple: des détecteurs des points pertinents, des descripteurs locaux, le modèle sac-de-mot et la localisation d'actions en utilisant des caractéristiques locales. Cependant, malgré le progrès récent, il existe relativement peu de descripteurs locaux en vidéos qui bénéficient conjointement de l'information spatiale et temporelle.

Ce chapitre présente un nouveau descripteur spatio-temporel de caractéristiques locales en vidéo. S'appuyant sur le succès des histogrammes de gradients orientés (HOG) pour des images statiques [Dalal et al., 2006, Lowe, 2004], nous généralisons les concepts clés du HOG à la 3D. À cette fin, nous étudions les polyèdres réguliers et les coordonnées sphériques afin de discrétiser l'orientation des gradients spatio-temporels. En outre, nous employons des vidéos intégrales pour rendre le calcul des gradients plus efficace. Les paramètres de notre descripteur sont évalués sur quatre bases de données différentes (*KTH*, *Weizmann*, *YouTube* et *Hollywood2*) et ils sont optimisés pour la reconnaissance d'action dans des vidéos.

3

A spatio-temporal descriptor based on 3D-gradients

Contents

Based on recent developments of visual recognition in static images, many concepts have been successfully extended to video sequences, for instance: feature detectors, feature descriptors, bag-of-features (BoF) representations, local features based voting for localization. However, despite recent developments, relatively few local descriptors in videos exist that benefit from combined spatial and temporal information.

This chapter introduces a novel spatio-temporal descriptor for local features in video. Building on the success of descriptors based on histograms of oriented gradients (HOG) for static images [Dalal et al., 2006, Lowe, 2004], we view videos as spatio-temporal volumes and generalize the key HOG concepts to 3D. To this end, we investigate regular polyhedrons and spherical coordinates for 3D orientation quantization and employ integral videos for efficient computation of gradients. Descriptor parameters are evaluated on four action datasets (*KTH*, *Weizmann*, *YouTube*, *Hollywood2*) and are optimized for action recognition.

3.1 Introduction

Different types of descriptors have been investigated in the past. Dollár et al. [2005] compare descriptors based on pixel values, brightness gradients, and optical flow. Laptev and Lindeberg [2004] evaluate single- and multi-scale higher oder derivatives, histograms of optical flow, and histograms spatio-temporal gradients. Overall, the authors find gradient and optical flow based methods to yield best results in their experiments. However, in both works, descriptors are based on gradient magnitude components which are shown to suffer from sensitivity to illumination changes [Freeman and Roth, 1995].

At the same time, representations based on histograms of oriented gradients (HOG) have been shown suitable representations for images since orientation information is robust to changes in illumination [Freeman and Roth, 1995]. HOG is successfully used for local feature representations [Lowe, 2004, Mikolajczyk and Schmid, 2005] as well as for dense description of objects in images [Dalal and Triggs, 2005, Felzenszwalb et al., 2010].

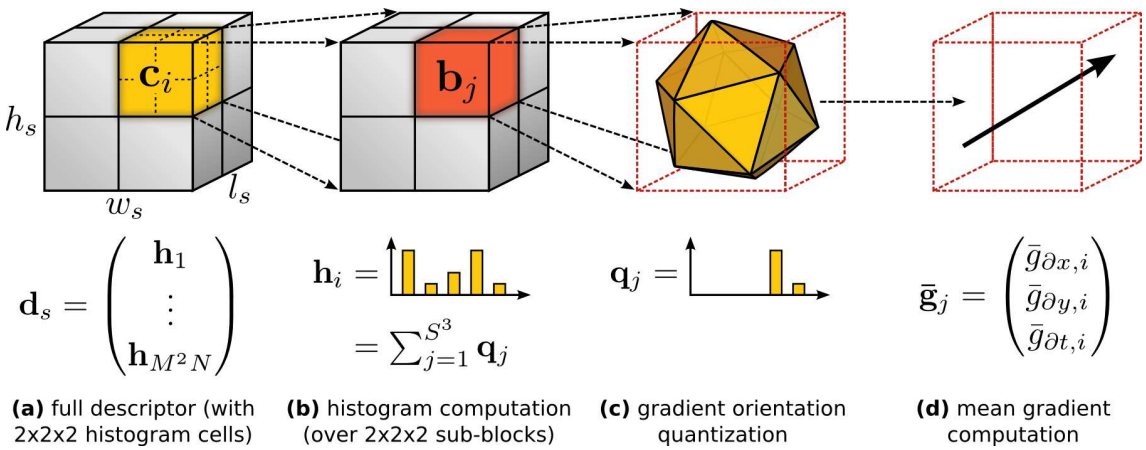
Laptev et al. [2008] use orientation information to recognize actions. The authors combine histograms of optical flow (HOF) and HOG descriptors to capture motion and appearance information. However, they only consider spatial gradients and employ for gradients and optical flow a rough orientation quantization into only four bins. Unlike Laptev et al., we base our descriptor on histograms of *spatio-temporal* 3D gradient orientation. Spatio-temporal gradients are fast and cheap to compute, as opposed to optical flow. In addition to this, they combine motion as well as appearance information in one representation.

The closest work to our descriptor is an extension of the popular SIFT image descriptor [Lowe, 2004] to the spatio-temporal domain proposed by Scovanner et al. [2007]. To quantize gradient orientations the authors use regular binning based on spherical coordinates. However, quantization and descriptor parameters are not evaluated and experiments are only carried out on a small dataset (*Weizmann* actions) with static background.

In our work, we evaluate descriptor parameters in depth on several datasets of varying degree of difficulty and optimize them for BoF-based action recognition. Furthermore we compare different gradient quantization strategies: orientation quantization with up to 20 bins using regular polyhedrons and spherical coordinates for which the amount of bins can be controlled separately for spatial and temporal gradient orientations. In addition to this, we employ integral histograms for memory-efficient computation of features at arbitrary spatial and temporal scales. This technique shows advantages over common approaches that need to precompute descriptor information for a coarse set of predefined spatio-temporal scales [Laptev et al., 2008, Dollár et al., 2005]. Integral videos are related to Willems et al. [2008] as well as Ke et al. [2005]. Both works employ integral histograms for videos to compute spatio-temporal Haar wavelets.

3.2 Spatio-temporal descriptor

A sampling point $(x, y, t, \sigma, \tau)^\mathsf{T}$ is located in the video sequence at position $(x, y, t)^\mathsf{T}$. Its characteristic spatial and temporal scale are given by σ and τ, respectively. The spatial



(a) full descriptor (with 2x2x2 histogram cells) **(b)** histogram computation (over 2x2x2 sub-blocks) **(c)** gradient orientation quantization **(d)** mean gradient computation

Figure 3.1: Overview of the descriptor computation; (a) the support region around a point of interest is divided into a grid of gradient orientation histograms; (b) each histogram is computed over a grid of mean gradients; (c) each gradient orientation is quantized using regular polyhedrons; (d) each mean gradient is computed using integral videos.

scale (σ) accounts for similar structures appearing at a different size in the image plane. The temporal scale (τ) models similar motion happening at a different speed, i.e., over a different length of time. σ and τ determine the spatial and temporal neighborhood size of the descriptor at position (x, y, t).

Figure 3.1 illustrates the different steps for computing our 3D gradient orientation descriptor. Each step is discussed in detail in the following. Section 3.2.1 explains the proposed efficient computation of 3D gradients with arbitrary spatial and temporal scales (fig. 3.1d). The orientation quantization of 3D gradients is presented in section 3.2.2 (fig. 3.1c). Section 3.2.3 summarizes the computation of orientation histograms (fig. 3.1b), and finally the construction of the descriptor itself is explained in section 3.2.4 (fig. 3.1a).

3.2.1 Gradient computation

A video sequence v is given as a function $v : \mathbf{R}^2 \times \mathbf{R} \to \mathbf{R}$. To account for space-time structures at different scales, its scale-space representation $L : \mathbf{R}^2 \times \mathbf{R} \times \mathbf{R}^2_+ \to \mathbf{R}$ is constructed by its convolution with an anisotropic Gaussian kernel with independent spatial and temporal variance (σ, τ) [Laptev, 2005]:

$$L(\cdot; \sigma, \tau) = G(\cdot; \sigma, \tau) * v(\cdot), \tag{3.1}$$

where the spatio-temporal separable Gaussian kernel is defined as

$$G(x, y, t; \sigma, \tau) = \frac{1}{\sqrt{(2\pi)^3 \sigma^4 \tau^2}} \exp\left(-\frac{(x^2 + y^2)}{2\sigma^2} - \frac{t^2}{2\tau^2}\right). \tag{3.2}$$

In order to compute histograms over 3D gradient orientations for different spatio-temporal scales, gradient vectors need to be computed efficiently for cuboid regions of different size (*cf.* fig. 3.1d).

One strategy to improve computational efficiency is to use spatio-temporal "pyramids". Such a pyramid is defined by a set of combinations of different temporal and spatial scales, and gradients could be precomputed for each scale combination. This approach is in the spirit of work by Dollár et al. [2005], Laptev [2005], Laptev et al. [2008]. However, for each given spatio-temporal scale, the video sequence needs to be rescaled and stored. Precisely, given N scale steps in total as well as a spatial and a temporal scaling factor s_σ, s_τ, this amounts in a factor $z = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} s_\sigma^{-2i} s_\tau^{-j}$ of additional data that needs to be stored as well as processed. For instance, if we assume a fine spatial and temporal scale grid with $s_\sigma = s_\tau = \sqrt[4]{2}$ over six octaves in total, i.e., $N = 24$, one will need to compute 24×24 different video scales. This results in a factor $z \approx 21$ of extra data. Therefore, only a rough representation of the scale space with a few scale combinations is commonly chosen in practice.

As memory-efficient yet still flexible alternative, we propose to use integral videos for computing mean gradient vectors. For this, we compute the gradient vector $(\mathrm{d}x, \mathrm{d}y, \mathrm{d}t)$ in the scale-space representation L of the video sequence v as

$$\nabla L_{\sigma,\tau} = \nabla(G_{\sigma,\tau} * v) = G_{\sigma,\tau} * \nabla v \approx B_{\sigma,\tau} * \nabla v, \tag{3.3}$$

with

$$L_{\sigma,\tau} = L(\cdot; \sigma, \tau), \qquad G_{\sigma,\tau} = G(\cdot; \sigma, \tau) \tag{3.4}$$

and we approximate the Gaussian kernel G with the box filter B. 3D gradients are thus first computed for all pixel positions in the original video sequence. By calculating their integral video representation (see below), the box filter can be computed for any arbitrary cuboid and thus for any arbitrary x-, y-, and t-scale in constant time.

The concept of integral images has been popularized by Viola and Jones [2001]. They used integral images as an intermediate representation for efficient computation of Haar features. We extend integral images to integral videos on gradient vectors. Given the video sequence $v(x, y, t)$ and its gradient representation $\nabla v = (\frac{\partial v}{\partial x}, \frac{\partial v}{\partial t}, \frac{\partial v}{\partial t})^\mathsf{T}$, its integral video representation can be described as

$$\nabla V(x, y, t) = \sum_{x' \le x, y' \le y, t' \le t} \nabla v(x', y', t'). \tag{3.5}$$

For any 3D cuboid $\mathbf{b} = (x, y, t, w, h, l)^\mathsf{T}$ described by its position $(x, y, t)^\mathsf{T}$ and its width (w), height (h), and length (l), we can compute the its mean gradient $\bar{\mathbf{g}}_\mathbf{b} = (\bar{g}_{\mathbf{b}x}, \bar{g}_{\mathbf{b}y}, \bar{g}_{\mathbf{b}t})^\mathsf{T}$ as

$$\begin{aligned}\bar{\mathbf{g}}_\mathbf{b} = {} & [\nabla V(x+w, y+h, t+l) - \nabla V(x, y+h, t+l) - \nabla V(x+w, y, t+l) + \nabla V(x, y, t+l)] \\ & - [\nabla V(x+w, y+h, t) - \nabla V(x, y+h, t) - \nabla V(x+w, y, t) + \nabla V(x, y, t)] \ . \end{aligned} \tag{3.6}$$

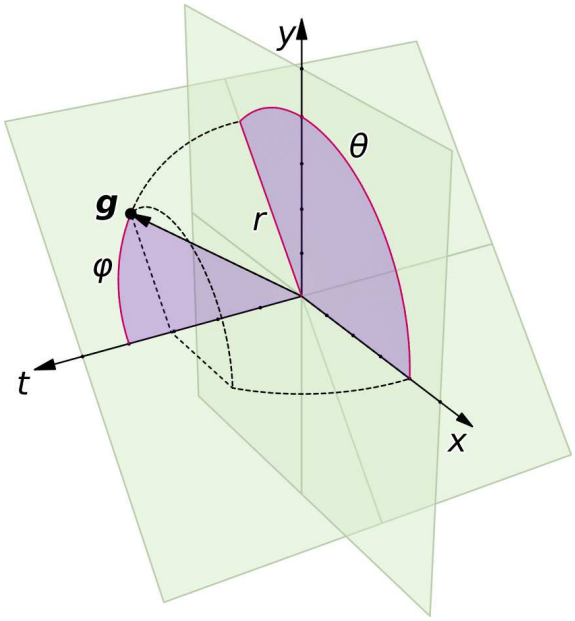


Figure 3.2: Illustration of quantization of 3D gradient orientations using spherical coordinates with azimuth (θ) and elevation angle (φ).

3.2.2 Orientation quantization

Given the 3D gradient $\bar{\mathbf{g}}_\mathbf{b}$, we seek to quantize its orientation into a histogram $\mathbf{q}_\mathbf{b}$ of discrete bins (*cf.* figure 3.1c). This can be seen as quantizing the surface of a unit sphere. In the following, we investigate two different aproaches. First, we discuss how to quantize the orientation of a 3D gradient using spherical coordinates with azimuth and elevation angle. Second, we propose a quantization strategy using regular polyhedrons.

Spherical coordinate based quantization. The orientation of a spatio-temporal gradient can be quantized using its spherical coordinate representation with azimuth (θ) and elevation angle (φ), as illustrated in figure 3.2. The spherical coordinate representation (r, θ, φ) for the gradient $\bar{\mathbf{g}}_\mathbf{b}$ is given by

$$r = ||\bar{\mathbf{g}}_\mathbf{b}||_2 \tag{3.7}$$

$$\theta = \arctan\left(\frac{\bar{g}_{\mathbf{b}y}}{\bar{g}_{\mathbf{b}x}}\right) \tag{3.8}$$

$$\varphi = \arccos\left(\frac{\bar{g}_{\mathbf{b}t}}{r}\right). \tag{3.9}$$

In order to compute a weighted histogram of gradient orientations, θ and φ are divided into B_θ and B_φ equally sized bins. This is similar to a division with meridians and parallels on a unit sphere. A gradient votes with its magnitude r into its four closest bins using bilinear interpolation: each entry into a bin is multiplied by a weight of $1 - d$; d is the distance of the sample to the central value of the bin measured in units of the histogram bin spacing.

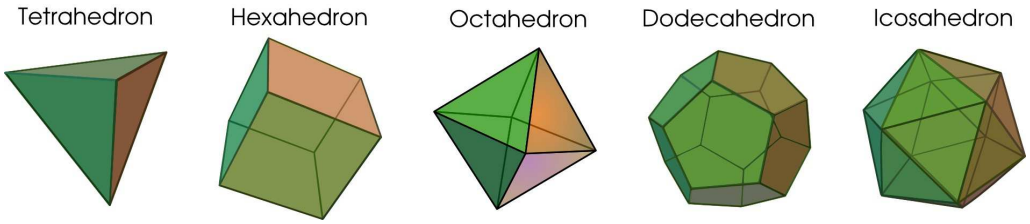


Figure 3.3: Illustration of the different existing regular polyhedrons (courtesy of Wikipedia [2010]).

Polyhedron based quantization. A n-bin histogram of gradient orientations in 2D (i.e., for static images) can be interpreted as an approximation of a circle (i.e., the continuous space of orientations) with a regular n-sided polygon. Each side of the polygon corresponds to a histogram bin. The equivalent of a two-dimensional polygon is a polyhedron for the three dimensional space. Regular polyhedrons with congruent faces are referred to as platonic solids; only five of them exist: the tetrahedron (4-sided), cube (6-sided), octahedron (8-sided), dodecahedron (12-sided), and icosahedron (20-sided) (*cf.* fig. 3.3). In our experiments, we consider the dodecahedron and the icosahedron for 3D gradient quantization since they result in the largest number of orientation bins.

Given a regular n-sided polyhedron, let its center of gravity lie at the origin of a three-dimensional Euclidean coordinate system. In order to quantize a 3D gradient vector $\bar{\mathbf{g}}_\mathbf{b}$ w.r.t. its orientation, we first project $\bar{\mathbf{g}}_\mathbf{b}$ on the axes running through the origin of the coordinate system and the center positions of all faces. This can be done with matrix multiplication. Let P be the matrix of the center positions $\mathbf{p}_1, \ldots, \mathbf{p}_n$ of all n faces

$$\mathsf{P} = (\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n)^\mathsf{T} \qquad \text{with} \qquad \mathbf{p}_i = (x_i, y_i, t_i)^\mathsf{T} \ . \tag{3.10}$$

For instance, the icosahedron can be described with the following 20 center points:

$$(\pm 1, \pm 1, \pm 1) \quad (0, \pm 1/\phi, \pm\phi) \quad (\pm 1/\phi, \pm\phi, 0) \quad (\pm\phi, 0, \pm 1/\phi) \tag{3.11}$$

with the golden ratio $\phi = \frac{1+\sqrt{5}}{2}$. The projection $\hat{\mathbf{q}}_\mathbf{b}$ of $\bar{\mathbf{g}}_\mathbf{b}$ is obtained through:

$$\hat{\mathbf{q}}_\mathbf{b} = (\hat{q}_{\mathbf{b}1}, \ldots, \hat{q}_{\mathbf{b}n})^\mathsf{T} = \frac{\mathsf{P} \cdot \bar{\mathbf{g}}_\mathbf{b}}{\|\bar{\mathbf{g}}_\mathbf{b}\|_2} \ . \tag{3.12}$$

Thus, each $\hat{q}_{\mathbf{b}i}$ of $\hat{\mathbf{q}}_\mathbf{b}$ holds the normalized projection of the gradient vector $\bar{\mathbf{g}}_\mathbf{b}$ onto the axes through the face center $\mathbf{p}_i$, i.e.,

$$\hat{q}_{\mathbf{b}i} = \|\mathbf{p}_i\|_2 \cdot \cos_\angle(\mathbf{p}_i, \bar{\mathbf{g}}_\mathbf{b}) = \|\bar{\mathbf{g}}_\mathbf{b}\|_2^{-1} \cdot \mathbf{p}_i{}^\mathsf{T} \cdot \bar{\mathbf{g}}_\mathbf{b}. \tag{3.13}$$

For a histogram with half orientation, opposite directions can be associated into the same bin by halving the set of face centers and taking the absolute value of $\hat{q}_{\mathbf{b}i}$.

Next, the resulting vector $\hat{\mathbf{q}}_\mathbf{b}$ of the projection is thresholded. This is done, since $\bar{\mathbf{g}}_\mathbf{b}$ is expected to vote into only one bin in case it is perfectly aligned with the corresponding axis running through the origin and the face center. By comparing two neighboring axes $\mathbf{p}_i$ and $\mathbf{p}_j$, this threshold value is given by $t = \mathbf{p}_i^\mathsf{T} \cdot \mathbf{p}_j$. For the icosahedron given in (3.11) $t \approx 1.29107$. Threshold t is subtracted from $\hat{\mathbf{q}}_\mathbf{b}$ and all negative elements are set to zero. The gradient magnitude is distributed according to the thresholded histogram $\hat{\mathbf{q}}'_\mathbf{b}$:

$$\mathbf{q}_\mathbf{b} = \frac{||\bar{\mathbf{g}}_\mathbf{b}||_2 \cdot \hat{\mathbf{q}}'_\mathbf{b}}{||\hat{\mathbf{q}}'_\mathbf{b}||_2} \quad . \tag{3.14}$$

In our experiments (see section 3.3.2 for details) we have found that the type of quantization is dataset dependend.

3.2.3 Histogram computation

A histogram of gradient orientations is computed over a set of gradient vectors. Given a particular cell in our descriptor (*cf.* figure 3.1b), denoted as $\mathbf{c} = (x_\mathbf{c}, y_\mathbf{c}, t_\mathbf{c}, w_\mathbf{c}, h_\mathbf{c}, l_\mathbf{c})^\mathsf{T}$, we divide $\mathbf{c}$ into $S \times S \times S$ subblocks $\mathbf{b}_i$. These S^3 subblocks form the set over which the cell histogram is computed. For each of the subblocks $\mathbf{b}_i$, the corresponding mean gradient $\bar{\mathbf{g}}_{\mathbf{b}_i}$ is computed using integral videos as defined in equation (3.6). $\bar{\mathbf{g}}_{\mathbf{b}_i}$ is subsequently quantized as $\mathbf{q}_{\mathbf{b}_i}$ employing a regular polyhedron (see equation (3.14)). The histogram $\mathbf{h}_\mathbf{c}$ for the region $\mathbf{c}$ is then obtained by summing the quantized mean gradients $\mathbf{q}_{\mathbf{b}_i}$ of all subblocks $\mathbf{b}_i$:

$$\mathbf{h}_\mathbf{c} = \sum_{i=1}^{S^3} \mathbf{q}_{\mathbf{b}_i} \quad . \tag{3.15}$$

With a fixed number of supporting mean gradient vectors (S^3), and by using integral videos for computing mean gradients of subblocks, a histogram can be computed for any arbitrary scale at x, y, t. At the same time the memory requirements for storage are linear in the number of pixels in the video sequence. They do not depend on a number of predefined spatio-temporal scales.

Our experiments on two different datasets show (see section 3.3.2 for details) that $S = 4$, resulting in 64 supporting mean gradient vectors yields best performance for action recognition irrespective of the dataset.

3.2.4 Descriptor computation

A sampling point $\mathbf{s} = (x_\mathbf{s}, y_\mathbf{s}, t_\mathbf{s}, \sigma_\mathbf{s}, \tau_\mathbf{s})^\mathsf{T}$ is located in the video sequence at $(x_\mathbf{s}, y_\mathbf{s}, t_\mathbf{s})^\mathsf{T}$ with characteristic spatial and temporal scale $(\sigma_\mathbf{s}, \tau_\mathbf{s})$, respectively. The final descriptor $\mathbf{d}_\mathbf{s}$

for $\mathbf{s}$ is computed for a local support region $\mathbf{r_s} = (x_\mathbf{r}, y_\mathbf{r}, t_\mathbf{r}, w_\mathbf{r}, h_\mathbf{r}, l_\mathbf{r})^\mathsf{T}$ around the position $\mathbf{s}$ (see figure 3.1a) with width ($w_\mathbf{s}$), height ($h_\mathbf{s}$), and length ($l_\mathbf{s}$) given by

$$w_\mathbf{s} = h_\mathbf{s} = \sigma_0\sigma_\mathbf{s} \ , \qquad l_\mathbf{s} = \tau_0\tau_\mathbf{s} \ . \tag{3.16}$$

The parameters σ_0 and τ_0 characterize the relative size of the support region around $\mathbf{s}$.

Similar to other approaches [Dollár et al., 2005, Laptev, 2005, Laptev and Lindeberg, 2004, Laptev et al., 2008, Scovanner et al., 2007], the local support region $\mathbf{r_s}$ is divided into a set of $M \times M \times N$ cells $\mathbf{c}_i$. For each cell, an orientation histogram is computed (see equation (3.15)). Each cell histogram is finally normalized by its $\mathcal{L}_2$-norm and concatenated to one feature vector $\mathbf{d_s} = (d_1, \ldots, d_{M^2N})^\mathsf{T}$.

For different datasets, we found the scale parameters of $\sigma_0 = [16, 24]$ and $\tau_0 = [4, 12]$ to yield satisfying results (*cf.* section 3.3.2). The number of spatial cells showed to be more dependent on a specific dataset with values in the range $M = [2, 5]$. For the number temporal divisions, N seemed to be relatively insensitive to different values. In practice $N = 4, 5$ obtained best performance.

3.3 Experimental results

In the following sections, we present experimental results for our descriptor. Section 3.3.1 details the setup for experiments and section 3.3.2 presents results for learning parameters. Section 3.3.3 compares results on four datasets to current state-of-the-art approaches.

3.3.1 Experimental setup

Bag-of-features. We evaluate the performance of our descriptor on the task of action recognition by employing the bag-of-features setup as detailed in section A.1 (k-means for codebook generation, codebook size 4000, χ^2-kernel SVM). For interest point detection, we use the Harris3D feature detector [Laptev et al., 2008] (*cf.* section 4.2.1). When learning the parameter settings (section 3.3.2), we employ random sampling on training features for codebook generation in order to speed up computations (also codebook size 4000).

Baseline features. As baseline method, we use the HOG (histograms of oriented spatial gradients) and HOF (histograms of optical flow) descriptors proposed by Laptev et al. [2008]. Their HOG and HOF variant use rough orientation binning (4 bins) in a $3 \times 3 \times 2$ grid layout (for more details see section 4.2.2). Their HOG description only uses spatial gradients. In our experiments, we report results for each descriptor separately and combined (HOG-HOF) via concatenation of their feature vectors. Descriptors are computed for the same Harris3D interest points as used for our method.

Datasets. For a better insight into the descriptor's performance, we employ different datasets in our experiments. We perform parameter optimization on the training sets of *KTH* and *Hollywood2* (*cf.* sections 2.2.2 and 2.2.5). Their size proofed to be large enough to optimize the parameter settings. For the comparison to the state-of-the-art, we run additional experiments on *Weizmann* and *YouTube* datasets (*cf.* sections 2.2.1 and 2.2.4).

3.3.2 Parameter learning

In order to determine an appropriate set of parameters for the descriptor introduced in this chapter, we optimize parametric settings on the training data of two datasets: *KTH* actions and *Hollywood2* human actions. Subject to optimization is for *KTH* the *average class accuracy* on the training set obtain via leave-one-person-out cross-validation, i.e., all training sequences belonging to the same person are associated with the same fold. For *Hollywood2*, we optimize the *mean average precision* (mAP) on the training set over all action classes using leave-one-movie-out cross-validation, i.e., all actions coming from the same movie are assigned to one fold.

For parameter learning, the following parameters are optimized jointly: spatial and temporal support (σ_0, τ_0), number of histogram cells (M, N), number of supporting mean gradients (S), orientation type (full or half orientation), quantization type (icosahedron or dodecahedron). We then learn optimal values for the number of spatial and temporal quantization bins (B_θ, B_φ) for spherical coordinates. For this, the previously learned parameters are applied and fixed.

To limit the number of parameters during optimization, we fix the codebook size to $V = 4000$—which has empirically shown good performance over a range of datasets Laptev et al. [2008]. We report values for different sizes after optimization.

For the parameter learning, we divide the parameter space into a rough grid and start at a meaningful manually chosen point. The optimization is a gradient ascent method that evaluates for each parameter its two neighboring values of the current position on the grid. To account for a sometimes significantly large variance, we perform for each point in the parameter space three runs separately. By caching results of previous runs, the approximation of the true mean becomes more precise with each iteration. For each new iteration, the point with the highest mAP among all results previously computed is chosen as the current maximum. The optimization is stopped on convergence, when the maximum remains stable for three consecutive runs.

Table 3.1 summarizes separately for *KTH* and *Hollywood2* the final set of parameters obtained with our optimization strategy. The influence of each parameters evaluated at the optimal settings is shown in figures 3.4 and 3.5. Error bars indicate the standard deviation of several separate runs. In both figures, we show the performance for cross-validation on the training set (denoted as *train*) and, for completness, also on the test set (denoted as *test*).

Overall, we can observe that the parameters most sensitive to changes are the grid layout, the type of quantization as well as the codebook size. The number of spatial grid cells (M)

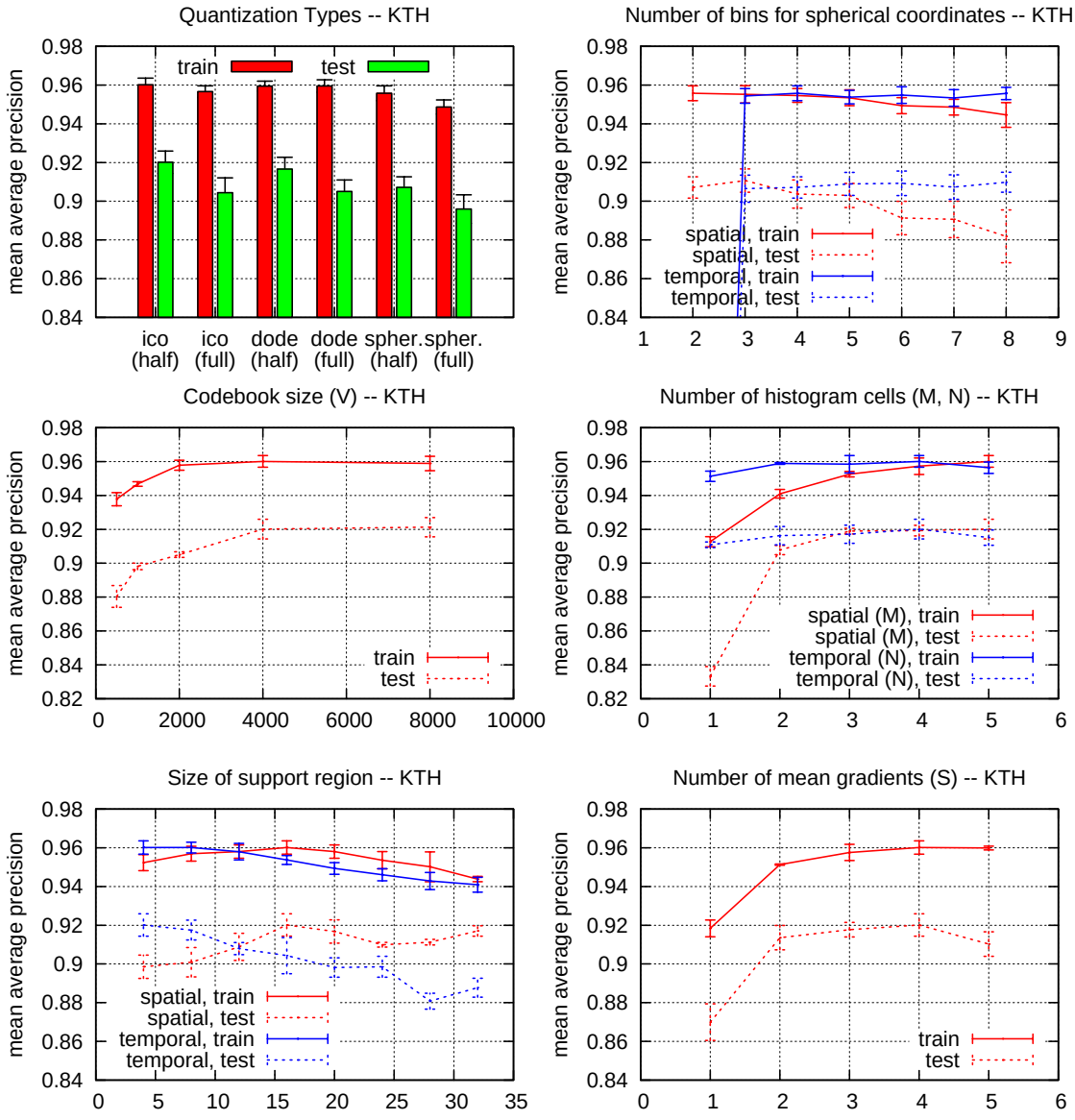


Figure 3.4: Parameter evaluation on the *KTH* dataset for neighboring values around the optimized parameter settings; the average class accuracy on the training set and on the testing set is plotted against different parameter settings, standard deviation denoted by error bars.

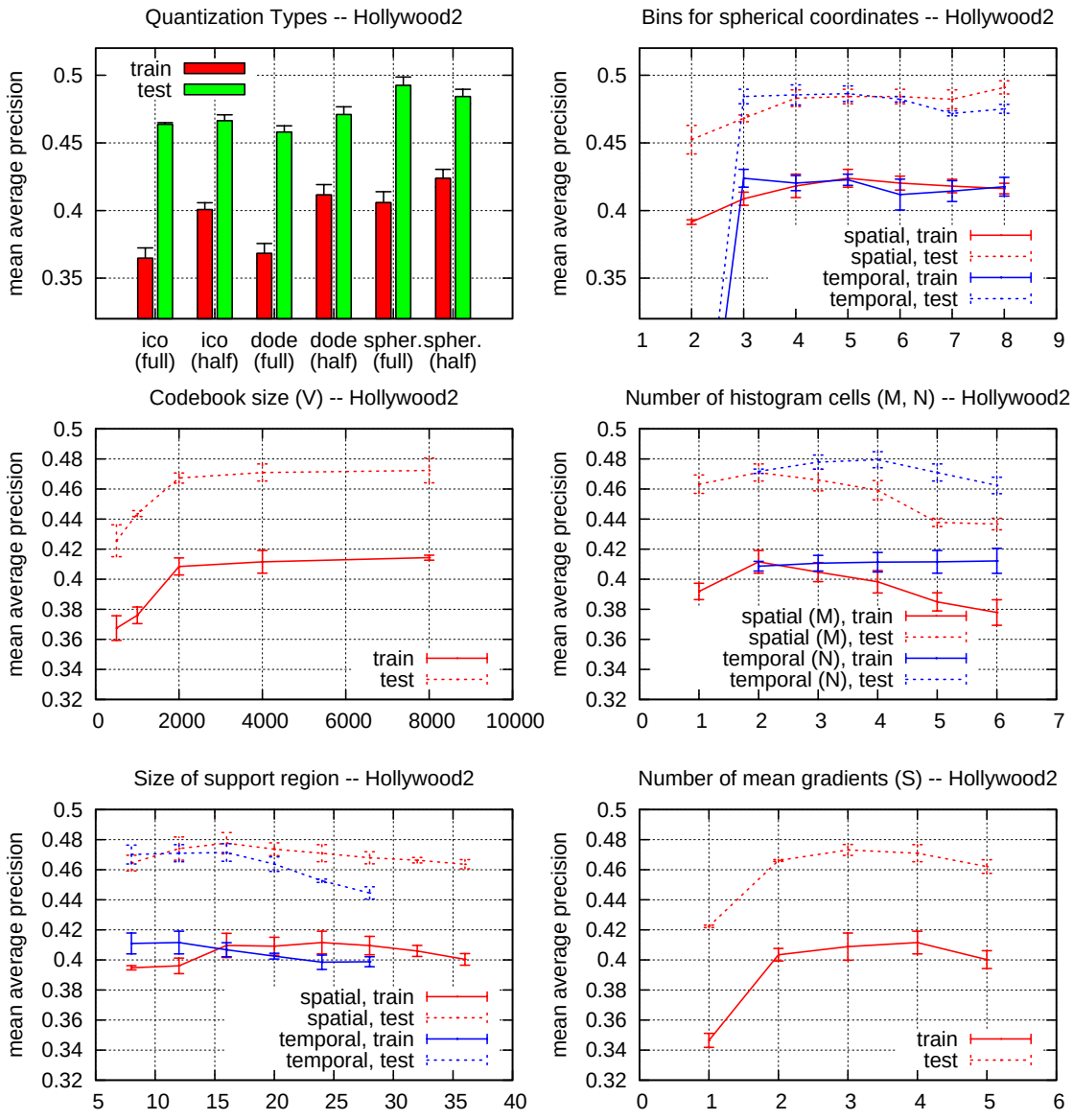


Figure 3.5: Parameter evaluation on the *Hollywood2* dataset for neighboring values around the optimized parameter settings; the mean average precision (mAP) on the training set and on the testing set is plotted against different parameter settings, standard deviation denoted by error bars.

parameter		*KTH*	*Hollywood2*
spatial support	σ_0	16	24
temporal support	τ_0	4	12
number of spatial cells	M	5	2
number of temporal cells	N	4	5
number of mean gradients	S	4	4
orientation type		half	half
quantization type		icosahedron	spherical quant.
spatial coordinate bins	B_θ	–	5
temporal coordinate bins	B_σ	–	3
descriptor dimensionality		1000	300

Table 3.1: Optimized parameter settings obtained separately on *KTH* and *Hollywood2*.

reaches optimal performance on *KTH* for $M = 5$ and on *Hollywood2* for $M = 2$. This can be explained due to the much lower inter- and intra-class variability for the different actions and actors on *KTH*. The finer spatial grid layout presumably helps to distinguish better between different action classes since more spatial information is encoded. As for the number of temporal divisions (N), the grid layout seems relatively stable. On both datasets, results are in favor of a higher number of temporal divisions, presumambly in order to capture more motion information.

The type of quantization varies significantly between both datasets. Where on *KTH* the icosahedron yields best results for quantization, on *Hollywood2* quantization based on spherical coordinates has the edge. For spherical coordinates, interesting to note is that with $B_\theta = 5$ for half spatial orientation and $B_\varphi = 3$ for temporal orientation, more spatial information is encoded in the histogram. Information about velocity of action elements seems thus to play a less important role.

The codebook size as well as the number of mean gradients show across both datasets the most consistent behaviour. The performance increases with increasing codebook size and saturates at about $V = 4000$. The number of mean gradients shows best performance for $S = 4$ which corresponds to $4 \cdot 4 \cdot 4 = 64$ gradient vector votes per histogram cell. For higher values, results drop slightly. Lowest results are obtained for $S = 1$ with only one gradient vote per histogram cell.

On both datasets, one can observe that the scale parameters for the descriptor support size (σ_0, τ_0) seem to favor a smaller temporal ($\tau_0 = [4, 12]$ pixels) than spatial ($\sigma_0 = [16, 24]$ pixels) support. This presumably helps to better describe fast changes in motion. Especially for *KTH*, action classes like jogging, running, and walking can necessite descriptors that are able to distinguish between similar types of motion at different velocities.

3.3.3 Comparison to state-of-the-art

By learning parameter values on two distinct datasets, we obtain settings that are suitable for different types of video sequences: rather simple sequences with homogenous

	KTH	*Weizmann*	*YouTube*	*Hollywood2*
ours (*KTH* optimized)	**92.6%** (±0.2)	**90.7%** (±1.0)	68.3% (±1.0)	45.1% (±0.3)
ours (*Hollywood2* opt.)	89.5% (±0.7)	85.6% (±3.2)	68.1% (±1.2)	**48.6%** (±0.5)
HOGHOF	91.1% (±0.6)	85.6% (±1.1)	**71.2%** (±0.7)	47.7% (±0.1)
HOG	81.9% (±1.1)	75.3% (±3.5)	68.0% (±0.4)	38.2% (±0.2)
HOF	**92.7%** (±0.8)	88.8% (±1.7)	63.9% (±0.5)	43.8% (±0.6)

Table 3.2: Performance comparison over all datasets. Results are shown for our descriptor in combination with Harris3D and our baseline (Harris3D with HOG and HOF descriptors). Performance measure is mean AP for *Hollywood2* and average class accuracy otherwise.

background and low amount of clutter as well as realistic sequences with a large amount of clutter and complex motion patterns. In the following, we apply both settings (denoted as *KTH* and *Hollywood2* optimized) additionally on the *YouTube* and *Weizmann* dataset and compare our results to the baseline (Harris3D with HOG and HOF descriptor) and the state-of-the-art. For the state-of-the-art, we limit our comparison mainly to local features evaluated in a standard bag-of-features (BoF) framework.

Table 3.2 (first column) shows results on the *KTH* dataset. The difference between the two parameter settings are at about 3%. This shows the importance of adapted parameter values. We are on par (92.6%) with the best results of our baseline (HOF, 92.7%). Interesting to note for baseline results is that the combination of shape (HOG) and motion information (HOF) decreases results by 1.6%. This can be explained by the fact that the background is static and actors as well as actions visually similar. In comparison to the state-of-the-art (*cf.* table 2.2), higher results for mere BoF approaches have not been published to the best of our knowledge. Overall best results, irrespective of the method used, have been reported by Gilbert et al. [2009] with 94.5%. They use an approach that incorporates hierarchically context information.

On the *Weizmann* dataset, we outperform (90.7%) results of the baseline (88.8%) with the set of parameters learned on *KTH*. Results are shown in table 3.2 (second column). In these experiments, we employed a smaller codebook size (1000) than for previous ones in order to account for its limited size (we obtain ca. 19,000 interest points in total). This improves results over a codebook size of 4000 (ca. 6% for our descriptor). For the baseline, the HOF descriptor alone yields best results (88.8%) and its combination with HOG degrades performance by 3.2% to 85.6%. The reason is presumably similar to *KTH*: static background and visually similar actions/actors. Among reported results for BoF approaches (*cf.* table 2.1), Liu et al. [2008] obtain the best accuracy known to us (90.4%) by combining and weighting multiple feature types. Our results with only one descriptor type shows comparable performance. Overall best results have been reported with 100% by several authors [Fathi and Mori, 2008, Weinland and Boyer, 2008, Schindler and van Gool, 2008]. All these works employ additional information via foreground masks obtained with background substraction.

	ours *HW2* opt.	ours *KTH* opt.	HOGHOF	HOG	HOF
Eat	55.8% (±1.7)	52.1% (±0.8)	**63.1%** (±0.9)	43.4% (±6.3)	58.6% (±0.9)
Phone	16.3% (±0.6)	**18.0%** (±0.3)	15.3% (±1.0)	11.8% (±0.8)	11.6% (±0.4)
Run	**71.7%** (±0.8)	67.8% (±0.6)	67.2% (±0.2)	62.1% (±0.6)	68.5% (±0.2)
SitDown	47.6% (±0.5)	47.6% (±3.0)	**57.3%** (±1.9)	30.3% (±1.2)	**56.4%** (±0.8)
HugPerson	**47.9%** (±2.2)	32.1% (±4.0)	38.6% (±1.2)	29.6% (±1.1)	30.9% (±1.8)
SitUp	**22.2%** (±0.6)	18.4% (±2.7)	**22.5%** (±2.0)	16.1% (±0.2)	8.5% (±0.4)
GetOutCar	**35.7%** (±2.3)	**35.0%** (±1.7)	32.3% (±1.6)	24.9% (±2.5)	19.6% (±3.4)
DriveCar	**86.3%** (±0.7)	81.6% (±0.1)	**85.8%** (±0.4)	79.0% (±0.4)	84.8% (±0.3)
Kiss	**51.1%** (±1.2)	**49.1%** (±2.0)	**49.3%** (±1.5)	43.5% (±1.0)	45.1% (±1.4)
StandUp	15.6% (±4.4)	**18.6%** (±0.8)	**20.4%** (±2.5)	**20.9%** (±4.7)	**18.9%** (±1.2)
HandShake	**55.7%** (±1.6)	52.1% (±1.5)	49.5% (±1.5)	36.3% (±2.7)	50.2% (±0.9)
Fight	**77.2%** (±0.5)	69.4% (±0.3)	71.3% (±0.5)	60.4% (±0.4)	72.1% (±1.0)
Average	**48.6%** (±0.5)	45.1% (±0.3)	47.7% (±0.1)	38.2% (±0.2)	43.8% (±0.6)

Table 3.3: Average precision on the *Hollywood2* dataset separately for each action class. Results are shown for our descriptor in combination with Harris3D, our baseline (Harris3D with HOG and HOF descriptors).

Results on the *YouTube* dataset (table 3.2 (third column)) show similar performance for both parameter settings of our descriptor (68.1%, 68.3%) and as well for the baseline HOG descriptor (68.0%). Best results (71.2%) are achieved with the baseline's HOG-HOF combination. On this dataset, the HOG-HOF combination improves over both single descriptors and matches the performance that was published by the authors [Liu et al., 2009]. Other results have not been reported in the literature to the best of our knowledge.

For *Hollywood2*, table 3.2 (fourth column) resumes average recognition results and table 3.3 details results per class. Overall, our descriptor parameters learned on the *Hollywood2* training set compare favorably (48.6%) to the baseline (HOG-HOF, 47.7%). Per class, it outperforms the baseline in 5, loses in 2, and is on par in 5 out of 12 action classes. Parameters learned on the training set of *KTH* obtain only 45.1% mAP, i.e., 3.5% lower. Due to its more realistic videos and richer set of action classes, the adaptation of descriptor parameters to this dataset shows to be important. For the baseline, we can note that the HOG-HOF descriptor combination improves performance by 3.9% over the best single descriptor (HOF, 43.8%). Currently best results have been published by Gilbert et al. [2009], as for *KTH*.

3.4 Summary

In this chapter, we have introduced a video descriptor based on histograms of 3D gradient orientations [1]. For this, we have extended the concept of integral images to integral videos for efficient 3D gradient computation, and we have developed a quantization method for 3D orientation based on regular polyhedrons. All descriptor parameters were thoroughly

1. The descriptor software can be downloaded at: http://lear.inrialpes.fr/software.

evaluated and optimized for the task of action recognition in videos. We obtained two optimized parameter settings: one for video sequences with static, homogeneous background and another one for Hollywood-style movies exhibiting complex motions, background clutter, camera ego-motion etc. Finally, the performance of the proposed descriptor was evaluated on a total of four different datasets on which it showed excellent results.

Évaluation de caractéristiques spatio-temporelles locales pour la reconnaissance d'actions

Au cours des dernières années, différentes méthodes pour la détection de points pertinents et pour la description de caracteristiques locales en vidéo ont été proposées dans la littérature (*cf.* section 2.1.3). Toutefois, à cause de limitations et de différences dans les évaluations expérimentales publiées (au niveau des bases de données, des définitions des données d'apprentissage et d'évaluation, des méthodes comparées, des méthodes de classification, etc.), une comparaison équitable de ces méthodes n'est en général pas possible. Afin de permettre une meilleure comparaison, ce chapitre étudie les différentes méthodes pour localiser et décrire des caractéristiques locales dans des vidéos, en se plaçant dans une configuration expérimentale fixe, sur diverses bases de données et avec divers degrés de difficulté.

4

Evaluation of local spatio-temporal features for action recognition

Contents

Over the past years, different methods for feature localization and description in video sequences have been proposed in the literature (*cf.* section 2.1.3). However, given the strongly varying experimental settings under which their evaluations have been carried out, a fair comparison is in general not possible. To allow a better comparison, this chapter studies different methods for localizing and describing local spatio-temporal features in a common experimental setup and on various datasets.

4.1 Introduction

Several different space-time feature detectors [Laptev and Lindeberg, 2003, Dollár et al., 2005, Willems et al., 2008, Jhuang et al., 2007, Wong and Cipolla, 2007, Oikonomopoulos et al., 2006] and descriptors [Laptev et al., 2008, Willems et al., 2008, Kläser et al., 2008, Scovanner et al., 2007, Laptev and Lindeberg, 2004] have been proposed in the past few years. Feature detectors usually select spatio-temporal locations and scales in video by maximizing specific saliency functions. The detectors differ in the type and the sparsity of selected points. Feature descriptors capture shape and motion in the neighborhoods of selected points using image measurements such as spatial or spatio-temporal image gradients and optical flow.

While specific properties of detectors and descriptors have been advocated in the literature, their justification is often insufficient due to the limited and non-comparable experimental evaluations used in current papers. For example, results are frequently presented for different datasets such as the *KTH* dataset [Schüldt et al., 2004, Kläser et al., 2008, Laptev et al., 2008, Willems et al., 2008, Dollár et al., 2005, Wong and Cipolla, 2007, Jhuang et al., 2007], the *Weizmann* dataset [Blank et al., 2005, Scovanner et al., 2007] or the aerobic actions dataset [Oikonomopoulos et al., 2006]. For the common *KTH* dataset [Schüldt et al., 2004], results are often non-comparable due to the different experimental settings used. Schüldt et al. [2004], Kläser et al. [2008], Laptev et al. [2008], Willems et al. [2008] use the standard training/test split of samples defined by Schüldt et al. [2004], other papers [Dollár et al., 2005, Wong and Cipolla, 2007] report results for a simpler leave-one-out setting or a different training and test split [Jhuang et al., 2007]. The comparison is further complicated by the different recognition methods used.

Furthermore, most of the previous evaluations were reported for actions in controlled environments such as in *KTH* and *Weizmann* datasets. It is therefore unclear how these methods generalize to action recognition in realistic setups [Laptev et al., 2008, Rodriguez et al., 2008] which are especially of interest for the present disseration.

A few evaluations of local space-time features have been reported in the past. Laptev [2004] evaluated the repeatability of space-time interest points as well as the associated accuracy of action recognition under changes in spatial and temporal video resolution as well as under camera motion. Similarly, Willems et al. [2008] evaluated repeatability of detected features under scale changes, in-plane rotations, video compression and camera motion. Local space-time descriptors were evaluated by Laptev and Lindeberg [2004], where the comparison included families of higher-order derivatives (local jets), image gradients and optical flow. Dollár et al. [2005] compared local descriptors in terms of image brightness, gradient and optical flow. Scovanner et al. [2007] evaluated the 3D-SIFT descriptor and its two-dimensional variants. Jhuang et al. [2007] evaluated local descriptors in terms of the magnitude and orientation of space-time gradients as well as optical flow. Kläser et al. [2008] compared a spatio-temporal HOG3D descriptor with HOG and HOF descriptors [Laptev et al., 2008]. Willems et al. [2008] evaluated the extended SURF descriptor. In general, however, evaluations in these works are usually limited to a single detection or description method as well as to a single dataset.

The main contribution of this chapter is an evaluation and fair comparison for a number of local space-time detectors and descriptors. We evaluate performance of three space-time interest point detectors and six descriptors along with their combinations on three datasets with varying degree of difficulty and a total of 25 action classes. Moreover, we compare with dense features obtained by regular sampling of local space-time patches, as excellent results were obtained by dense sampling in the context of object recognition [Jurie and Triggs, 2005, Nowak et al., 2006]. We, furthermore, investigate the influence of spatial video resolution as well as shot boundaries on the performance and compare methods in terms of their sparsity. All experiments are reported for the same bag-of-features SVM recognition framework. Among interesting conclusions, we demonstrate that regular sampling consistently outperforms all tested space-time detectors for human actions in realistic setups. We also demonstrate a consistent ranking for the majority of methods across datasets.

4.2 Local spatio-temporal video features

This section describes the local feature detectors and descriptors used in the following evaluation. Methods were selected based on their use in the literature as well as the availability of the implementation. In all cases we use the original implementation and parameter settings provided by the authors.

4.2.1 Detectors

Harris3D. The Harris3D detector was proposed by Laptev and Lindeberg [2003] as a space-time extension of the Harris detector [Harris and Stephens, 1988]. The authors compute a spatio-temporal second-moment matrix at each video point as

$$\mu(\cdot;\,\sigma,\tau) = G(\cdot;\,s\sigma,s\tau) * \begin{pmatrix} L_x^2 & L_xL_y & L_xL_t \\ L_xL_y & L_y^2 & L_yL_t \\ L_xL_t & L_yL_t & L_t^2, \end{pmatrix} \tag{4.1}$$

using independent spatial and temporal scale values σ, τ, a separable Gaussian smoothing function G, and a parameter s that relates the integration scale for G to the local scales σ, τ. The first-order derivatives of the video sequence v are defined as

$$L_x(\cdot;\sigma,\tau) = \partial_x(G * v), \tag{4.2}$$
$$L_y(\cdot;\sigma,\tau) = \partial_y(G * v), \tag{4.3}$$
$$L_t(\cdot;\sigma,\tau) = \partial_t(G * v). \tag{4.4}$$

The final locations of space-time interest points are given by local maxima of

$$H = \det(\mu) - k\,\mathrm{trace}^3(\mu),\ H > 0. \tag{4.5}$$

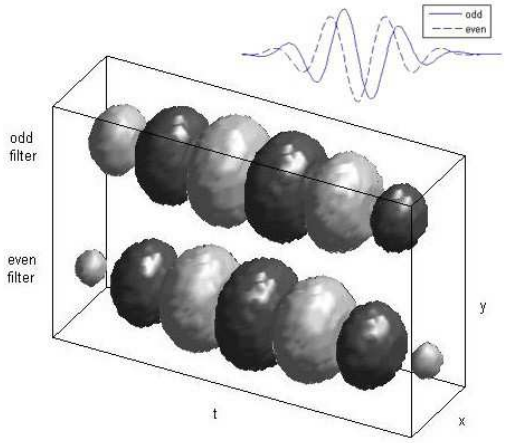


Figure 4.1: Illustration of the response function for interest point detection proposed by Dollár et al. [2005] and given in equation (4.7) (courtesy of Dollár et al. [2005]).

The authors proposed also a mechanism for spatio-temporal scale selection based on the scale-normalized spatio-temporal Laplacian operator:

$$\nabla^2 L = \sigma^2 \tau^{1/2}(L_{xx} + L_{yy}) + \sigma \tau^{3/2} L_{tt}. \tag{4.6}$$

Final interest points are required to be local maxima with respect to the Harris cornerness criterion, i.e., equation 4.5, as well as to be local extrema with respect to the scale normalized Laplacian operator. Following Laptev et al. [2008], we do not perform scale selection, but we use points extracted at multiple scales based on a regular sampling of the scale parameters σ, τ. We use the original implementation available on-line [1] and standard parameter settings $k = 0.0005$, $\sigma^2 = 4, 8, 16, 32, 64, 128$, $\tau^2 = 2, 4$. Figure 4.3(second row) shows example detections on consecutive video frames.

Gabor. The Gabor detector is based on temporal Gabor filters and was proposed by Dollár et al. [2005]. The response function is given by

$$R = (I * G * h_{ev})^2 + (I * G * h_{od})^2, \tag{4.7}$$

with a 2D spatial Gaussian smoothing kernel $G(x, y; \sigma)$ and a quadrature pair of 1D Gabor filters h_{ev} and h_{od} which are applied temporally. The Gabor filters are defined by

$$h_{ev}(t; \tau, \omega) = -\cos(2\pi t\omega) e^{-t^2/\tau^2} \tag{4.8}$$

$$h_{od}(t; \tau, \omega) = -\sin(2\pi t\omega) e^{-t^2/\tau^2} \tag{4.9}$$

with $\omega = 4/\tau$. Figure 4.1 illustrates the response function. The two parameters σ and τ of the response function R correspond roughly to the spatial and temporal scale of the

1. http://www.irisa.fr/vista/Equipe/People/Laptev/download.html#stip

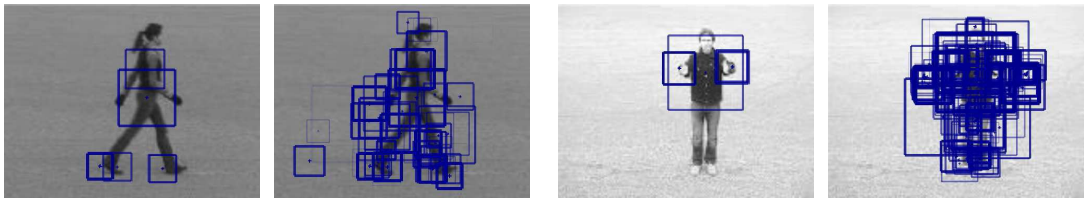


Figure 4.2: Illustration of spatio-temporal interest points detected using the Hessian saliency measure used by Willems et al. [2008] for different thresholds (courtesy of Willems et al. [2008]).

detector. Interest points are the local maxima of the response function R. We use the code from the authors' website [2] and detect features using standard scale values $\sigma = 2, \tau = 4$. Figure 4.3(third row) shows example detections on consecutive video frames.

Hessian3D. The Hessian3D detector was proposed by Willems et al. [2008] as a spatio-temporal extension of the Hessian saliency measure used by Beaudet [1978] and Lindeberg [1998] for blob detection in images. The detector measures the saliency with the determinant of the 3D Hessian matrix:

$$H(\cdot;\sigma,\tau) = \begin{pmatrix} L_{xx} & L_{xy} & L_{xt} \\ L_{yx} & L_{yy} & L_{yt} \\ L_{tx} & L_{ty} & L_{tt} \end{pmatrix}. \tag{4.10}$$

The strength of each interest point at a certain scale is given by the determinant of its Hessian matrix $|\det(H)|$. For the case of perfect Gaussian blobs, the determinant can be approximated with its first term as $\det(H) \approx L_{xx}L_{yy}L_{tt}$. By using the scale-normalized spatio-temporal Laplacian, Willems et al. localize final interest points in the 5D scale space as local maxima of

$$S = \sigma^{2p}\tau^{2q}L_{xx}L_{yy}L_{tt}. \tag{4.11}$$

In order to speed up the detector, the authors used approximative box-filter operations on an integral video structure. Responses for different scales are computed by upscaling the box-filters. The determinant of the Hessian is computed over several octaves of both the spatial and temporal scales. A non-maximum suppression algorithm selects joint extrema over space, time and scales: (x, y, t, σ, τ). Figure 4.2 illustrates some detected interest points for different thresholds. We use the executables from the authors' website [3] and employ the default parameter setting. Figure 4.3(fourth row) shows example detections on consecutive video frames.

Dense sampling. Dense sampling extracts video blocks at regular positions and scales in space and time. There are 5 dimensions to sample from: (x, y, t, σ, τ), where σ and

2. http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html
3. http://homes.psat.kuleuven.be/~gwillems/research/Hes-STIP/

Figure 4.3: Visualization of interest points detected by the different detectors: Harris3D (second row), Gabor (third row), Hessian3D (fourth row).

τ are the spatial and temporal scale, respectively. After evaluating different spatial patch sizes for dense sampling (*cf.* section 4.3.7), we set for our experiments the minimum size of a 3D patch to 18×18 pixels and 10 frames. Spatial and temporal sampling are done with 50% overlap. Multi-scale patches are obtained by multiplying σ and τ by a factor of $\sqrt{2}$ for consecutive scales. In total, we use 8 spatial and 2 temporal scales since we consider the spatial scale to be more important than the time scale. We consider all combinations of spatial and temporal scales, i.e., we sample a video 16 times with different σ and τ parameters.

4.2.2 Descriptors

For each given sample point (x, y, t, σ, τ), a feature descriptor is computed for a 3D video patch centered at (x, y, t). Its spatial size $\Delta_x(\sigma), \Delta_y(\sigma)$ is a function of σ and its temporal length $\Delta_t(\tau)$ a function of τ.

Gradient. Dollár et al. [2005] proposed the Gradient descriptor along with the Gabor detector. The size for the descriptor is given by

$$\Delta_x(\sigma) = \Delta_y(\sigma) = 2 \cdot \text{ceil}(3\sigma) + 1, \tag{4.12}$$

$$\Delta_t(\tau) = 2 \cdot \text{ceil}(3\tau) + 1. \tag{4.13}$$

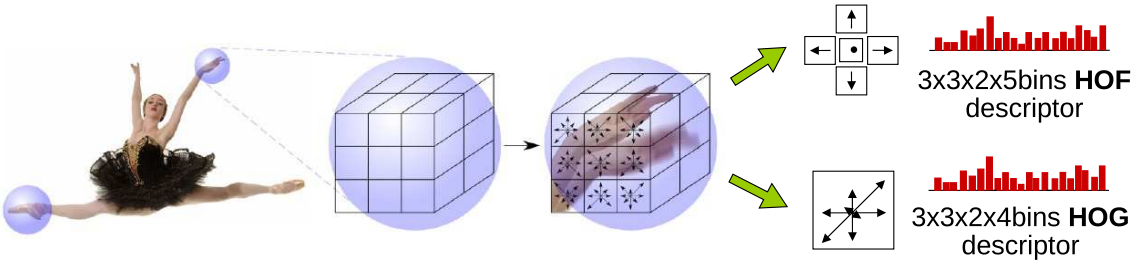


Figure 4.4: Illustration of the HOG/HOF descriptor: an interest point is described by a cuboid region divided into a grid of cells; for each cell, a histogram of oriented spatial gradients (HOG) as well as a histogram of optical flow (HOF) is computed; for the final descriptor, all cell HOG and HOF descriptors are concatenated (courtesy of Laptev et al. [2008]).

We follow the authors' setup and concatenate the gradients computed for each pixel in the patch into a single vector. Principal component analysis (PCA) is computed on the training samples and is used to project the feature vector to a lower dimensional space. The descriptor size after PCA projection is 100. We download the code from the authors' website [2] and use its default settings.

HOG/HOF. The HOG and HOF descriptors were introduced by Laptev et al. [2008]. To characterize local motion and appearance, the authors combine histograms of oriented spatial gradients (HOG) and histograms of optical flow (HOF) in a late fusion approach. The histograms are accumulated in the space-time neighborhood of detected interest points, where the descriptor region is given by a cuboid of the size $\Delta_x(\sigma) = \Delta_y(\sigma) = 18\sigma$ and $\Delta_t(\tau) = 8\tau$. Each cuboid region is subdivided into a $n_x \times n_y \times n_t$ grid of cells; for each cell, 4-bin HOG histograms and a 5-bin HOF histogram (with 4 directions and an additional zero-bin) are computed. The normalized cell histograms are concatenated into the final HOG and HOF descriptor. We investigate in our experiments the performance of the combined HOG/HOF descriptor (by concatenation) as well as its HOG and HOF parts. In our evaluation we used the grid parameters $n_x, n_y = 3$, $n_t = 2$ as suggested by the authors. We use the original implementation available on-line [1].

For computing HOG/HOF descriptors with scale parameters σ, τ returned by the Hessian3D detector, we optimize Δ_x, Δ_y to yield best performance. Our final cuboid size is then given by $\Delta_x(\sigma) = \Delta_y(\sigma) = 13\sigma$ and $\Delta_t(\tau) = 8\tau$. The Gabor detector computes interest points only for a single spatio-temporal scale. For it combination with the HOG/HOF descriptor, we fix the region size to $\Delta_x = \Delta_y = 36$ and $\Delta_t = 11$.

HOG3D. The HOG3D descriptor was proposed originally in [Kläser et al., 2008] and further extended in chapter 3. It is based on histograms of 3D gradient orientations and can be seen as an extension of the SIFT descriptor [Lowe, 2004] to video sequences. Gradients are computed using an integral video representation. Both, regular polyhedrons and spherical coordinates are used to quantize the orientation of spatio-temporal gradients.

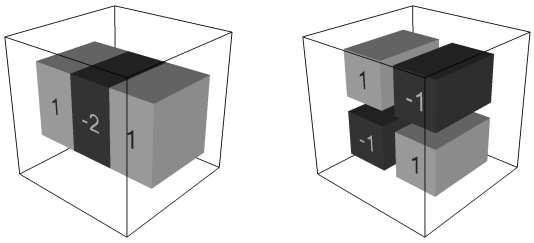


Figure 4.5: Two types of box filter approximations for the Gaussian second order partial derivatives employed by Willems et al. [2008] (courtesy of Willems et al. [2008]).

The descriptor therefore describes shape and motion information at the same time. A given 3D patch is divided into $n_x \times n_y \times n_t$ cells. The corresponding descriptor concatenates 3D gradient histograms of all cells which are normalized separately. The executable is available on-line [4].

For this descriptor, two different parametric settings were proposed (*cf.* section 3.3.2). The first one has been obtained via optimization on the training set of *KTH* (in the following denoted by HOG3D[1]). It is applicable to more controlled datasets containing video sequences with a static and rather homogeneous background. The descriptor size is given as $\Delta_x(\sigma) = \Delta_y(\sigma) = 16\sigma, \Delta_t(\tau) = 4\tau$. The number of spatial and temporal cells is $n_x = n_y = 5, n_t = 4$, and icosahedron with half orientation is used as polyhedron type for quantizing orientations. The resulting dimensionality of the descriptor is $5 \cdot 5 \cdot 4 \cdot 10 = 1000$. We employ these settings for our experiments on the *KTH* and *UCF* dataset.

The second setting was learned on the training set of *Hollywood2* (denoted as HOG3D[2]in our experiments). This set of parameters is adapted to datasets that include more challenging type of video data featuring cluttered background, complex motion patterns, camera ego-motion, and a large variety of actions. The descriptor size is defined by $\Delta_x(\sigma) = \Delta_y(\sigma) = 24\sigma$ and $\Delta_t(\tau) = 12\tau$. The number of spatial and temporal cells is $n_x = n_y = 2, n_t = 5$, and spherical coordinates for half orientation with 5 spatial and 3 temporal bins are used for orientation quantization. The resulting dimensionality of the descriptor is $2 \cdot 2 \cdot 5 \cdot 10 = 300$. This set of parameters is applied for our experiments on the *Hollywood2* dataset.

Extended SURF. Willems et al. [2008] proposed the extended SURF (ESURF) descriptor which extends the image SURF descriptor [Bay et al., 2006] to videos. Like for previous descriptors, the authors divide 3D patches into $n_x \times n_y \times n_t$ cells. The size of the 3D patch is given by $\Delta_x(\sigma) = \Delta_y(\sigma) = 3\sigma$ and $\Delta_t(\tau) = 3\tau$. For the feature descriptor, each cell is represented by a vector of weighted sums $v = (\sum d_x, \sum d_y, \sum d_t)$ of uniformly sampled responses of the Haar-wavelets d_x, d_y, d_t along the three axes (illustration in figure 4.5). We use the executables from the authors' website [3] with the default parameters defined in the executable.

4. http://lear.inrialpes.fr/software

	HOG3D[1]	HOG/HOF	HOG	HOF	Gradient	ESURF
Harris3D	**92.4%**	**91.8%**	80.9%	**92.1%**	–	–
Gabor	91.4%	88.7%	82.3%	88.2%	89.1%	–
Hessian3D	88.1%	88.7%	77.7%	88.6%	–	81.4%
Dense	88.5%	86.1%	79.0%	88.0%	–	–

Table 4.1: Average accuracy for various detector/descriptor combinations on *KTH* actions.

4.3 Experimental results

This section presents experimental results for various detector/descriptor combinations. We start with the details for our experimental setup (section 4.3.1). Results are presented for the different datasets in sections 4.3.2-4.3.4. Sections 4.3.5-4.3.8 evaluate the influence of shot boundaries, the influence of subsampling, different parameters for dense sampling, and compare the density of the different detection methods.

4.3.1 Experimental setup

For the experiments, we evaluate the different features in a bag-of-features based action classification task. The exact experimental setup follows the description of section A.1. We employ k-means for vocabulary construction and fix the codebook size to 4000. Due to random initialization of k-means used for codebook generation, we observed a standard deviation of approximately 0.5% in our experiments.

We carry out experiments on three different action datasets: *KTH*, *UCF* sports, and *Hollywood2* actions datasets. We follow the original experimental setups of the authors as described in section 2.2. For the evaluation, we report average accuracy over all classes for the *KTH* and *UCF* dataset and mean average precision (mAP) over all classes for the *Hollywood2* dataset.

Due to high memory requirements of some descriptor/detector code, we subsample original *UCF* and *Hollywood2* sequences to half spatial resolution in all our experiments. This enables us to compare all methods on the same data. We evaluate the effect of subsampling for the *Hollywood2* data set in section 4.3.4. The ESURF and Gradient descriptors are not evaluated for other detectors than those used in original papers. Unfortunately, separate implementations of these descriptors were not available.

4.3.2 KTH actions dataset

Our results for different combinations of detectors and descriptors evaluated on *KTH* are presented in table 4.1. Overall, the best results are obtained with Harris3D as interest point detector and HOG3D, HOF, as well as HOG/HOF for description. This is less surprising considering the fact that both descriptors, HOG/HOF and HOG3D, have been engineered to work well with this detector.

	HOG3D[1]	HOG/HOF	HOG	HOF	Gradient	ESURF
Harris3D	77.6%	78.1%	71.4%	75.4%	–	–
Gabor	**85.0%**	77.7%	72.7%	76.7%	76.6%	–
Hessian3D	78.9%	79.3%	66.0%	75.3%	–	77.3%
Dense	**84.8%**	81.6%	77.4%	**82.6%**	–	–

Table 4.2: Average accuracy for various detector/descriptor combinations on the *UCF* dataset.

Among the detector/descriptor combinations, best results are obtained for Harris3D + HOG3D (92.4%). This is a clear improvement over the results previously reported in [Wang et al., 2009] due to updated parameter settings as given in chapter 3. Comparable results are achieved with Harris3D + HOF (92.1%) and HOG/HOF (91.8%) which match the 91.8% published in [Laptev et al., 2008] for Harris3D + HOG/HOF.

For the Gabor detector, the best result (91.4%) is obtained with the HOG3D descriptor. In its combination with the Gradient descriptor, we reach 89.1% which is significantly higher than published in the original work by Dollár et al. [2005] (81.2%). This is presumably due to their different classification method (SVM with RBF kernel).

The performance of Hessian3D and Dense detectors are below Harris3D and Gabor. Our results for Hessian3D with ESURF are ca. 3% below the performance as reported by Willems et al. [2008]. In contrast to our BoF implementation, the authors employed a soft voting strategy to build BoF histograms. The low performance of dense sampling on *KTH* may be explained by the large number of features corresponding to the static background. The large number of uninformative background features may have an unfavorable influence on the distance computation. For a comparison with the state-of-the-art, see section 2.2.2.

4.3.3 UCF sports dataset

The results for different combinations of detectors and descriptors evaluated on *UCF* sport actions are illustrated in table 4.2. The best result over different detectors is obtained by with Gabor detector (85.0%) and dense sampling (84.8%). For dense features, this can be explained by the fact that they capture different types of motions as well as background which may provide useful context information. Scene context information can indeed help for to classifiy sports actions which often involve specific equipment and scene types as illustrated in figure 2.17. The Gabor detector is, compared to the other two feature detectors, the one that provides the densest number of features (*cf.* section 4.3.8). As can be seen in figure 4.3(bottom), features include more background and thus context information than for the other detectors.

Also above 80% are dense points in combination with HOG/HOF and HOF. Harris3D and Hessian3D detectors perform similar at the level of 80%. Among different descriptors, HOG3D provides best results for Gabor and dense sampling and is on par with HOG/HOF for Harris3D and Hessian3D. The authors of the original paper, Rodriguez et al. [2008],

	HOG3D[2]	HOG/HOF	HOG	HOF	Gradient	ESURF
Harris3D	44.3%	45.2%	32.8%	43.3%	–	–
Gabor	**46.1%**	**46.2%**	39.4%	42.9%	45.0%	–
Hessian3D	43.5%	**46.0%**	36.2%	43.0%	–	38.2%
Dense	44.8%	**47.4%**	39.4%	45.5%	–	–

Table 4.3: Mean AP for various detector/descriptor combinations on the *Hollywood2* dataset.

report 69.2% for this dataset. However, note that their result does not correspond to the version of *UCF* dataset available on-line (*cf.* section 2.2.3) used in our evaluation.

4.3.4 Hollywood2 dataset

Evaluation results for *Hollywood2* actions are presented in table 4.3. As for the *UCF* dataset, the best results are obtained for dense sampling (47.4%) and the Gabor detector (46.1% for HOG3D and 46.2% for HOG/HOF). In addition to this, the Hessian3D detector achieves in combination with HOG/HOF results also comparable results (46.0%). We assume dense sampling and Gabor again benefits from a more complete description of motions and the rich context information.

Among the different evaluated descriptors, HOG/HOF performs best. Unlike in results for *KTH* actions, here the combination of HOF and HOG improves over HOF by about 2 percent. The HOG3D descriptor performs best in combination with the Gabor detector and interestingly performs worse in combination with dense sampling. Still dense sampling performs sligthly better than Harris3D on which the descriptor parameters were optimized.

4.3.5 Shot boundary features

Since action samples in *Hollywood2* are collected from movies, they contain many shot boundaries which cause artificial interest point detections. To investigate the influence of shot boundaries on recognition results, we compare in table 4.4 the performance of the Harris3D detector with and without shot boundary features. Results for HOG demonstrate 2% improvement when removing shot boundar features, and changes for HOG/HOF are negligible. HOG3D shows a significant performance drop without using features at shot boundaries. This can have to do with its parameter optimization that included features at shot boundary positions. In fact, shot boundaries hold context information that can help classification. Given these results, we can conclude that shot boundary features do not harm action classification.

4.3.6 Influence of subsampling

We also investigate the influence of reduced spatial resolution adopted in our *Hollywood2* experiments. In table 4.4 recognition results are reported for videos with full and half

	HOG3D[2]	HOG/HOF	HOG	HOF
reference	44.3%	45.2%	32.8%	43.3%
w/o shot boundary features	42.1%	45.7%	35.3%	43.4%
full resolution videos	**48.8%**	47.6%	39.7%	43.9%

Table 4.4: Comparison of the Harris3D detector on (top) videos with half spatial resolution, (middle) with removed shot boundary features, and (bottom) on the full resolution videos.

Spatial Size	*Hollywood2*				*UCF*			
	HOG3D[2]	HOG/HOF	HOG	HOF	HOG3D[1]	HOG/HOF	HOG	HOF
18 × 18	44.8%	**47.4%**	**39.4%**	**45.5%**	84.8%	**81.6%**	**77.4%**	82.6%
24 × 24	**46.0%**	**47.7%**	**39.4%**	**45.8%**	**86.1%**	**81.4%**	**76.8%**	**84.0%**
36 × 36	**46.1%**	**47.3%**	**36.8%**	**45.6%**	83.2%	79.1%	76.5%	82.4%
48 × 48	44.4%	46.5%	35.8%	**45.5%**	81.7%	78.6%	73.9%	79.0%
72 × 72	42.2%	45.2%	32.2%	43.0%	78.7%	78.8%	69.6%	78.4%

Table 4.5: Average accuracy for dense sampling with varying minimal spatial sizes on the *Hollywood2* and *UCF* sports dataset.

spatial resolution using the Harris3D detector. The performance is consistently and significantly increased for all tested descriptors for the case of full spatial resolution. Especially the HOG3D detector shows a large gain in this experiment and sligthly outperforms HOG/HOF (by 1.2%). Note that for full resolution, we obtain approximately 3 times more features per sequence than for half resolution.

4.3.7 Dense sampling parameters

Given the best results obtained with dense sampling, we further investigate the performance as a function of different minimal spatial sizes of dense descriptors (cf. table 4.5). As before, further spatial scales are sampled with a scale factor of $\sqrt{2}$. As in sections 4.3.3 and 4.3.4, we present results for *Hollywood2* and *UCF* videos with half spatial resolution. We observed no significant improvements for different temporal lengths, therefore we fixed the temporal length to 10 frames. The overlapping rate for dense patches is set to 50%. We can see that the performance increases with smaller spatial size, i.e., when we sample denser. However, the performance saturates in general at a spatial size of 24×24 for *Hollywood2* and 18×18 for *UCF*.

4.3.8 Feature density

We compare the tested detectors by the number of detected interest points. The comparison was performed on a set of videos from *Hollywood2* with spatial resolution of 360×288 pixels (half resolution) and about 8000 frames length in total. Table 4.6 presents results for the three detectors and dense sampling in terms of average number of features per frame. Among the detectors, Gabor extracts the densest features (44 features/frame) and

	Harris3D	Hessian3D	Gabor	Dense
Features/frame	31	19	44	643

Table 4.6: Average number of generated features for different detectors.

Hessian3D extracts the sparsest set features (19 features/frame). Obviously, dense sampling extracts many more features than interest point detectors, for this particular setup roughly 20 times more features are extracted than for the interest point detectors.

4.4 Conclusion

Among the main conclusions, we note that dense sampling overall outperforms interest point detectors in realistic video settings, but performs worse on the simple *KTH* dataset. This indicates both (a) the importance of using realistic experimental video data as well as (b) the limitations of current interest point detectors. Note, however, that dense sampling also produces a very large number of features (usually 15-20 times more than feature detectors). This is more difficult to handle than the relatively sparse number of interest points. We also note a rather similar performance of interest point detectors for each dataset. Across datasets, Harris 3D performs better on *KTH* dataset, while the Gabor detector gives better results for *UCF* and *Hollywood2* datasets.

Among the tested descriptors, the combination of gradient based and optical flow based descriptors seems to be a good choice. The combination of dense sampling with the HOG/HOF descriptor provides best results for the most challenging *Hollywood2* dataset. On the *UCF* dataset, the HOG3D descriptor performs best in combination with dense sampling as well as with the Gabor detector. On *KTH*, both descriptors, HOG3D and HOG/HOF, show comparable results, with HOG3D having a slight edge. This also motivates further investigations of optical flow based descriptors.

La reconnaissance d'actions à l'aide de trajectoires locales

Dans le chapitre précédent, nous avons évalué différents détecteurs et descripteurs de caractéristiques locales. Tous les détecteurs que nous avons étudiés sont basés sur des critères de pertinence spatio-temporelle afin de détecter des points d'intérêt dans des vidéos. Comme approche plus intuitive pour les vidéos, nous proposons dans ce chapitre une représentation de caractéristiques locales basée sur des trajectoires. Contrairement aux points d'intérêt spatio-temporels, les trajectoires permettent une représentation plus adaptée pour la vidéo et elles sont en mesure de bénéficier de l'information de mouvement, car elles suivent le mouvement des points locaux (*cf.* figure 5.1).

5

Action recognition with feature trajectories

Contents

In the previous chapter, we have evaluated various feature detectors and descriptors. All of the feature detectors that we investigated are based on spatio-temporal saliency criteria to detect interesting 3D positions in video. As a more intuitive approach to videos, we propose in this chapter a local feature representation for video sequences based on feature trajectories. In contrast to spatio-temporal interest points, feature trajectories allow for a more adapted representation and are able to benefit from the rich motion information captured by the trajectories (*cf.* figure 5.1).

Figure 5.1: Feature trajectories for sample actions of the *Hollywood2* dataset. Left column: sample sequence for the action "StandUp" where the person on the right side stands up, and the trajectories accurately capture the body motion. Right column: sample frames from a "Kiss" action. The motion of two persons approaching each other can be clearly deduced from the trajectories. Red dots indicate trajectory position in the current frame.

5.1 Introduction

Tracking is a natural way of capturing moving objects, and it is widely used for motion analysis [Gavrila, 1999]. Many traditional approaches in action recognition are based on tracking human body models or segmenting human silhouettes [Blank et al., 2005]. However, tracking humans in realistic video settings is difficult and prone to errors: object parts may be occluded or simply out of view, and actions can contain strong and abrupt motions that make tracking infeasible. Local feature trajectories combine the concept of local features with traditional tracking approaches which makes them suitable for realistic videos.

A significant amount of research has been devoted to action recognition using trajectory information [Moeslund et al., 2006] (*cf.* section 2.1.1). Some recent methods [Messing et al., 2009, Sun et al., 2009, Matikainen et al., 2009, 2010] show promising results on challenging

human actions datasets by employing trajectories as local features (*cf.* section 2.1.3). While these approaches only use the shape information of feature trajectories, we propose to augment the trajectory description by additionally appearance and motion information in the local neighborhood surrounding the trajectory. For this, we introduce a novel local descriptor based on histograms of motion boundaries. The final descriptor is based on a combination of histograms of oriented gradients (HOG) to encode appearance information and histograms of optical flow (HOF) as well as motion boundary histograms (MBH) to encode motion information. Employing our full trajectory descriptor in a standard BoF representation, we evaluate its parameters and demonstrate a significant improvement for video classification. We outperform the current state-of-the-art on two benchmark datasets and are on par for a third one.

5.2 Feature trajectory description

5.2.1 Extraction of feature trajectories

Feature trajectories are obtained with a pyramidal implementation [Bouguet, 1999] of the Lucas-Kanade feature tracker [Lucas and Kanade, 1981]. Interest points are extracted with the detector proposed by Shi and Tomasi [1994] at multiple spatial scales $\{\sigma_i\}$ with $\sigma_0 = 1$ and $\sigma_{i+1} = \sqrt{2} \cdot \sigma_i$. We fix the number of spatial scales to 8. For both, the interest point detector as well as the feature tracker, we use the implementations provided by the OpenCV library [1].

For a given frame, we track trajectories from the previous frame [Bouguet, 1999]. Furthermore, we detect additional interest points [Shi and Tomasi, 1994], but discard points that lie in the direct neighborhood (i.e., , with a distance smaller than 3 pixels) of an existing trajectory. All remaining points are added as new trajectory seeds to the tracking process. Since trajectories tend to drift away from their original position over time, we limit the length of a trajectory to L frames. As soon as the trajectory length exceeds L, it is removed from the tracking process. Consequently, this allows new interest points in its neighborhood to be detected and tracked again.

Since, for action recognition, we are mainly interested in dynamic information of a video sequence, static trajectories are pruned in a pre-processing stage. Trajectories with a sudden large displacement, most likely to be erroneous, are also removed.

5.2.2 Trajectory descriptor

To encode shape and motion information surrounding the local neighborhood of a given feature trajectory, we extend the trajectory shape descriptor by appearance and motion information. To this end, descriptors based on gradient (HOG), optical flow (HOF), and motion boundary information (MBH) are computed. The process of feature extraction in the vicinity of the trajectory is shown in figure 5.2 (right). We detail the computation of descriptors in the following.

1. http://opencv.willowgarage.com/wiki/

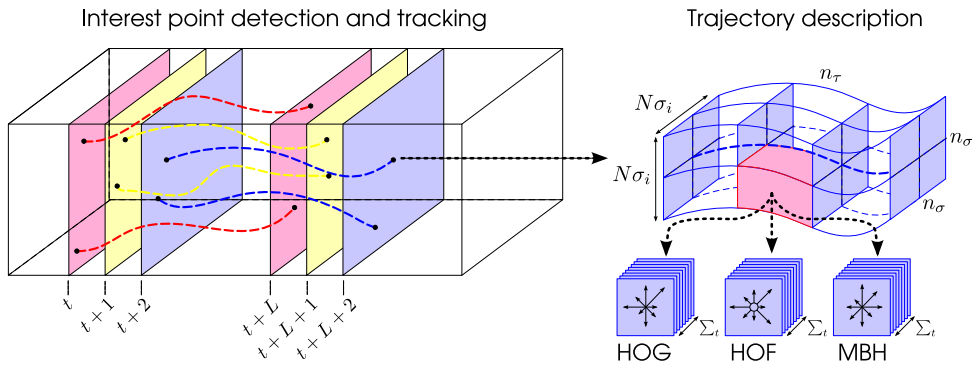


Figure 5.2: An overview of the feature trajectory description. Interest points are detected and tracked at multiple spatial scales $\{\sigma_i\}$. New interest points are detected in each frame and their trajectory is limited to a length of L frames. The description of the trajectory shape is encoded by its displacement vectors. Static as well as motion appearance are described by histograms of oriented gradients (HOG), histograms of optical flow (HOF), and motion boundary histograms (MBH). Given a trajectory extracted for spatial scale σ_i, descriptors are computed for a supporting neighborhood of $N \cdot \sigma_i \times N \cdot \sigma_i$ pixels along the trajectory. The trajectory neighborhood is split into a spatio-temporal grid of size $n_\sigma \times n_\sigma \times n_\tau$.

Trajectory shape. The shape of a trajectory encodes local motion patterns. For a given trajectory and a fixed length L, we describe its shape at time t by a sequence $s = (\Delta\mathbf{x}_t, \ldots, \Delta\mathbf{x}_{t+L-1})$ of displacement vectors $\Delta\mathbf{x}_j$ with $\Delta\mathbf{x}_j = \mathbf{x}_{j+1} - \mathbf{x}_j$ and $\mathbf{x}_j = (x_j, y_j)$. The resulting vector is normalized by the sum over the magnitudes of its displacement vectors:

$$s' = \frac{(\Delta\mathbf{x}_t, \ldots, \Delta\mathbf{x}_{t+L-1})}{\sum_{j=t}^{t+L-1} ||\Delta\mathbf{x}_j||}. \tag{5.1}$$

Appearance and motion description. Static appearance information as well as motion information provide important cues for recognizing actions [Bobick and Davis, 2001, Jhuang et al., 2007, Schindler and van Gool, 2008, Laptev et al., 2008]. We propose to augment the trajectory description with histograms of oriented gradients (HOG) [Dalal and Triggs, 2005], histograms of optical flow (HOF) [Laptev et al., 2008], and motion boundary histograms (MBH) [Dalal et al., 2006] (*cf.* figure 5.3). Motion boundary histograms were introduced in the context of pedestrian detection in video sequences to capture motion information. The MBH description separates the optical flow field I_x, I_y into its x and y component and computes for both I_x and I_y a separate HOG descriptor. Since it represents the gradient of the optical flow, constant motion information—and thus also camera ego motion—is suppressed and only information on changes of the flow field (i.e., motion boundaries) is kept (see figure 5.3 (right)). Therefore, the MBH descriptor can be seen as complementary to HOG and HOF descriptors.

A description is computed for a space-time volume around a feature trajectory where we align the volume at each frame with the feature trajectory, see figure 5.2 (right). Given

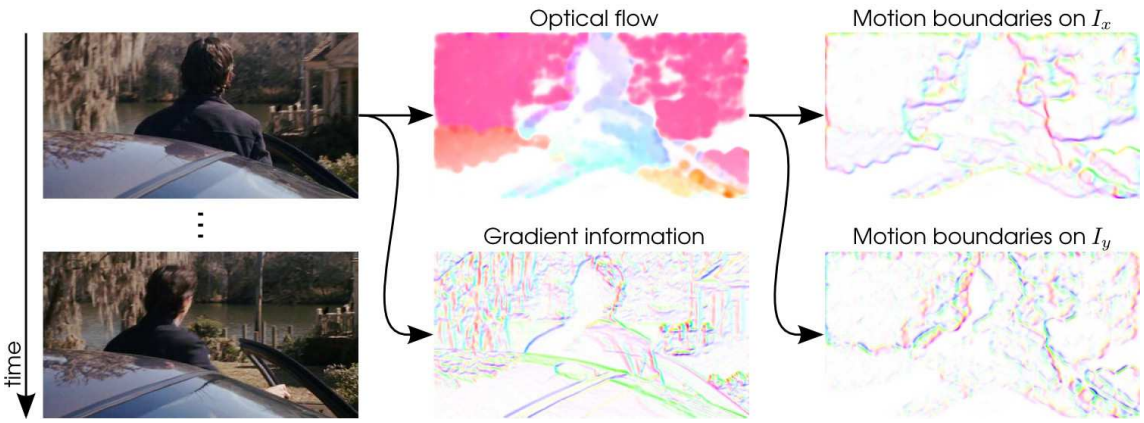


Figure 5.3: Illustration of the information captured by HOG, HOF, and MBH descriptors. For each image, gradient/flow orientation is indicated by color (hue) and magnitude is indicated by saturation. Motion boundary information is computed as gradient information separately on the x and y flow components. Compared to optical flow, motion boundaries suppress most camera motion in the background and highlight foreground motion.

a trajectory for a spatial scale σ_i, i.e., its initial interest point is detected at this scale, descriptors are computed for a support region of $N \cdot \sigma_i \times N \cdot \sigma_i$ pixels along the trajectory, as illustrated in figure 5.2. The support region is split into a spatio-temporal grid of size $n_\sigma \times n_\sigma \times n_\tau$. We evaluate parameters (*cf.* section 5.3.2) and use for our experiments $N = 32, n_\sigma = 2, n_\tau = 3$.

For each grid cell, HOG, HOF, and MBH histograms are extracted over all contributing frames (*cf.* figure 5.2 (right, bottom)). For all descriptors, orientations are quantized into 8 bins using full orientation, with an additional zero bin for HOF (i.e., in total 9 bins). The three descriptors are separately normalized with their L2 norm. The final descriptor is the concatenation of the HOG, HOF, and MBH descriptors with the trajectory shape descriptor.

5.3 Experimental results

Before discussing the results, we detail our experimental setup along section 5.3.1. A study of descriptor parameters is then given in section 5.3.2. Section 5.3.3 shows final results on three benchmark datasets and compares them to the state-of-the-art in section 5.3.4.

5.3.1 Experimental setup

In order to evaluate the performance of our descriptor, we use the bag-of-features representation as presented in section A.1. The visual vocabulary is created using k-means with the number of visual words fixed to 4000. As baseline for comparison, we employ spatio-temporal 3D Harris points in combination with HOG-HOF features as detailed in sections 4.2.1 and 4.2.2.

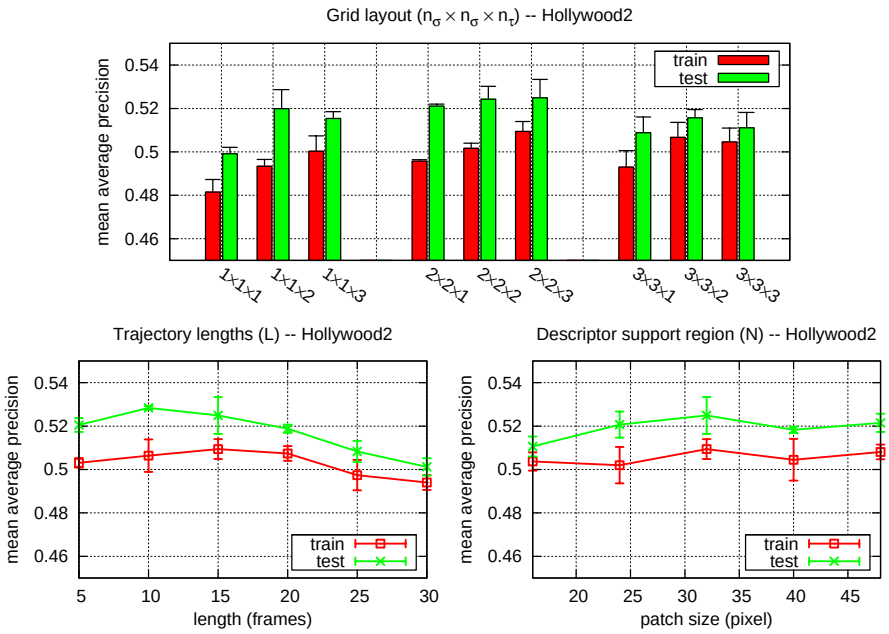


Figure 5.4: Evaluation of the influence of the descriptor parameters on training and test set of the *Hollywood2* dataset: (top) spatio-temporal grid layout $n_\sigma \times n_\sigma \times n_\tau$, (bottom left) trajectory length L, (bottom right) patch size for descriptor computation support region around a trajectory N. We optimize the descriptor parameters based on the training set to $L = 15, N = 32, n_\sigma = 2, n_\tau = 3$.

In our evaluation, we employ the three different datasets: *KTH*, *YouTube*, and *Hollywood2*. *KTH* actions (*cf.* section 2.2.2) has been a popular dataset for action classification over the past years. Since the complexity of the video sequences is rather limited on this dataset, we include *YouTube* (*cf.* section 2.2.4) as well as *Hollywood2* datasets (*cf.* section 2.2.5) which feature more realistic setups. Especially *Hollywood2* consists of rich type of video data with close-up and distant views, camera ego-motion as well as background clutter.

5.3.2 Evaluation of the descriptor parameters

In this section we investigate the performance of our descriptor on the *Hollywood2* dataset with respect to the values of the different parameter. We optimize the parameters on the training set using 10 fold cross-validation (figure 5.4, results in red). For completeness, we also report results on the test set (figure 5.4, results in green).

The descriptor *grid layout* (*cf.* section 5.2.2) controls the spatio-temporal resolution of HOG, HOF, and MBH descriptors. Figure 5.4 (top) shows the performance of our descriptor as a function of the grid layout. It can be observed that the performance improves for a higher number of temporal splits. With respect to the spatial division, a layout of 2×2 cells seems most appropriate. Overall, the layout with $2 \times 2 \times 3$ cells, i.e., $n_\sigma = 2, n_\tau = 3$, yields highest performance on both, train and test set.

The *trajectory length* defines for how many frames a feature point is tracked in a video sequence (*cf.* section 5.2.1). Figure 5.4 (bottom, left) illustrates the performance for different lengths in the range from 5 to 30 frames. According to the training set, optimal performance is achieved with a length of $L = 15$. For longer trajectories, performance drops. This can be explained by the fact that long trajectories tend to drift away from the initial interest point or get lost due to occlusion and rapid motion.

The scale factor N regulates the size of the *support region* surrounding the the trajectory (*cf.* section 5.2.2). This region is encoded by HOG, HOF, and MBH descriptors. According to the results (figure 5.4 (bottom, right)) we can observe that this parameter is not very sensitive to the parameter setting and that results for train and test set behave similarly. An optimum is achieved for $N = 32$, i.e., a supporting size of 32×32 pixels.

These best parameters are used for the remained of our experimental results.

5.3.3 Experimental results

Table 5.1 presents the results for the different descriptors (Trajectory, HOG, HOF, MBH) on three benchmark dataset (results reported as average over at least three separate runs). We give results obtained with each descriptor separately, but also for all possible combinations. We can observe that trajectory information alone does not suffice to give an improvement over our baseline method using Harris3D and HOG-HOF features (table 5.1 (second last row)). The HOG+HOF descriptor combination of our trajectory features compares favorably to the baseline (with the same type of descriptors). With the full descriptor combination, results improve even further. This suggests (i) that a representation based on local feature trajectories is in general beneficial for BoF based action recognition; (ii) that our descriptors for trajectory shape and motion boundaries (MBH) help to improve recognition results even further.

Moreover it can be seen that our proposed MBH descriptor shows excellent results. Using MBH alone achieves state-of-the-art results on *Hollywood2* and gives even slightly better results than the full combination on *KTH* and *YouTube*. This clearly shows the advantage of motion boundaries: static background clutter and camera ego motion are suppressed and only information at boundaries of motion fields is retained in the description. Presumably due to simpler background and less clutter, MBH has the edge on *KTH* and *YouTube* over the full descriptor combination. However, the combination proofs beneficial on *Hollywood2* since video sequences contain more complex motion patterns, camera ego-motion, and strongly cluttered background.

Some partial combinations show sligthly better performance then the full one. On *KTH*, any sub-set of descriptors that includes MBH achieves similar results. For *YouTube*, we can observe that the combination of MBH with the trajectory descriptor outperforms the full descriptor, and on *Hollywood2* this is the case for the concatenation of trajectory+MBH+HOF descriptors. Nevertheless, it is the full combination of all descriptor types that shows overall the best and the most stable results.

The number of features computed with our method in comparison to the baseline, shows that both are comparable. The average number of features per frame on *Hollywood2*

	KTH	*YouTube*	*Hollywood2*
Trajectory	87.8%	64.5%	47.6%
HOG	85.2%	73.9%	40.7%
HOF	92.5%	70.3%	48.1%
MBH	**94.3%**	**80.8%**	50.6%
Trajectory+HOG	86.5%	73.9%	42.4%
Trajectory+HOF	92.5%	71.0%	49.9%
Trajectory+MBH	**94.3%**	**81.3%**	51.4%
HOG+HOF	92.9%	79.2%	51.1%
HOG+MBH	**94.3%**	**80.7%**	47.3%
HOF+MBH	93.4%	76.6%	52.0%
Trajectory+HOG+HOF	93.1%	78.0%	51.1%
Trajectory+HOG+MBH	**94.3%**	**81.0%**	48.4%
Trajectory+HOF+MBH	93.8%	75.4%	**52.9%**
HOG+HOF+MBH	93.9%	**80.5%**	**52.3%**
Full combination	**94.2%**	**79.8%**	**52.5%**
Baseline [Laptev et al., 2008]	92.0%	68.7%	47.3%
State-of-the-art	**94.5%** [Gilbert et al., 2009]	71.2% [Liu et al., 2009]	50.9% [Gilbert et al., 2009]

Table 5.1: Classification results of our method on *KTH*, *YouTube*, and *Hollywood2* datasets. Row 1 to 14 show the performance of all possible descriptor combinations. The sixth row gives the performance with a combination of all descriptors (Trajectory+HOG+HOF+MBH). The last two rows report baseline results with Harris3D + HOG-HOF and the current state-of-the-art. All results are presented as an average over at least three separate runs.

sequences is for our method about 77.2 and for the baseline with Harris3D about 52.4 features per frame.

5.3.4 Comparison to the state-of-the-art

We can observe that our proposed descriptor, i.e., the *combination* of shape, appearance and motion, significantly outperforms the state of the art on *YouTube* and *Hollywood2* and is on par with it for *KTH*. Note that for all experiments we used a common parameter setting that was optimized on the training set of *Hollywood2*.

Results on the *KTH actions* dataset are presented in Table 5.1, first column. We also refer to section 2.2.2 for a more complete listing of current state-of-the-art results. Our combination of trajectory, HOG, HOF, and MBH descriptors (94.2%) significantly outperforms the HOG-HOF descriptor of the baseline by 2.2%. In comparison to the state-of-the-art (*cf.* section 2.2.2), our method is able to be on par with previously reported results. Gilbert *et al.* Gilbert et al. [2009] achieved (94.5%), however, they use higher level knowledge with an hierarchical approach.

Table 5.1, second column, summarizes results on the YouTube *actions* dataset. Our combination of trajectory, HOG, HOF, and MBH descriptors improves results over our baseline

by 11.1% to 79.8%. Our results also improve over the originally reported 71.2% accuracy by the authors of the dataset [Liu et al., 2009]. However, note that we cannot directly compare to their results since Liu et al. carried out experiments on a smaller version of the dataset containing 11 categories with 1168 sequences.

Results for the Hollywood2 *actions* dataset are presented in Table 5.1, last column. *Hollywood2* contains a large amount of camera motion which renders feature tracking more difficult. The combined descriptor gives 52.5% which is an improvement of 5.2% over our baseline. Our combined trajectory descriptor proves to outperform significantly previously reported results in the literature on this most challenging dataset. As for *KTH*, the current state-of-the-art for this dataset (see also section 2.2.5) has been obtained by Gilbert et al. [2009] with 50.9% [2].

5.4 Conclusion

This chapter introduced a novel descriptor based on feature trajectories and evaluated its performance for bag-of-features based action recognition in videos. Our descriptor combines trajectory information with motion and appearance information using histograms of oriented gradients, optical flow, and motion boundary histograms. Experimental results demonstrate its effectiveness on three benchmark datasets. Our method outperforms the current state of the art on *YouTube* and *Hollywood2* datasets and is on par for *KTH*. Furthermore, we introduced a motion boundary descriptor for action recognition. This descriptor can cope with camera ego-motion as well as cluttered background and gives excellent results on all datasets.

2. Unpublished results, personal communication with the authors.

La détection de personnes, peut-elle aider la reconnaissance d'actions?

Dans les chapitres précédents, nous avons étudié des méthodes existantes et de nouvelles méthodes basées sur la représentation par sac-de-mots dans le cadre de vidéos réalistes. Toutefois, une limitation de cette représentation est qu'elle ne tient pas explicitement compte d'objets ou d'acteurs en raison de sa représentation non-ordonnée et basée uniquement sur des caractéristiques locales. Par conséquent, ce manque de connaissances explicites d'objets empêche la modélisation de l'information structurale qui peut améliorer la performance en classification [Dalal and Triggs, 2005, Lazebnik et al., 2006]. En outre, le modèle sac-de-mots est comme représentation globale intrinsèquement sensible au bruit de fond [Zhang et al., 2007]. D'un autre côté, les approches holistiques (section *cf.* 2.1.2) modélisent par définition l'information structurale et elles sont robustes aux variations de fond car elles sont centrées sur l'homme.

Afin de bénéficier des avantages de ces deux approches, nous examinerons dans ce chapitre une méthode qui combine un modèle sac-de-mots avec une approche holistique. Pour ce faire, nous examinons comment et dans quelle mesure la détection et le suivi des acteurs en vidéo peut améliorer la reconnaissance d'actions (*cf.* figure 6.1).

6

Will person detection help bag-of-features action recognition?

Contents

In previous chapters, we have investigated existing as well as new methods based on bag-of-features (BoF) representations for realistic video settings. However, one limitations of BoF is that it has no explicit notion of objects or actors due to its orderless representation. Consequently, this lack of explicit object knowledge prevents modeling of spatial layout information which has been shown to increase performance [Dalal and Triggs, 2005, Lazebnik et al., 2006]. Furthermore, BoF provides a global video representation which is inherently sensitive to background clutter [Zhang et al., 2007]. On the other hand, human-centric (or holistic) approaches (*cf.* section 2.1.2) inherently model spatial layout information and are robust to background variations since they are based on human detections or tracks.

In order to benefit from the strength of both approaches, we explore in this chapter a method that combines a "loose" bag-of-features model with a human centric approach. For this, we investigate how tracking of human actors can address the aforementioned deficiencies of the bag-of-features representation and to which extent it can improve action recognition performance (*cf.* figure 6.1).

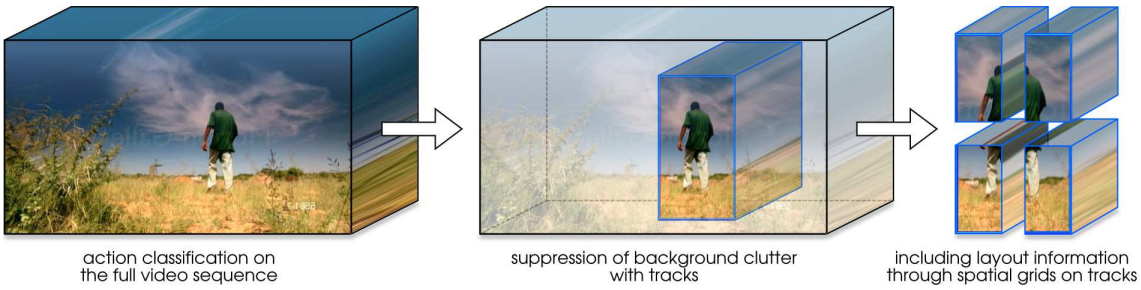


Figure 6.1: This chapter analyzes the importance of human-centered attention for bag-of-features based action recognition. We use human tracks to suppress background (middle) and improve spatial modeling of human actions (right).

6.1 Introduction

The contribution of this chapter is two-fold. First, we treat human tracks as an approximate actor-background segmentation to suppress clutter (*cf.* middle part of figure 6.1). Intuitively, narrowing down the attention to actors should benefit action recognition accuracy. The result is nevertheless worth quantifying, since in natural settings context might play an important role in recognition.

Second, we incorporate human layout information in our action models (*cf.* right part of figure 6.1). For this, we make the hypothesis that narrowing down the attention to the actor will allows us to enforce more spatial constraints in the model, which in turn should result in better accuracy for action recognition. We propose to control the amount of layout information by varying the resolution of spatial grids [Lazebnik et al., 2006] and verify our hypothesis experimentally.

To obtain human tracks for the experiments mentioned above, we use off-the-shelf pedestrian and upper body detectors [Dalal and Triggs, 2005, Ferrari et al., 2008] and combine detection into tracks according to Everingham et al. [2006]. We also use "ground-truth" tracks emulating an "ideal" detector. This allows us to make conclusions regarding desirable system designs which might concern both current and future systems. Furthermore, we run our experiments on three datasets of varying complexity—basic *KTH* (*cf.* section 2.2.2), realistic *UCF* (*cf.* section 2.2.3) and challenging *Hollywood1* (*cf.* section 2.2.5)—in order to investigate how our conclusions might depend on the task.

6.2 Action description

In the following, we give details of our action description and how we combine orderless BoF representations with information on human localization. Subsection 6.2.1 discusses how human tracks are obtained for the three different datasets that we investigate (*KTH*, *UCF*, and *Hollywood1*) and how features are computed. Details on how we gradually incorporate human layout information in the bag-of-features representation are given in section 6.2.2.

6.2.1 Human tracks

Human tracks are constructed from a set of bounding-boxes connected in time. In this work, bounding boxes are obtained either automatically using off-the-shelf pedestrian and upper-body detectors [Dalal and Triggs, 2005, Ferrari et al., 2008] or they are provided as ground-truth. In order to obtain features on the foreground (actors and their closest vicinity), we reuse features from the full videos and keep only those that fall into the bounding box of a human track.

Automatic tracks. For the *KTH* dataset we use the pedestrian detector of Dalal and Triggs [2005] and apply it to all frames. Since only one person is visible per sequence, we obtain tracks by applying a simple outlier removal strategy along with temporal smoothing and interpolation. Results are shown in the top row of figure 6.2.

Since the *UCF* dataset often involves several people in the scene, we run the same pedestrian detector [Dalal and Triggs, 2005] and link detections into tracks using agglomerative clustering as proposed by Everingham et al. [2006]. We exploit temporal consistency to improve detection results by (i) removing short tracks (ii) filling in missing detections within tracks and (iii) applying temporal smoothing of detections. *UCF* sequences contain high variation of the background and highly articulated human poses, which results in a decreased precision and recall of human detection. Example detections are shown in the middle row of figure 6.2.

On *Hollywood1*, humans are in general visible only with their upper body. Therefore, we employ the same detector [Dalal and Triggs, 2005] as for *KTH* and *UCF*, but trained for upper bodies as proposed by Ferrari et al. [2008]. We also use the same temporal association [Everingham et al., 2006] as for *UCF*. Figure 6.2, bottom row, shows several sample frames of our final tracks.

Ground truth tracks. We do not use ground truth tracks for *KTH* since our automatic ones are of a sufficiently good quality. For the *UCF* dataset, tracks of the person performing an action are provided with the dataset (*cf.* figure 6.3, top). For *Hollywood1*, we manually annotate upper body tracks (cf. figure 6.3, bottom). Training is performed using all tracks with humans performing a given action, and for testing, all visible humans are annotated and used, mimicking a perfect human detector.

6.2.2 Spatial bags-of-words

To encode layout information within the BoF representation, we employ spatial grids [Laptev et al., 2008, Lazebnik et al., 2006], see figure 6.4. The video sequence is split into (spatial) subsequences, and a histogram is computed for each subsequence. The final histogram is obtained by concatenating histograms of all cells in the grid. In order to compare to the performance with tracks, we introduce as our baseline method a standard BoF over the whole video using the same grid layouts as for the tracks. For human tracks,

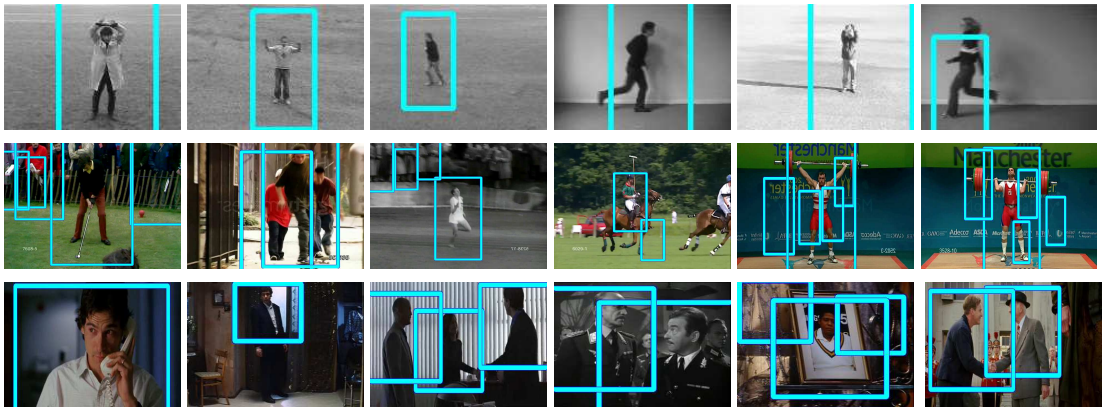


Figure 6.2: Examples of automatic tracks on *KTH* (top) *UCF* (middle) and *Hollywood1* (bottom) action datasets.

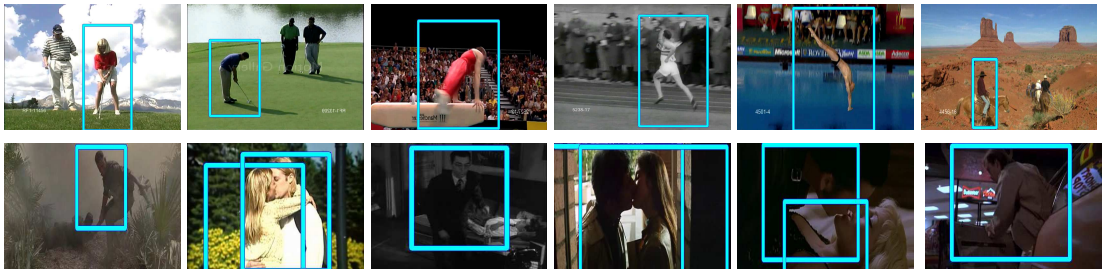


Figure 6.3: Ground-truth tracks for *UCF* (top) and *Hollywood1* (bottom) datasets.

the grid position of a feature is defined relatively to the position of the track's bounding box at the corresponding time instant (*cf.* figure 6.1(right)). In the case of multiple tracks, BoF histograms of all tracks in the sequence are summed up. Features that belong to different overlapping tracks can vote multiple times into the final histogram, i.e., once for each track.

For our experiments, we need to quantify the "amount" of layout information used for action recognition. For this, the first n of the following grid layouts are combined (cf. figure 6.4):

$$\mathcal{L} = \{\mathcal{L}_i\} = \{1{\times}1,\ 2{\times}1,\ 2{\times}2,\ 3{\times}2,\ 3{\times}3,\ 4{\times}3,\ 4{\times}4,\ 5{\times}4,\ 5{\times}5\}. \tag{6.1}$$

The larger n, the more layout information is incorporated into the action model. Note that we slightly prefer vertical divisions to horizontal ones. This is motivated by the fact that naturally vertical variations are smaller than horizontal variations (i.e., a person in an image is rather upright than upside-down). For classification, we combine the different grid layouts with a non-linear SVM with multi-channel kernel, as detailed in section A.1.

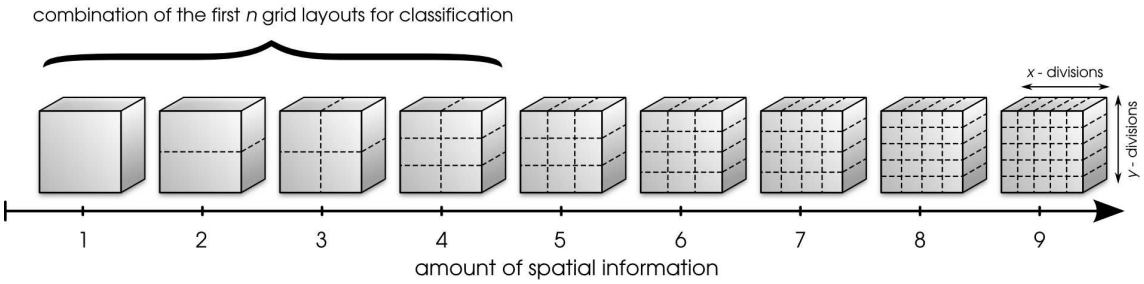


Figure 6.4: Human layout information is encoded through spatial grids. We use a sequence of grids of increasing density to control the amount of spatial information (spatial constraints) included. We combine the first n grid layouts, in this example $n = 4$.

6.3 Experimental results

Our goal is to quantify the improvement in human action recognition when extending the BoF representation with knowledge about actor localization. In the following subsections, we compare the performance of our baseline BoF system (cf. section 6.2) with the same system, but with background features removed based on human tracks. We show a separate figure for each dataset. The recognition accuracy is given as a function of the amounts of spatial constraints. We compare results for the BoF baseline (red squares) and each of the track types used to select features (blue triangles for tracks automatically obtained from person detections; green circles for ground-truth tracks)—see figures 6.5-6.7. For each of the datasets, we draw two types of observations. First, we evaluate the gain due to background suppression by comparing the performance of the orderless representation (only one "grid" level, leftmost measurement on each plot, highlighted). Second, we assess the gain due to stronger layout (indicated by the tangent of each plot).

In the following, we give implementation details in section 6.3.1 and discuss then results on the datasets employed for our experiments one by one: *KTH* actions (section 6.3.2, *UCF* sports (section 6.3.3 and *Hollywood1* datasets (section 6.3.4).

6.3.1 Implementation details

For our experiments, we employ as local feature descriptor the spatio-temporal HOG3D descriptor (see chapter 3) with the parameter settings as given in [Kläser et al., 2008]. Since HOG3D quantizes 3D gradient orientations, it enables us to account for appearance and motion information at the same time.Feature positions are sampled within a video sequence in a dense manner following our earlier setup described in section 4.2.1. We employ dense sampling with a spatial stride of 12×12 (for *UCF* and *Hollywood1*) as well as 6×6 pixels (for *KTH* due to its smaller resolution) and a temporal stride of 3 frames throughout all our experiments. This allows for a sufficient coverage on tracks for experiments using human position information (section 6.2.1). Other parameters correspond to section 4.2.1. For vocabulary construction, we fix its size to 4000 and use random sampling (*cf.* section A.1). All experiments are repeated three times, each time with a new

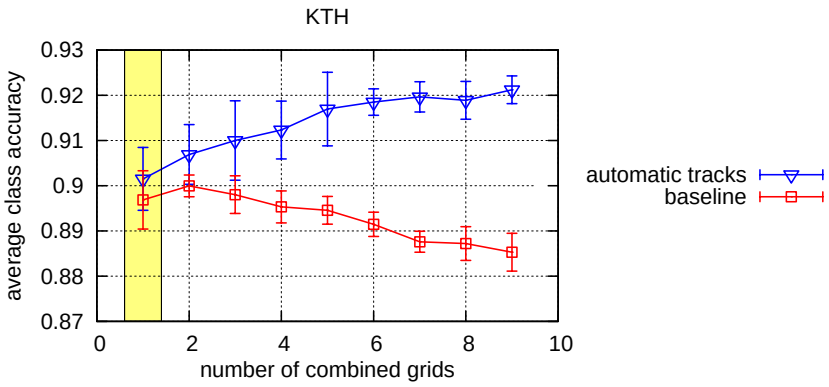


Figure 6.5: Performance plots for the *KTH* actions dataset. Bars indicate standard deviation from the mean.

randomly created codebook. This allows us to estimate mean and standard deviation in the experiments.

6.3.2 KTH actions

Results for the *KTH* dataset are plotted in figure 6.5. Comparing the values for orderless BoF (highlighted measurements in the leftmost column of the plot) allows to estimate the gain in recognition accuracy due to background suppression. For the *KTH* dataset the reduction of background clutter using automatically detected human tracks leads to a small accuracy gain of about 0.5%.

A more significant improvement of over 2% is possible by increasing the number of grids and encoding more layout information. Note, however, that this only holds for the features obtained using tracks, not for the full video where results degrade; the difference between the tracks and the baseline reaches almost 4% for the full combination. This demonstrates that layout information can help to learn a better action model if tracks are used.

The confusion matrix in table 6.1 shows that the main source of confusion is an inherent overlap between jogging and running. Looking at examples of these classes, we have observed that there is no visual difference between some sequences of the two classes.

We refer the reader to section 2.2.2 for a detailed overview of the current state-of-the-art for the *KTH* actions dataset. The currently best result on this dataset has been reported for the hierarchical data mining approach by Gilbert et al. [2009] which achieved 94.5%. Han et al. [2009] obtained 94.1% accuracy with a multi-kernel classifier. Among the results that have been reported with a pure BoF representation, the combination of Harris3D interest points together with HOF (92.1%) as well as HOG-HOF (91.8%) gave highest results [Wang et al., 2009] in the literature.

Our average accuracy over three runs (for our full method, i.e., using automatic detections to suppress background and combining all 9 grid layouts) is 92.1%. In general, our results are situated among the state-of-the-art results. However, our method is not optimized for high performance, yet rather for a fair comparison with the baseline. We showed that

True class \ Predicted class	boxing	clapping	waving	walking	jogging	running
boxing	98.4	0.0	0.0	1.6	0.0	0.0
clapping	3.9	96.1	0.0	0.0	0.0	0.0
waving	0.2	4.4	95.4	0.0	0.0	0.0
walking	0.0	0.0	0.0	96.5	3.5	0.0
jogging	0.0	0.0	0.0	3.0	93.3	3.7
running	0.0	0.0	0.0	0.0	18.5	81.5

Table 6.1: Confusion matrix for the *KTH* dataset. Classification was performed using our full system, i.e., features from detected actors and combining all 9 grid layouts. Note the confusion between running and jogging.

performance on *KTH* can be improved significantly using layout information on the tracks. Therefore our approach shows the potential to improve the performance of other methods, as well.

6.3.3 UCF Sports

Experimental results for the *UCF* dataset are presented in figure 6.6. If we compare the results for orderless BoF (highlighted measurements on the left of the plot), we clearly see a gain due to suppressing background features and narrowing down attention. The recognition accuracy improves significantly by 4% with "ideal" tracks provided as ground-truth. The off-the-shelf pedestrian detector is also able to out-perform the baseline by over 2%.

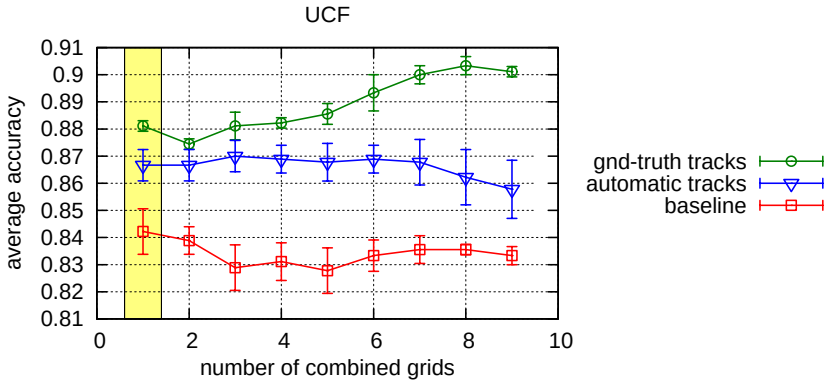


Figure 6.6: Performance plots for the *UCF* sport actions dataset. Bars indicate standard deviation from the mean.

Further interesting conclusions can be drawn from the evaluation of layout information. Enforcing stronger layout models can degrade the performance of the baseline and also of

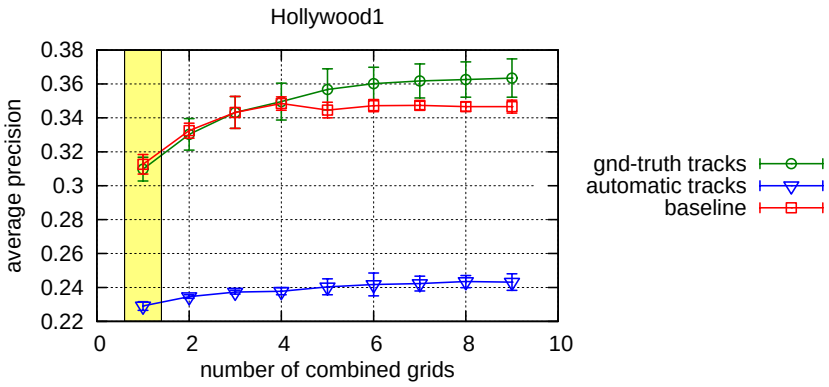


Figure 6.7: Performance plots for the *Hollywood1* actions dataset. Performance for ground-truth tracks is a learned combination of ground-truth tracks and the BoF baseline. Bars indicate standard deviation from the mean.

automatic tracks. For the baseline, the degradation of its results is permanent, while for the automatic case we can observe only a minor improvement up to three grid combinations. An ideal detector and tracker, however, allows to significantly and consistently improve the recognition accuracy when more layout information is included. This shows the importance of a good human tracker in order to fully exploit the knowledge about actor localization.

It is also interesting to look at the confusion matrices for this dataset. Table 6.2 compares the matrices obtained for the baseline with an orderless bag model (left) and by using the ground truth actor annotations and enforcing a stronger layout model (right). In the first case, note the general confusion for actions such as riding and weight lifting with other classes. This confusion is significantly reduced in the second case for most classes. Nevertheless, some confusion remains using tracks—the accuracy for running even dropped. This is presumably due to the reduced amount of context information, such as strong camera ego-motion during running. Other actions that remain confused are skateboarding and walking. This is explainable given their visual similarity.

Works that published results on the *UCF* sports dataset are Rodriguez et al. [2008] who also published the dataset and Wang et al. [2009] (*cf.* section 2.2.3). Rodriguez et al. reported an accuracy of 69.2% with a template matching approach, and Wang et al. obtained 85.6% in a BoF setup close to ours. In an "ideal" setup (i.e., with ground truth tracks), our system achieves 90.1% average accuracy (combining all 9 grid layouts) which is significantly higher than the current state-of-the-art. For the automatic case with human detections, we obtain with our features 86.7% by only considering foreground.

6.3.4 Hollywood actions

Experimental results for the *Hollywood1* dataset, the most challenging dataset in our setup are given in figure 6.8. Since the classification task for this dataset consists of multiple binary tasks, we show results for each class individually. One immediately notices that (unlike for the previous datasets) the results degrade significantly when using automatic

Predicted class

True class	dive	golf	walk	kick	run	lift	ride	skateboard	highbar	swing
dive	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
golf	0.0	79.6	10.2	10.2	0.0	0.0	0.0	0.0	0.0	0.0
walk	1.6	6.3	82.8	0.0	0.0	1.6	1.6	6.3	0.0	0.0
kick	1.6	8.1	0.0	83.9	1.6	1.6	0.0	3.2	0.0	0.0
run	0.0	8.0	0.0	12.0	76.0	0.0	4.0	0.0	0.0	0.0
lift	6.7	0.0	6.7	6.7	0.0	71.7	8.3	0.0	0.0	0.0
ride	2.6	10.5	6.6	10.5	5.3	5.3	59.2	0.0	0.0	0.0
skateboard	0.0	0.0	11.1	5.6	0.0	0.0	0.0	83.3	0.0	0.0
highbar	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0
swing	0.0	0.0	1.7	3.3	0.0	0.0	0.0	0.0	0.0	95.0

Predicted class

True class	dive	golf	walk	kick	run	lift	ride	skateboard	highbar	swing
dive	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
golf	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
walk	0.0	2.3	88.6	0.0	0.0	0.0	0.0	9.1	0.0	0.0
kick	0.0	0.0	0.0	95.0	0.0	0.0	0.0	0.0	0.0	5.0
run	7.7	0.0	0.0	23.1	51.3	0.0	17.9	0.0	0.0	0.0
lift	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0
ride	0.0	0.0	8.3	0.0	0.0	0.0	91.7	0.0	0.0	0.0
skateboard	0.0	0.0	23.6	0.0	0.0	0.0	0.0	76.4	0.0	0.0
highbar	0.0	1.3	0.0	0.0	0.0	0.0	0.0	0.0	98.7	0.0
swing	0.0	0.0	0.0	5.0	0.0	0.0	0.0	0.0	0.0	95.0

Table 6.2: Confusion matrices for (top) the *UCF* sports dataset using orderless features on the full video and (bottom) using (ground truth) actor annotation and spatial grids (all combinations). Note how the stronger layout model pruned the worst confusions.

tracks. This is largely due to dynamic camera, clutter and occlusion, which make human detection in *Hollywood1* videos difficult. For instance, people getting out of car are typically not visible at the beginning of the action and are often occluded by the door of the car throughout the action. Additional occlusion and non-upright poses render the detection of people difficult, as well, cf. figure 6.3. Furthermore, even a perfect detector is not guaranteed to improve recognition accuracy. This is most likely due to the fact that this dataset better reflects natural conditions where context can play an important role for action recognition, e.g., for actions such as getting out of a car or kissing. *Hollywood1* actions include interactions between different humans and interactions with objects that might also be harder to interpret without context information [Marszałek et al., 2009]. Overall, a significant gain can be observed for the classes HugPerson, StandUp and SitUp. For the classes AnswerPhone and SitDown we can note a slight improvement. However, the performace decreases for Kiss and GetOutCar, most likely due to the context information playing an important role for these action classes.

Since track information is not useful for all types of actions, we combine both representations—baseline and track-based. We employ a simple selector choosing the best representation for a particular action in an automatic manner. During training, the representation that performs best on the training set (evaluated via cross-validation) is selected. Figure 6.7 shows the average AP gain in such setup. The result is consistent with those for other datasets: the improvement due to background suppression is relatively small, while enforcing stronger layout information is beneficial.

For the *Hollywood1* dataset, our baseline (a single orderless channel) obtains 31.3% mean AP and outperforms the corresponding orderless HoG (27.0%) and HoF (21.5%) channels of Laptev et al. [2008]. It is also close to the performance of their best channel (32.2%). With an "ideal" detector in combination with the BoF on the full video, we improve up to 36.4% with a single feature type. Laptev et al. proposed a method to learn combinations of different features which they showed to lead to a higher average precision of up to 38.4% on this dataset. However, combining different feature types is beyond the scope of this work.

Section 2.2.5 gives an extensive list of recent state-of-the-art results. Similar to *KTH*, Gilbert et al. [2009] (53.5%) and Han et al. [2009] (47.5%) obtain overall highest results. Note that Han et al. yielded as performance of their best channel alone 33.3% which is comparable to our results. Compared to existing, standard BoF approaches, best results have been reported by Willems et al. [2009] (29.6%) by using a Hessian feature detector along with a variant of HOG3D.

Our results compare favorably to the state-of-the-art with only single feature types. As stated before, employing human localization offers cues for action recognition that are complementary to existing approaches, e.g., feature combination [Marszałek et al., 2009, Han et al., 2009]. In a combined setup, it can therefore further improve existing state-of-the-art methods.

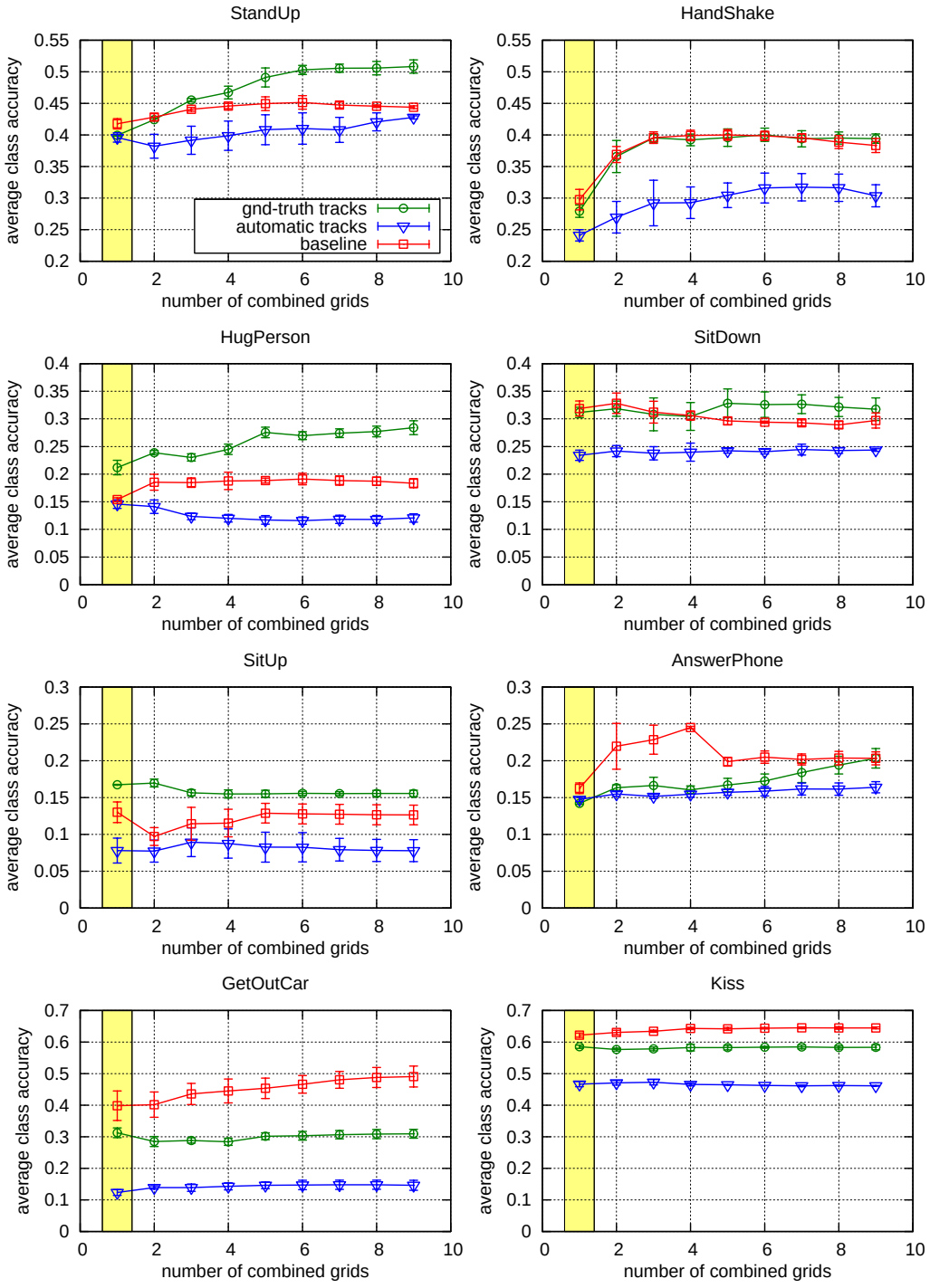


Figure 6.8: Per class results on *Hollywood1*. Note that a performance improvement using human tracks is dependent on the action class. A significant gain can be observed for the classes HugPerson, StandUp and SitUp. The performance decreases for Kiss and GetOutCar, most likely due to the context information playing an important role for these action classes.

6.4 Summary

In this chapter, we have shown that action recognition can benefit from human localizations in videos. Quite surprisingly, it turns out that this gain is not due to suppressing background clutter. Only in the case of simple scenarios, background suppression helps to improve classification results. However, for realistic settings, removing background can lead to removal of valuable context. Therefore background suppression resulted in general in only minor recognition accuracy improvement. In the case of a few action classes (getting out of a car, kissing) we observed even a performance degradation.

Furthermore, we have proposed to use human tracks to improve action modeling. We have redefined a popular spatial pyramid concept as a model with controlled levels of spatial constraints. We have shown that narrowing down the attention to human actors allows to incorporate more layout information into the learned model. In general, this positively benefited recognition accuracy. However, on realistic videos and for some action classes, we observed no or only minor improvement.

Localisation d'actions humaines dans des vidéos

Alors que les chapitres précédents ont abordé le problème de la *classification* de séquences d'action, ce chapitre se concentre sur la *localisation* d'actions dans l'espace (par une région 2D dans l'image) et le temps (par une plage temporelle). Comme données, nous utilisons des films réalistes avec des environnements dynamiques et surchargés, avec de l'occlusion partielle, du mouvement de caméra et du fond bruité. Comme le montrent les résultats du PASCAL Visual Object Challenge [Everingham et al., 2009b], la localisation est un problème plus exigeant que la classification.

Pour accomplir cette tâche, nous proposons une approche qui divise explicitement la localisation d'action en deux étapes. Dans un premier temps, les personnes dans une séquence vidéo sont détectées et suivies, ce qui détermine la localisation spatiale de l'action. Compte tenu de ces détections, nous déterminons dans un deuxième temps *si* l'action se déroule et *quand* (localisation temporelle) en appliquant un classificateur en fenêtre coulissante à un nouveau descripteur spatio-temporel adapté aux détections humaines.

7

Human focused action localization in video

Contents

While previous chapters have addressed the problem of *classifying* action sequences, this chapter concentrates on *localizing* human actions both in space (the 2D image region) and time (the temporal range). As type of data, we employ real-world movies with crowded, dynamic environment, partial occlusion and cluttered background. As is well known from the results of the PASCAL Visual Object Classes challenges [Everingham et al., 2009b], localization is much more demanding than classification.

To accomplish this task, we propose an approach which explicitly splits the action localization into two stages. In the first stage, humans are detected and tracked; this determines the spatial localization of the action. Given the track, we determine in a second stage *if* the action occurs and *when* (temporal localization) by using a sliding window classifier based on a novel spatio-temporal track-adapted 3D-HOG descriptor.

7.1 Introduction

While the idea of combining tracking and classification for action localization is not new (see also section 2.1.2), previously it has mainly been applied to video restricted to a static camera [Hu et al., 2009, Yuan et al., 2009] or simple background with limited clutter, as for example soccer or ice hockey fields [Efros et al., 2003, Lu and Little, 2006]. In such a context, techniques such as background subtraction, image differencing, or color segmentation (on the soccer field) can be employed to localize the actors. However, in movie-style video sequences, no such specific techniques can easily be employed to guide human detection.

A few recent approaches address the problem of localizing natural actions in realistic, cluttered videos (*cf.* section 2.1.2): Laptev and Perez [2007] use an action-pose specific human detector (e.g. for the moment of drinking) in combination with a spatio-temporal video block classifier; Willems et al. [2009] employed a voting approach based on discriminative visual words; Ke et al. [2007b] match spatio-temporal voxels to manually created shape templates. Unlike these works, our approach uses a generic human detector and tracker followed by a task-specific action detector. As will be demonstrated in the experiments, this choice is crucial for both efficiency and recognition accuracy. First, tracks help to narrow down the focus and thus to simplify the recognition task. And second, as opposed to a cuboidal action descriptor, tracks enable a more principled description of actions that is able to follow the actors motion and capture even more articulated actions. As will be shown in the comparison, our method substantially outperforms current state-of-the-art results reported by Laptev and Perez [2007], Willems et al. [2009].

7.2 Datasets and evaluation method

For our experiments, we use two movie datasets that differ from those used in previous chapters: *Coffee&Cigarettes* (*C&C*) on which we additionally evaluate the smoking action and our new *Hollywood–Localization* dataset.

Coffee&Cigarettes. The film *C&C* consists of 11 short stories, each with different scenes and actors. The dataset *C&C* introduced by Laptev and Perez [2007] consists of 41 drinking sequences from six short stories for training and 38 sequences from two other short stories for testing. Additionally, the training set contains 32 drinking samples from the movie *Sea of Love* and 33 drinking samples recorded in a lab. This results in a total of 106 drinking samples for training and 38 for testing. The total time of the testing sequences is about 24 minutes.

We evaluate additionally on smoking actions. Laptev and Perez [2007] provide with their dataset also annotations for smoking, however they did not report results for localization. The smoking training set contains 78 samples: 70 training samples are obtained from six short stories of *C&C* (the ones used for training the *drinking* action) and 8 from *Sea of*

Love. 42 samples from three other short stories of *C&C* are used for testing which amounts to about 21 minutes of video data.

We use the evaluation protocol of Laptev and Perez [2007] in our experiments: an action is correctly detected if the predicted spatio-temporal detection has an overlap with the ground truth annotation $O(X,Y) \geq 0.2$. The overlap between a ground truth cuboid Y and a track segment X is given by $O(X,Y) = (X \cap Y)/(X \cup Y)$. Once an annotated sample has been detected, any further detection is counted as a false positive.

Hollywood–Localization. To evaluate the performance of our approach on challenging video data, we introduce the *Hollywood–Localization* dataset based on sequences from Hollywood movies [Marszałek et al., 2009]. In total we annotated 130 clips containing the action *answer phone* and 278 clips with the action *standing-up*. The same number of randomly selected clips not containing the action are used as negatives in each case. We keep the training/test movies split from Marszałek et al. [2009] which roughly divides the samples into two halves. In total, the amount of testing data for answer phone and standing-up is about 17.5 and 39 minutes.

Since *Hollywood–Localization* actions are much more dynamic, a cuboid is no longer an adequate representation for the ground truth. Therefore, the ground truth we provide specifies an action by its temporal start and end frames, and a spatial localization rectangle for one of the intermediate frames. For evaluation we adapt the *C&C* protocol. The overlap in time is computed as $O_t(X,Y) = O(X_t,Y_t)$, and in space as $O_s(X,Y) = O(X_s,Y_s)$, where X_t and Y_t are the temporal extents of the track X and the annotation Y, and X_s and Y_s are the corresponding spatial rectangles in the annotated action frame. The final overlap is computed as $O'(X,Y) = O_t(X,Y) \times O_s(X,Y)$ and the accuracy threshold is set to 0.2 as for *C&C*.

7.3 Human detection and tracking

To detect (i.e. localize) and track human actors we use the tracking-by-detection approach [Cour et al., 2008, Everingham et al., 2006, Ferrari et al., 2008, Leibe et al., 2007] that has proved successful in uncontrolled video. This involves detecting humans in every frame, and then linking the detections using a simple general purpose tracker. We use this method in combination with human upper body detections based on the HOG descriptor [Dalal and Triggs, 2005] and a sliding window linear SVM classifier (section 7.3.1). Following Everingham et al. [2006], we use KLT [Shi and Tomasi, 1994] as the tracker. We extend the existing tracking approach with a new *interpolation* of missed detections (section 7.3.2) and a additional *classification* stage (section 7.3.3) for the final tracks in order to reduce false positives.

7.3.1 Upper body detection and association by tracking

Since humans in movies are recorded often in close-up or medium view, upper body detectors [Ferrari et al., 2008, Laptev and Perez, 2007] are suitable for movie. Based on

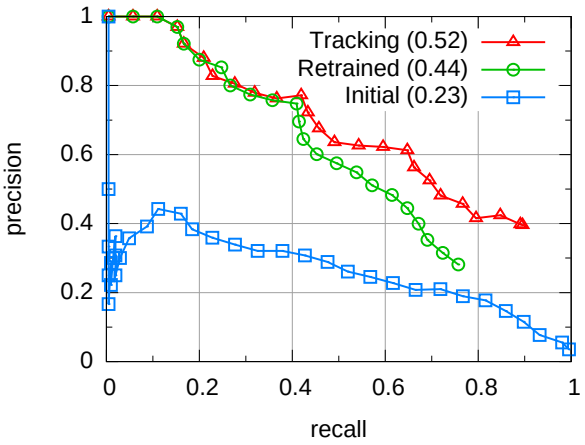


Figure 7.1: Upper body detector evaluated on *frames* from the *C&C* sequences not used for training. Average precision is given in parentheses. Note how precision is improved with detector retraining, and both precision and recall with tracking.

the method by Dalal and Triggs [2005], we train an upper body detector in two stages. In the **initial stage**, positive and negative windows are extracted from the *Hollywood–Localization* training movies. For this purpose we have annotated heads in keyframes and automatically extended them to upper bodies. Each annotation window is jittered [Laptev, 2006] and flipped horizontally amounting to over 30k positive training samples in total. We sample about 55k negative training windows that do not overlap significantly with the positive annotations. For the second **retraining stage**, we follow the strategy of Dalal and Triggs [2005] and look for high ranked false positives using the initial stage detector. We retrieve additional 150k false positives from the *Hollywood–Localization* training movies, and also add over 6k jittered positives and 9k negatives from the *C&C* training set.

Figure 7.1 compares the precision-recall plots obtained for the two stages of the detector and for the final tracker. We evaluate the detectors based on a total of 260 upper bodies that we annotate in 137 frames taken from the *C&C* drinking and smoking test sets [Laptev and Perez, 2007]. A person is considered to be correctly localized when the predicted and ground truth bounding box overlap (intersection to union) ratio is above 0.5. Re-training improves the precision for low recalls but with some loss of recall (blue initial and green retrained lines). However, the recall is largely recovered by the interpolating tracker (red line) which fills in missing detections (as described in section 7.3.2).

Upper body detections are associated between frames using a KLT [Shi and Tomasi, 1994] feature tracker. In a similar manner to Everingham et al. [2009a], the number of KLT features passing through two detections (both forwards and backwards in time) is used to compute a connectivity score between them, and detections are then linked by agglomerative clustering.

7.3.2 Interpolation and smoothing

Detections can be missing in some frames, and hence the tracks formed by agglomerative clustering can have temporal gaps. To construct *continuous* tracks, it is necessary to fill in these gaps (otherwise the subsequent computation of the action descriptor is more difficult). Furthermore, the position and scale of the upper body detections can be noisy. In order to provide a stable reference frame for the subsequent action classification, we smooth (and complete by interpolation) the estimated detection window by optimizing over the track parameters $\{\mathbf{p}_t\}$:

$$\min_{\{\mathbf{p}_t\}} \sum_{t \in T} \left(||\mathbf{p}_t - \bar{\mathbf{p}}_t||^2 + \lambda^2 ||\mathbf{p}_t - \mathbf{p}_{t+1}||^2 \right) \tag{7.1}$$

where $\mathbf{p}_t = (x_t, y_t, w_t, h_t)$ denotes the position, width and height of a bounding box at time instance t for a track T, $\bar{\mathbf{p}}_t = (\bar{x}_t, \bar{y}_t, \bar{w}_t, \bar{h}_t)$ are the detections and λ is a temporal smoothing parameter. Note that if a detection is missed, then the appropriate term $\bar{\mathbf{p}}_t$ is removed from the cost function for that frame. Optimizing (7.1) results in a linear equation with a tri-diagonal matrix, which can be solved efficiently by Gaussian elimination with partial pivoting. Setting $\lambda = 4$ for 25Hz videos results in a virtual "steadi-cam" with no adverse oversmoothing.

Figure 7.1 shows the gain from smoothing and completing detections to form tracks. Exploiting the temporal consistency (tracking) significantly improves the recall of the retrained human detector.

7.3.3 Classification post-processing

Since the upper body detector considers only a single frame, background clutter can generate many false positives. Some of these are quite stable and survive tracking to produce erroneous human tracks that should be removed.

We take a principled approach and in a final stage train a classifier to distinguish correct from false tracks. To this end, we define 12 track measures based on track length (since false tracks are often short); upper body SVM detection score (false detections normally have a lower score than true ones); scale and position variability (those often reveal artificial detections); and occlusion by other tracks (patterns in the background often generate a number of overlapping detections). For these measures we compute a number of statistics (min, max, average) where applicable and form a 12-dimensional feature vector used to classify the track. We obtain ground-truth for the tracks using 1102 annotated keyframes from *Hollywood–Localization* training movies (a track is considered positive if it coincides with an actor in the annotated keyframe, and negative otherwise) and train an SVM classifier (linear and RBF). The SVM is then used to classify the tracks.

Table 7.1 compares different methods used to remove erroneous tracks resulting from background clutter. The detection score turns out to be crucial for recognizing true human tracks. Nevertheless, training an SVM classifier on all 12 track measures significantly

	recall					
	0.99	0.95	0.90	0.85	0.80	0.70
RBF-SVM	0.18	**0.42**	**0.60**	**0.68**	**0.73**	**0.78**
Lin-SVM	**0.19**	0.41	0.58	**0.68**	**0.73**	**0.78**
AvgScore	0.18	0.21	0.27	0.35	0.38	0.50
Occlusion	0.14	0.19	0.23	0.24	0.24	0.25
Length	0.14	0.15	0.18	0.22	0.26	0.27
No filtering	0.14	0.14	0.14	0.14	0.14	0.14

Table 7.1: Precision of *tracks* for various filtering methods at recall rates of interest on *C&C* stories not used for training. Note the huge improvement obtained by classifying on a set of track properties, rather than using the properties individually.

improves recognition precision compared to any heuristics on the individual measures. Using either a linear or a non-linear SVM, the precision at a useful recall of 0.8 improves from 0.14 to 0.73, i.e., the number of false positives is reduced by more than two thirds. The benefits to both precision and recall are evident in figure 7.1.

Overall, the proposed human detection and tracking method copes with a rich set of articulations, viewing angles and scales, as illustrated in figure 7.2, and results significantly improve over the individual human detections. Missed actors arise from unusual shots with camera roll, face close-ups or distant views. In crowded scenes, background actors might be missed, but most of the foreground characters are detected.

7.4 Action localization

Given a set of human tracks, the goal is to determine which tracks contain a given action and to localize the action within the track. Our approach is based on a temporal sliding window, that is, we search for a range of frames which contains the action. Due to the tracks, the spatial extent of the action is already fixed. Consequently, we only need to delimit the beginning and length of an action (a two dimensional search space). This is in contrast with an exhaustive search, which needs to determine also the 2D image region corresponding to the human, i.e., its position and scale in the case of a sliding window approach.

Actions are represented by a spatio-temporal window descriptor. Our descriptor extends the HOG image descriptor [Dalal and Triggs, 2005] to spatio-temporal volumes, and goes beyond a rigid spatio-temporal cuboid [Laptev and Perez, 2007, Willems et al., 2009], as it adjusts piecewise to the spatial extent of the tracks. This introduces a more flexible representation, where the description will remain centred on the deforming human action. This descriptor is termed *HOG-Track*, and is described in section 7.4.1. For temporal localization we use a state-of-the-art two stage sliding window classifier [Harzallah et al., 2009, Vedaldi et al., 2009] on the tracks.

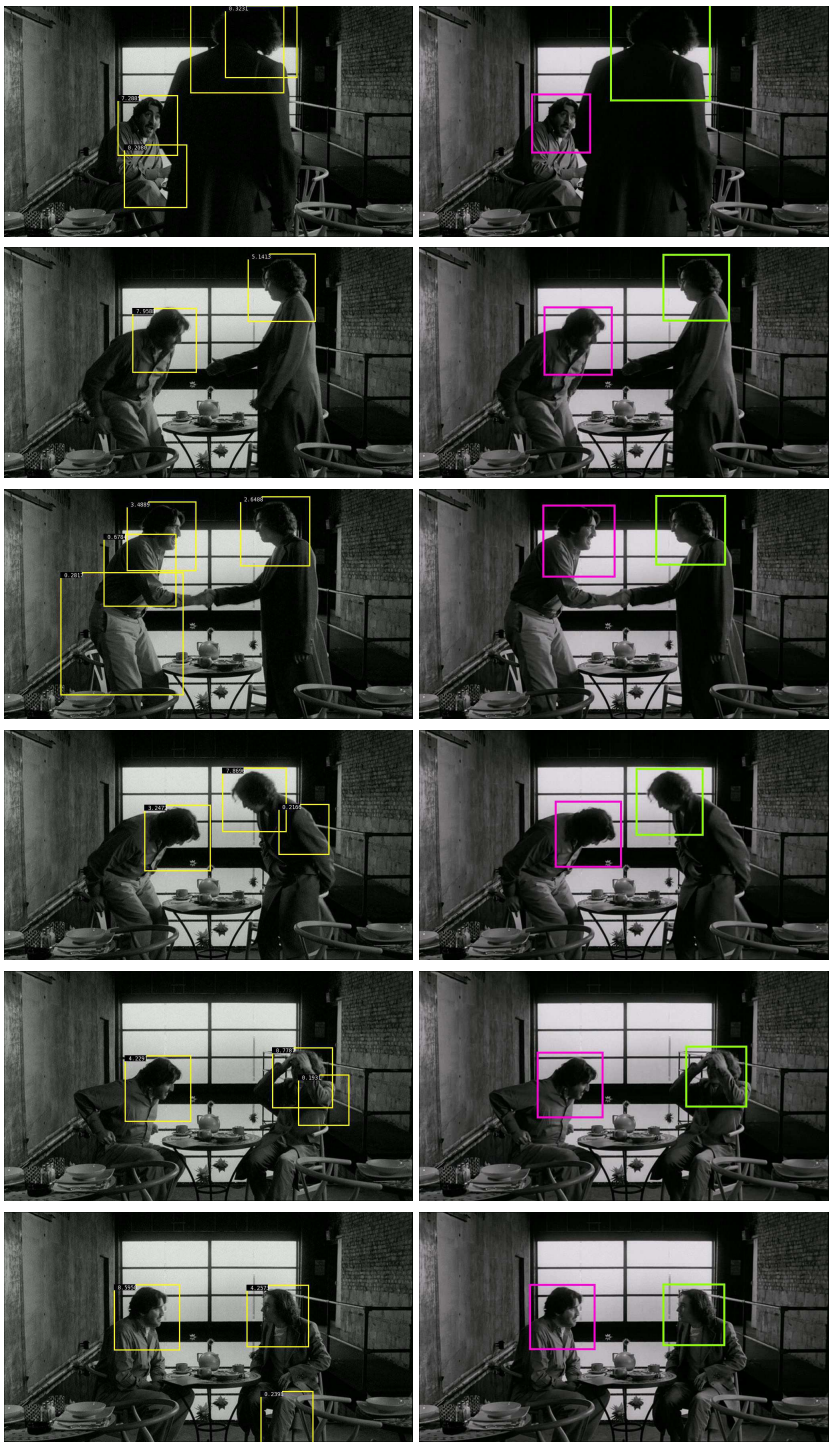


Figure 7.2: Upper body detections (left column) and tracks (right column) after classification post-processing for a sample test sequence of *C&C*. The bounding box colours indicate different tracks. Note the improvement due to the tracking where false positives have been removed, as well as the high accuracy despite motion, articulations and self-occlusion.

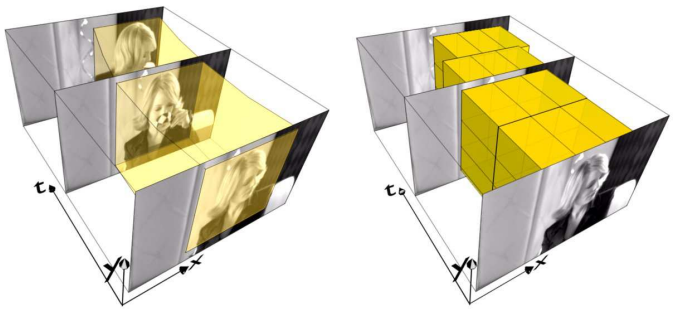


Figure 7.3: The HOG-Track descriptor: (left) the human tracker detects and tracks a human upper body; (right) the *HOG-Track* descriptor divides the track into temporal slices. Each slice is aligned with the bounding box of its centre frame and is divided into a spatial grid of cuboid cells.

7.4.1 HOG-Track descriptor

The *HOG-Track* action descriptor divides a track segment into cells. As in the original HOG [Dalal and Triggs, 2005], there are cells in the 2D spatial domain, but additionally the track segment is divided into temporal slices. These slices are aligned with a human track, as illustrated in figure 7.3. In more detail, a given track segment is defined by a temporal sequence of bounding boxes. This sequence is divided into equally long temporal slices and the spatial image region corresponding to the slice is given by the bounding box of its centre frame. This ensures that our descriptor follows the variation of spatial position of a human within the spatio-temporal volume of the video.

Each slice is split into a spatial grid of cuboid cells as illustrated in figure 7.3 and each cell is represented by a histogram of spatio-temporal (3D) gradient orientations, following our method presented in chapter 3. Orientation is quantized over an icosahedron—a regular polyhedron with 20 faces. Opposing directions (faces of the icosahedron) are identified into one bin, i.e., there are a total of 10 orientations. Each gradient votes with its magnitude into the neighbouring bins, where weights are distributed based on interpolation.

For better invariance to position, we design spatially adjacent cells to have an overlap of 50%. All cell descriptors in a slice are L2 normalized per slice, and the final descriptor concatenates all cell descriptors. The parameters of the descriptor (the spatial grid and temporal slice granularity) are determined by cross-validation, as described in section 7.5. On the drinking and smoking actions the training performance is optimized for a spatial grid of 5×5 and 5 temporal slices. The dimensionality of the resulting descriptor is 10 orientation bins $\times$ 5^2 spatial cells $\times$ 5 temporal slices $=$ 1250. This configuration is used in all our experiments.

7.4.2 Action classification and localization

Our temporal sliding window approach extracts descriptors at varying locations and scales. To classify these descriptors, we use a state-of-the-art two stage approach [Harzallah et al.,

2009, Vedaldi et al., 2009] which rejects most negative samples with a linear SVM, and then uses a non-linear SVM with an RBF kernel to better score the remaining samples.

When training the sliding window classifier, the ground-truth annotations are matched to the tracks and the action part of the track is used for training. The *HOG-Track* is computed for this temporal section, i.e., the temporal slices are aligned with the ground-truth begin and end time stamps of the action. The spatial regions are obtained from the track bounding box of the centre frame of each slice. Training is very similar to the detector training of section 7.3: additional positives are generated here by jittering the original positives in time, duration, and spatial scale. Initial negative samples are obtained by randomly sampling positions with varying lengths in the tracks, which do not overlap with any positive annotations, and in a re-training stage hard negatives are added to the training set. The C parameter and weight for positive samples are determined on the training set using a leave-one-video-out cross-validation. The second stage classifier uses a *non-linear* SVM with an RBF kernel and is trained on the same training data as the linear one. Again, we optimize the parameters via cross-validation.

At test time, a sliding window is used to localize actions. Seven temporal window scales are evaluated starting from a minimum length of $l = 30$ frames, and increasing by a factor of $\sqrt{2}$. The window step size is chosen as one fifth of the current scale. The *HOG-Track* descriptor for each window is classified with the linear SVM. Non-maxima suppression then recursively finds the global maximum in a track and removes all neighbouring positive responses with an overlap greater than 0.3. The remaining detections are re-evaluated with the non-linear SVM classifier. As will be seen next, this second re-scoring stage improves classification results considerably.

7.5 Experimental results

7.5.1 Coffee&Cigarettes

Tracks for action localization. Our action localization method depends on correct track positions in space and time. When training the sliding window classifier, the ground-truth is matched to the tracks and the corresponding tracks are used for training. We only keep samples that have an overlap of at least 0.5. This results in a loss of around 10% of the training samples. During testing an action can not be detected if the track is not localized. This reduces the maximum possible recall by again around 10%.

Descriptor evaluation. In order to determine a suitable layout of our *HOG-Track* descriptor, we evaluate its parameters using cross-validation on the training set. Best results are obtained for 5 or 7 temporal slices; we use 5 as it results in a lower dimensional descriptor. The performance is quite sensitive to the number of spatial cells, best results are obtained for 5×5. This behaviour translates also to the test set which is illustrated in figure 7.4. The performance is averaged over three independent runs.

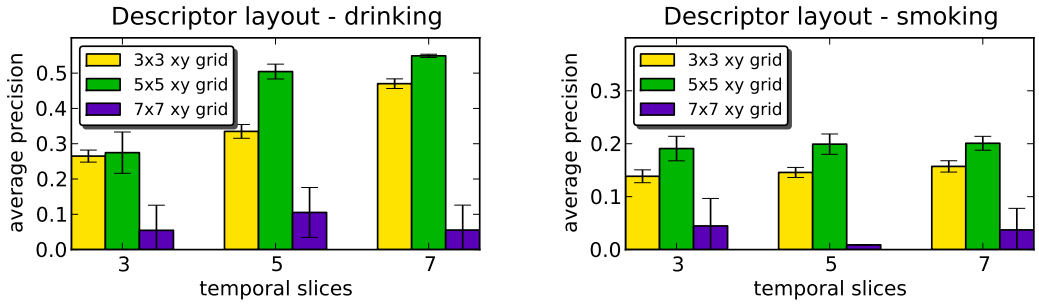


Figure 7.4: HOG-Track descriptor evaluation: for a varying number of spatial cells and temporal slices for drinking and smoking actions on the *C&C* test dataset averaged over three runs.

Localization results & comparison to state of the art. Figure 7.5 presents precision-recall curves for localizing drinking and smoking actions in *C&C*. The detectors are trained on the training part of each dataset and evaluated on the corresponding test sets. Figure 7.5 (left) evaluates the detection results for localizing *drinking* actions. Under the same experimental setup, the linear classifier (51.8%) substantially outperforms the state-of-the-art, i.e., Willems et al. [2009] (45.2%) and Laptev and Perez [2007] (43.4%). The non-linear classifier further improves the results (55.4%). Note the excellent precision (100%) up to a recall of ca. 30%. Figure 7.6 illustrates the corresponding top 12 drinking localizations ordered by their SVM score. Note the variety of camera viewpoints and lighting.

Figure 7.5 (right) evaluates the detection results for localizing *smoking* actions. The non-linear classifier turns out to be crucial, improving the performance by +6.1% to 22.8% in terms of AP. The noticeably lower performance for smoking (when compared to drinking) can be explained by the large intra-class variability of this action. Temporal boundaries of a smoking action can in fact be only loosely defined and smoking often happens in parallel with other activities (like talking or drinking). Furthermore, a cigarette is smaller and less distinctive than a cup. Previous action analysis on this dataset [Laptev and Perez, 2007, Willems et al., 2009] did not include smoking, so no comparisons can be given. The top 12 smoking localizations are shown in figure 7.7. Interestingly, some of the false positives (e.g., rank 4, 10) include rapid vertical motion of the hand towards head and mouth.

Since drinking and smoking actions seem to be visually similar, it is interesting to assess the discriminative power of both classifiers. For this, we measure the performance of a drinking classifier for the task of localizing smoking and vice versa. Table 7.2 displays the *confusion* between the actions drinking and smoking. In both cases the performance is very low (around 5% AP) which shows that both classifiers are able to learn discriminative models that can distinguish visually similar, yet different actions successfully.

Comparison with other action descriptors. To show the importance of computing the *HOG-Track* descriptor on the spatial extent of humans determined by tracks, we

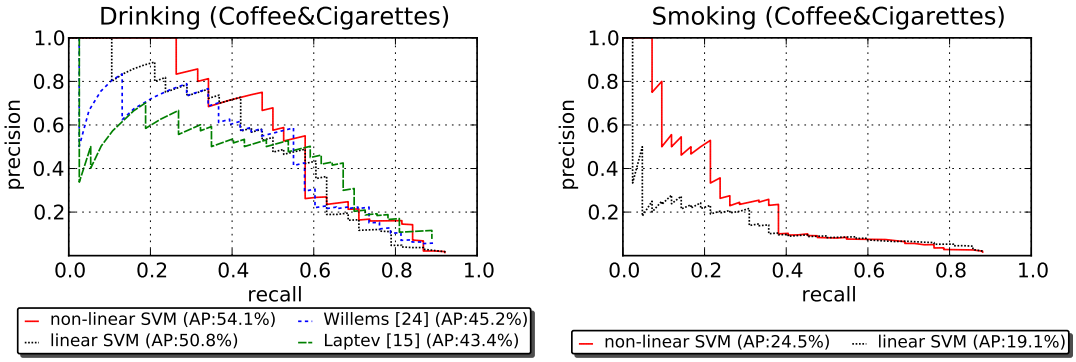


Figure 7.5: Precision-recall curves on the *C&C* test set. Human actions evaluated: drinking (left) and smoking (right). We compare our linear and non-linear detectors and report state-of-the-art results where applicable.

conduct experiments with a number of baseline classifiers. We keep the experimental setup and descriptor parameters the same.

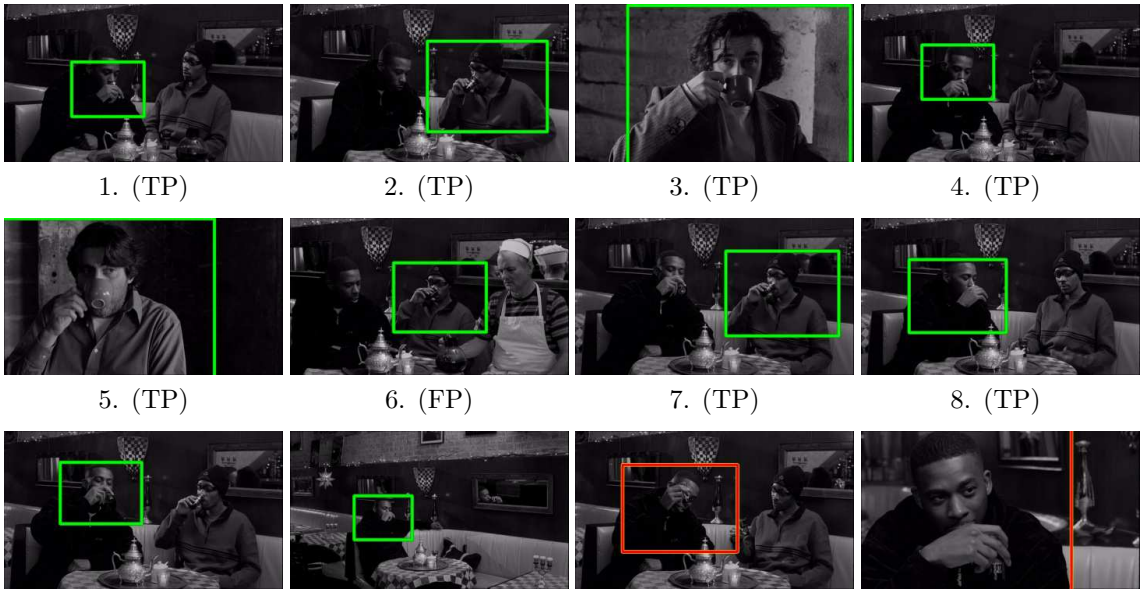
First, we extract our spatio-temporal descriptor for the entire video frame, i.e., ignore the tracks. In this case the evaluation criterion only measures the overlap in time, as we do not determine the spatial extent. The average precision for the linear baseline classifier on the *C&C* drinking dataset is 8.1% (vs 51.8% with tracks) and for the non-linear one it is 17.1% (vs 55.4%). Clearly, such baseline is able to localize drinking actions to some extent, but its performance is inferior without the spatial localization provided by the tracks.

Next, we evaluate the importance of adapting the *HOG-Track* descriptor to tracks. We compute the descriptor for a spatio-temporal cuboid region tangent to the track. Precisely, we align the centre of the cuboid with the track, but do not "bend" it along the track. The performance for the linear classifier on drinking is 28.9% (vs 51.8% with adaptation) and this improves to 48.1% (vs 55.4%) with the non-linear classifier. This confirms the importance of descriptor adaptation.

Finally, we further evaluate the cuboid representation by performing an exhaustive (i.e., not using tracks) spatio-temporal search for an action. The non-linear classifier achieves an AP of 24.3% (vs 55.4%) for drinking. Figure 7.8 compares all these different methods. We also include results for the exhaustive cuboid search carried out by Laptev and Perez [2007]. Overall, using tracks to drive the action localization significantly outperforms the other approaches.

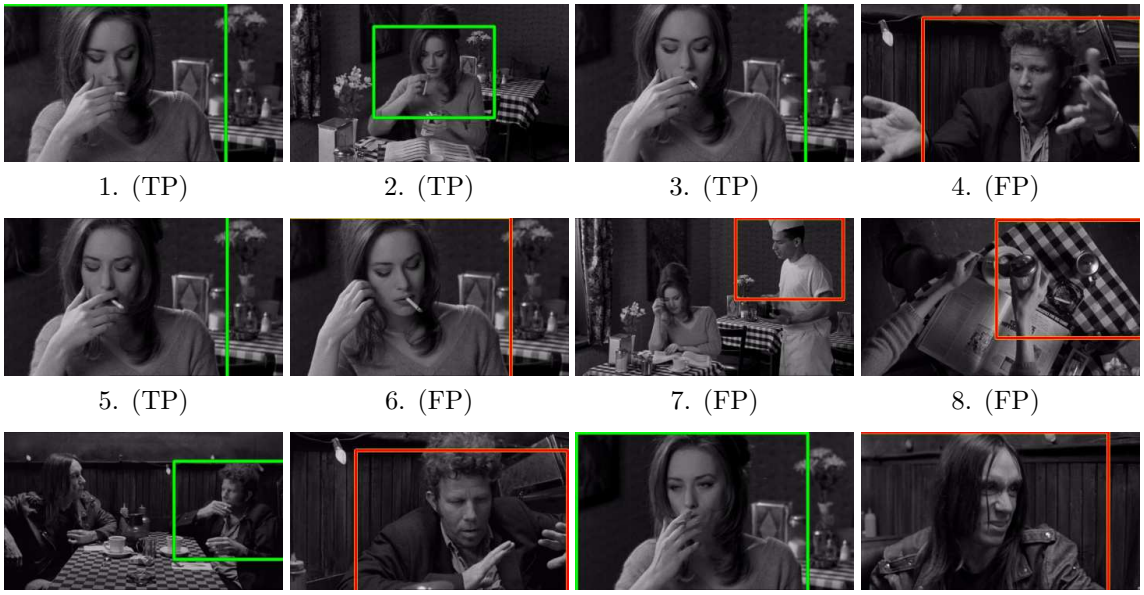
	Drinking action	Smoking action
Drinking detector	55.4%	5.3%
Smoking detector	5.0%	22.8%

Table 7.2: Performance (AP) of drinking and smoking classifiers when localizing drinking and smoking actions. Note that the classifiers do not confuse the actions.



1. (TP) 2. (TP) 3. (TP) 4. (TP)
5. (TP) 6. (FP) 7. (TP) 8. (TP)
9. (TP) 10. (TP) 11. (FP) 12. (FP)

Figure 7.6: The twelve highest ranked drinking detections on *C&C*.



1. (TP) 2. (TP) 3. (TP) 4. (FP)
5. (TP) 6. (FP) 7. (FP) 8. (FP)
9. (TP) 10. (FP) 11. (TP) 12. (FP)

Figure 7.7: The twelve highest ranked smoking detections on *C&C*.

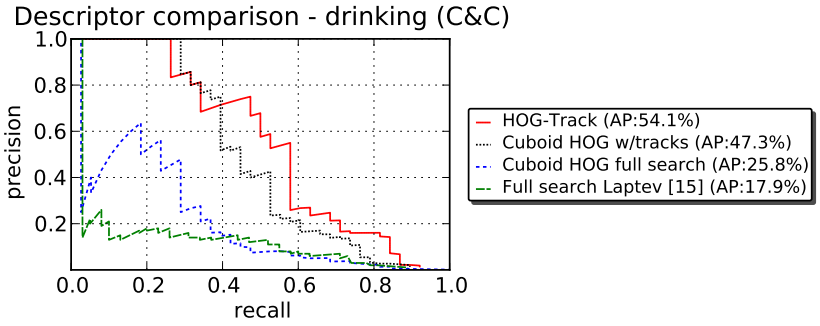


Figure 7.8: Precision-recall curves comparing HOG-Track to other action descriptors on *C&C* for the action drinking.

Complexity. In the following we investigate the theoretical and practical time complexity of our localization approach. We also discuss memory requirements and compare to an exhaustive "sliding cuboid" baseline.

For the theoretical analysis, without loss of generality we assume a linear one-against-rest classifier. We consider the number of multiplications in classifier evaluation (i.e., computing the dot product in the linear case) as the complexity measure. In a standard sliding window scheme the classifier is evaluated once for each window. Consequently, the total recognition cost will linearly depend on (a) the number of actions considered, (b) the number of windows evaluated, and (c) the dimensionality of the descriptor. The complexity of the "sliding cuboid" baseline can therefore be written as $O(a \cdot s_x^2 s_t \cdot r_x^2 r_t)$ where a is the number of actions, s_x/s_t denote spatial/temporal size of the problem (video), and r_x/r_t correspond to spatial/temporal resolution (dimensionality) of the descriptor.

Our approach combines a spatial sliding window human classifier and a temporal detector. Its complexity can be written as $O(s_x^2 s_t \cdot r_x^2 + a \cdot t s_t \cdot r_x^2 r_t)$ where t corresponds to the number of tracks in the video. Note that the above expression is normally dominated by the spatial search (left term). Compared to the exhaustive approach, we gain from having an action-agnostic classifier (no factor a) and using a simpler detector first (no factor r_t). The temporal search (right term) is fast since it searches only one dimension and $t \ll s_x^2$.

In practice, the difference in the runtime is even more significant due to limited memory. Computing the video descriptor does not allow for many optimizations which are possible for a single frame/image – like pre-computing or caching the gradient histograms for instance. This in practice adds another factor to the sliding cuboid complexity. It does not affect our method since in our case the complexity is dominated by human detection, where memory requirements are not a problem.

The theoretical analysis above is confirmed in practice. Processing about 25 minutes of video using our method takes about 13 hours in total on a standard workstation. Human detection takes under 10 hours, tracking humans adds 3 hours, action localization is performed in under 10 minutes. For comparison, running an exhaustive cuboid search on the same data takes over 100 hours.

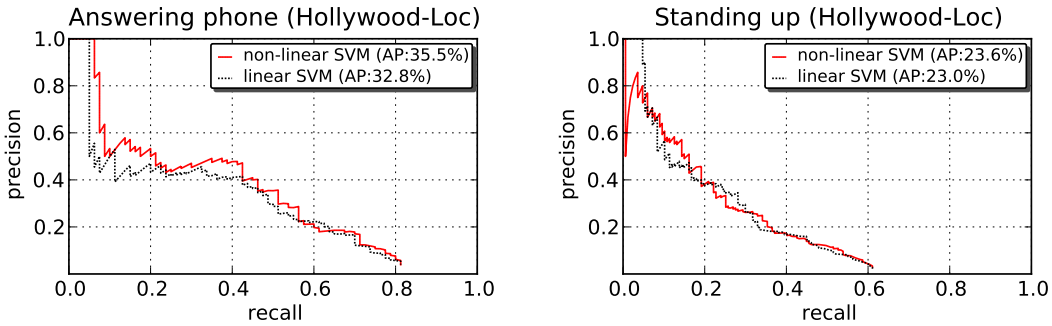


Figure 7.9: Precision-recall curves for the actions answering phone and standing-up of the *Hollywood–Localization* test set.

7.5.2 Hollywood–Localization

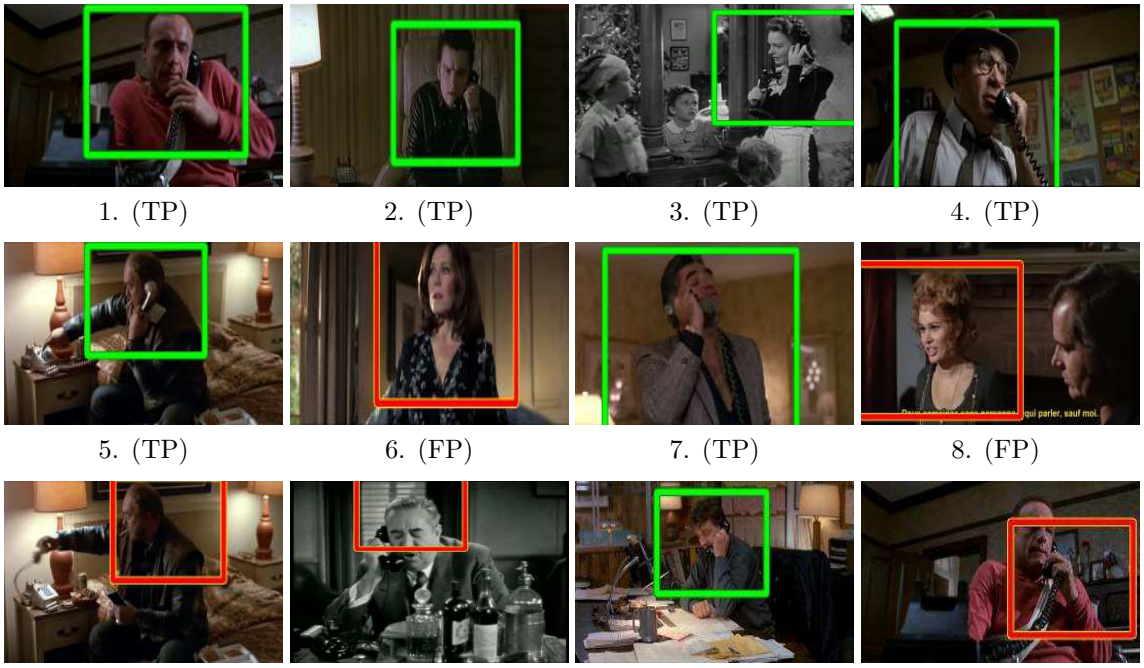
For this dataset we use the same parameters throughout as those used for *C&C*. Figure 7.9 (left) evaluates the detection results for localizing *phoning* actions in our *Hollywood–Localization* dataset. Due to the much larger variety of the videos (Hollywood movies), this dataset is much more challenging than *C&C*. The difficulty of the task is further increased by the fact that negative samples contain, without exception, other dynamic human actions. Some of those actions, like eating for example, might share similar motion patterns. Nevertheless, the recognition performance is satisfactory. In almost 40 minutes of video we can correctly localize over 80% of phoning actions and retrieve the top ones with high precision. The top 12 phoning localizations on the test set are shown in figure 7.10. The true positive detections cover a large variety of poses and scenes. The top false positives detections mostly involve a rapid vertical hand movement.

Figure 7.9 (right) evaluates the detection results for localizing *standing-up* actions, and figure 7.11 shows the top 12 detections. This action differs from the previous three as it does not involve the hand moving towards the head. The results are promising; the recall is worse than for all the other classes, but the precision is satisfactory.

7.6 Conclusion

We have demonstrated the value of using human tracks for visual action localization. In each dataset the same tracks support localization of different types of actions. This allows natural human actions to be effectively recognized in challenging environments.

A track introduces a separation between the human foreground and background of a scene, and either or both may provide information. In this paper we have proposed a robust model for foreground regions. In the future, given this separation, appropriate descriptors and classifiers can then be learnt for the foreground and background regions. For example, if the camera is panning to follow a person, then the motion from the background can be suppressed. However, for some actions it will be the background (the context) or background motion that is more informative, e.g. perhaps in the case of a person standing up.



1. (TP) 2. (TP) 3. (TP) 4. (TP)
5. (TP) 6. (FP) 7. (TP) 8. (FP)
9. (FP) 10. (FP) 11. (TP) 12. (FP)

Figure 7.10: The twelve highest ranked phoning actions detected on *Hollywood–Localization*.

1. (TP) 2. (FP) 3. (TP) 4. (TP)
5. (TP) 6. (TP) 7. (TP) 8. (FP)
9. (TP) 10. (TP) 11. (FP) 12. (TP)

Figure 7.11: The twelve highest ranked standing-up actions detected on *Hollywood–Localization*.

Conclusion

Cette thèse a présenté et évalué plusieurs contributions pour la reconnaissance d'actions dans des vidéos réalistes. Pour conclure notre travail, nous résumons dans la suite nos conclusions principales.

Notre première contribution est un descripteur local basé sur des histogrammes d'orientation de gradients spatio-temporels (HOG3D) que nous avons évalué pour la tâche de reconnaissance d'actions. Les expériences ont montré l'importance des paramètres adaptés aux tâches. En comparaison directe avec des descripteurs de pointe, notre approche a montré de meilleurs résultats sur trois des quatre bases de données considérées.

Nous avons évalué et comparé les méthodes existantes pour la détection et la description de caractéristiques locales pour pour la tâche de classification d'action. Parmi les détecteurs, le détecteur de Gabor a montré de bons résultats et il a en général atteint la couverture spatio-temporelle la plus dense. Parmi les descripteurs, HOG/HOF et notre descripteur HOG3D ont obtenu les meilleurs résultats.

Une autre contribution est un descripteur pour la reconnaissance d'actions basé sur des trajectoires locales. Dans nos expériences, l'extension du descripteur de trajectoires avec des informations d'apparence et de mouvement dans le voisinage local de la trajectoire a été l'élément clé pour améliorer la performance. En comparaison avec l'état de l'art actuel, des méthodes de pointe, nous obtenons des résultats comparables pour une base de données et significativement meilleurs sur deux autres bases.

Nous avons étudié la représentation par sac-de-mots avec la détection de personne et nous avons quantifié son gain pour la reconnaissance d'actions. De nos expériences, nous avons conclu que la suppression de fond ne conduit qu'à un gain de performance mineur, car elle supprime l'information de contexte qui peut pourtant être utile pour la classification. Pour quelques catégories d'action (sortir de la voiture et embrasser, par exemple), nous avons même observé une dégradation des performances. En outre, nous avons montré qu'en général des informations structurelles permettent d'améliorer la précision de reconnaissance. Seulement pour certaines catégories d'action, nous n'avons observé aucune ou uniquement une mineure amélioration.

Notre dernière contribution-clé consiste en une approche centrée sur des personnes pour localiser des actions humaines dans des films hollywoodiens temporellement ainsi que spatialement. Nos expériences ont montré que des détections de personne sont en mesure d'améliorer non seulement l'efficacité du calcul, mais elles contribuent aussi à augmenter la précision de la reconnaissance grâce à une description plus sophistiquée. Dans nos évaluations, notre approche dépasse l'état de l'art actuel de 9% de précision moyenne, et elle a montré des résultats prometteurs sur notre nouvelle base de données constituée sur de films hollywoodiens.

8

Conclusion and perspectives

Contents

This dissertation has presented and evaluated several contributions for action recognition in realistic video data. To conclude our work, we summarize our key contributions and discuss conclusions from our experiments in section 8.1. Based on these conclusions, we will then indicate interesting directions for future research in this field (section 8.2).

8.1 Key contributions

Local descriptor based on histograms of 3D gradients. Our first contribution is a local descriptor based on histograms of oriented spatio-temporal gradients (HOG3D) which we evaluated for the task of action recognition. In order to quantize gradient orientations, we introduced an approach using regular polyhedrons which we compared to quantization based on spherical coordinates. For gradient computation of arbitrary scales, we extended the concept of integral images to integral videos. Parameters were evaluated in depth and optimized for action recognition on realistic as well as simplified video data. Experiments showed the importance of task-specific parameter settings. In direct comparison with a current state-of-the-art descriptor, our approach improved results on three out of four datasets.

Evaluation of local space-time features. We have evaluated and compared existing methods for feature detection and description on action classification tasks. For this, a standard bag-of-features approach was employed and experiments were carried out on three different datasets with a total of 25 action classes. Our conclusions are that dense sampling in general outperforms interest point detectors on realistic data, while Harris3D works better on simple data (*KTH* actions dataset). Among the detectors, the Gabor detector showed good results and provided the densest coverage. Among the descriptors, HOG/HOF and our HOG3D descriptor showed best results.

Local feature trajectories. A further contribution is a descriptor for action recognition based on feature trajectories. Contrary to existing methods which solely used trajectory shape information, we extended the trajectory description with additional descriptors capturing appearance and motion information in the local neighborhood of the trajectory. In our experiments, this extension showed to be the key element for improving performance. Furthermore, we have introduced a descriptor based on motion boundary histograms (MBH). It showed excellent results alone and in combination with descriptors based on gradient and optical flow orientations. We evaluated the full descriptor on three different datasets and optimized its parameters for realistic video settings. In comparison with current state-of-the-art methods, we are on par for one dataset and improve significantly for the two others.

Combination of bag-of-features with human detection. We have investigated the combination of the bag-of-features representation with person localization (human tracks) and quantified its benefit for action recognition. To accomplish this, we redefined the concept of spatial pyramids as a model with controlled levels of spatial constraints. In a first step, we considered simple background suppression and concluded that it leads only to a minor performance gain since context information can play an important role in classifying actions, especially in the case of realistic videos. For a few action classes (getting out of the car and kissing), we even observed performance degradation. In a second step, we showed that narrowing down attention to human actors allows to incorporate more layout information which, in general, helps improving recognition accuracy. However, on realistic videos and for some action classes, we observed no or only minor improvement.

Action localization in realistic video data. Our last key contribution consists of a human-centric approach to localize human actions temporally as well as spatially in Hollywood-style movie data. To allow for robust localization of humans, we have developed an upper-body human tracker that is able to cope with realistic video settings. For the action representation, we have introduced a spatio-temporal HOG descriptor adapted to human tracks. Our experiments have shown that tracks improve not only computational efficiency, but they also help to increase recognition accuracy due to a more principled action description. In the evaluations, our approach exceeded the current state-of-the-art by 9% average precision and showed promising results on our new dataset based on actions from Hollywood movies.

8.2 Future work

Towards realistic action recognition. In sections 3.3 and 6.3, we discussed classification performance per action class and concluded that each class has specific characteristics that could benefit from an adapted description. This has been especially obvious for actions in Hollywood movies. Consequently, it seems necessary to adapt the visual description method to each type of action individually. One aspect is the parametrization of a specific descriptor. A second aspect is the combination of different information

cues, such as appearance, motion, structure [Laptev et al., 2008], and context information [Marszałek et al., 2009], but also the presence or absence of certain objects [Han et al., 2009]. Multiple Kernel Learning [Sun et al., 2009, Han et al., 2009] shows promising results as a late-fusion technique, but also early-fusion approaches can be important (*cf.* chapter 5). Other directions included hybrid fusion approaches, e.g., Khan et al. [2009] use color attention to pre-weight quantized features for shape information.

Action modeling with feature trajectories. One limitation of bag-of-features representations is that mutual information of neighboring features cannot be modeled and is thus lost. Some recent approaches [Sun et al., 2009, Gilbert et al., 2009] propose to overcome this limitation by combining features in a local context.

In chapter 5, we have shown that local feature trajectories in combination with appearance and motion descriptors yield excellent results. Since trajectories follow local movements over time, they offer interesting possibilities for more principled action modeling. Matikainen et al. [2009] have shown that trajectories of similar shape can be grouped together. An interesting possibility is to use this grouping of local features in order to model relations between local regions with coherent motion.

Motion boundary histograms for action recognition. As motion boundary histograms (MBH) showed excellent results for action recognition using local features (*cf.* chapter 5), their application to other problems seems appealing and should certainly be investigated. A possible application is our system for action localization. Especially in Hollywood movies, we noted the presence of camera ego-motion which is explicitly encoded into descriptors based on optical flow and spatio-temporal gradients. In contrast, motion boundaries are invariant to camera motion and can thus help to improve results.

For the MBH descriptor itself, it has been shown [Dalal et al., 2006] that the underlying optical flow algorithm can play an important role. Since it is currently not clear to which extent different algorithms influence the performance for action recognition, this should be investigated in the future.

Human tracks for action localization. Our approach to localize human actions in realistic video settings improved results over the current state-of-the-art significantly. An interesting path for future work can be based on human tracks for multiple body parts, e.g., for head, upper body, and full body. First, this can help to render the tracking process more robust since additional constraints for relations between the body parts are available [Mikolajczyk et al., 2004]. Second, it can also allow for a more principled action localization: actions can be learned for each body part separately, and they can be evaluated jointly for localization. A further possibility is to incorporate multi-view information in action modeling, i.e., explicitly modeling of an action for frontal and lateral views. This can enable a more discriminative action modeling.

A

Common methods

A.1 Bag-of-features

Various contributions that have been proposed in this dissertation make use of the *bag-of-features* (BoF) representation for action classification. Since a very similar representation is employed in the different chapters, we will detail it in the following here.

A bag-of-features representation for video sequence is a loose representation of a set of local space-time features (*cf.* section 2.1.3) If not otherwise stated, we obtain a sparse set of spatio-temporal interest points by applying the space-time extension of the *Harris operator* [Laptev, 2005] (*cf.* section 4.2.1).

The bag-of-features representation requires a *visual vocabulary*. For this, we apply either random sampling or k-means on the set of training features. *Random sampling* has the advantage that it is very fast since only a subset of V random training features needs to be computed. For results using *k-means*, we cluster a subset of 100,000 randomly selected training features in order to limit computational complexity. We increase precision by initializing k-means 8 times and keeping the result with the lowest error. Features are assigned to their closest vocabulary word using Euclidean distance. The resulting histograms of visual word occurrences are used as video sequence representations.

Unless otherwise stated, we fix the number of visual words to $V = 4000$ which has shown to empirically give good results for a wide range of datasets [Laptev et al., 2008]. We also observed in our experiments, that results using random sampling were close to those obtained using vocabularies built with k-means.

Classification is done with non-linear support vector machines χ^2-kernel [Belongie et al., 2002]

$$K(H_i, H_j) = exp\left(-\frac{1}{A}D(H_i, H_j)\right) \quad , \tag{A.1}$$

where $H_i = \{h_{ik}\}$ and $H_j = \{h_{jk}\}$ are the histograms of word occurrences, $D(\cdot)$ is the χ^2-distance defined as

$$D(H_i, H_j) = \frac{1}{2}\sum_k \frac{(h_{ik} - h_{jk})^2}{h_{ik} + h_{jk}} \quad , \tag{A.2}$$

and A is the average distance between all N training samples [Zhang et al., 2007]:

$$A = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} D(H_i, H_j). \tag{A.3}$$

For multi-class classification, we use the *one-against-rest* approach. In our implementation, we use the code provided by LIBSVM [Chang and Lin, 2001].

Multi-channel classification. In the case of classification with multiple histogram types, we employ a multi-channel Gaussian kernel [Zhang et al., 2007]

$$K(H_i, H_j) = \exp\left(- \sum_t \frac{1}{A_t} D\left(H_i^{(t)}, H_j^{(t)}\right) \right) \quad , \tag{A.4}$$

where $H_i^{(t)} = \{h_{ik}^{(t)}\}$ and $H_j^{(t)} = \{h_{jk}^{(t)}\}$ are histograms of the type t, A_t is the average distance between all training samples for histogram type t.

List of Figures

List of Tables

Bibliography

A. Agarwal and B. Triggs. Recovering 3D human pose from monocular images. *IEEE T-PAMI*, 28, 2006.

J. K. Aggarwal and Q. Cai. Human motion analysis: A review. *CVIU*, 73:428–440, 1999.

S. Ali, A. Basharat, and M. Shah. Chaotic invariants for human action recognition. In *ICCV*, 2007.

F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *ICML*, 2004. ISBN 1-58113-828-5.

H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *ECCV*, 2006.

P. Beaudet. Rotationally invariant image operators. In *ICPR*, 1978.

S. Belongie, C. Fowlkes, F. R. K. Chung, and J. Malik. Spectral partitioning with indefinite kernels using the nyström extension. In *ECCV*, 2002.

M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, 2005.

A.F. Bobick and J.W. Davis. The recognition of human movement using temporal templates. *IEEE T-PAMI*, 23:257–267, 2001.

J.-Y. Bouguet. Pyramidal implementation of the Lucas-Kanade feature tracker description of the algorithm. Technical report, Intel Corporation, Microprocessor Research Labs, 1999.

M. Bregonzio, S. Gong, and T. Xiang. Recognising action as clouds of space-time interest points. In *CVPR*, 2009.

C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

H. Buxton. Learning and understanding dynamic scene activity: a review. *Image and Vision Computing*, 21:125–136, 2003.

C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001.

T. Cour, C. Jordan, E. Miltsakaki, and B. Taskar. Movie/script: Alignment and parsing of video and text transcription. In *ECCV*, 2008.

G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV*, 2004.

O.G. Cula and K.J. Dana. Compact representation of bidirectional texture functions. In *CVPR*, 2001.

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *ECCV*, 2006.

P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, 2005.

A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *ICCV*, 2003.

M. Everingham, J. Sivic, and A. Zisserman. Hello! my name is... buffy – automatic naming of characters in tv video. In *BMVC*, 2006.

M. Everingham, L. van Gool, C. Williams, J. Winn, and A. Zisserman. Overview and results of classification challenge, 2007. In *The PASCAL VOC'07 Challenge Workshop, in conj. with ICCV*, 2008.

M. Everingham, J. Sivic, and A. Zisserman. Taking the bite out of automatic naming of characters in TV video. *Image and Vision Computing*, 27:545–559, 2009a.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge (VOC) Results, 2009b.

A. Fathi and G. Mori. Action recognition by learning mid-level motion features. In *CVPR*, 2008.

P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE T-PAMI*, to appear, 2010.

V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *CVPR*, 2008.

F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. Multi-camera people tracking with a probabilistic occupancy map. *IEEE T-PAMI*, 30:267–282, 2008.

W. Freeman and M. Roth. Orientation histogram for hand gesture recognition. In *FG*, 1995.

Y. Freund and R. Schapire. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14:771–780, 1999.

D. M. Gavrila. The visual analysis of human movement: a survey. *CVIU*, 73:82–98, 1999.

A. Gilbert, J. Illingworth, and R. Bowden. Fast realistic multi-action recognition using mined dense spatial-temporal features. In *ICCV*, 2009.

Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. *Transactions on Pattern Analysis and Machine Intelligence*, 29: 2247–2253, 2007.

D. Han, L. Bo, and C. Sminchisescu. Selection and context for action recognition. In *ICCV*, 2009.

I. Haritaoglu, D. Harwood, and L. S. Davis. W4: Real-time surveillance of people and their activities. *IEEE T-PAMI*, 22:809–830, 2000.

C. Harris and M.J. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, 1988.

H. Harzallah, F. Jurie, and C. Schmid. Combining efficient object localization and image classification. In *ICCV*, 2009.

P. Havaldar. Sony pictures imageworks. In *SIGGRAPH Courses*, 2006. ISBN 1-59593-364-6.

T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR*, 1999. ISBN 1-58113-096-1.

W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man and Cybernetics*, 34:334–352, 2004.

Y. Hu, L. Cao, F. Lv, S. Yan, Y. Gong, and T. S. Huang. Action detection in complex scenes with spatial and temporal ambiguities. In *ICCV*, 2009.

H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In *ICCV*, 2007.

G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception and Psychophysics*, 14:201–211, 1973.

F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *ICCV*, 2005.

Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *ICCV*, 2005.

Y. Ke, R. Sukthankar, and M. Hebert. Spatio-temporal shape and flow correlation for action recognition. In *Visual Surveillance Workshop*, 2007a.

Y. Ke, R. Sukthankar, and M. Hebert. Event detection in crowded videos. In *ICCV*, 2007b.

F. Shahbaz Khan, J. van de Weijer, and M. Vanrell. Top-down color attention for object recognition. In *ICCV*, 2009.

T.-K. Kim, S.-F. Wong, and R. Cipolla. Tensor canonical correlation analysis for action classification. In *CVPR*, 2007.

A. Kläser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3D-gradients. In *BMVC*, 2008.

A. Kläser, M. Marszałek, C. Schmid, and A. Zisserman. Human focused action localization in video. Submitted to International Workshop on Sign, Gesture, Activity in conjunction with ECCV, 2010.

I. Laptev. *Local Spatio-Temporal Image Features for Motion Interpretation.* PhD thesis, Department of Numerical Analysis and Computer Science (NADA), KTH, 2004.

I. Laptev. On space-time interest points. *IJCV*, 64:107–123, 2005.

I. Laptev. Improvements of object detection using boosted histograms. In *BMVC*, 2006.

I. Laptev and T. Lindeberg. Space-time interest points. In *ICCV*, 2003.

I. Laptev and T. Lindeberg. Local descriptors for spatio-temporal recognition. In *SCVMA*, 2004.

I. Laptev and P. Perez. Retrieving actions in movies. In *ICCV*, 2007.

I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

B. Leibe, K. Schindler, and L. van Gool. Coupled detection and trajectory estimation for multi-object tracking. In *ICCV*, 2007.

B. Li, Q. Meng, and H. Holstein. Articulated motion reconstruction from feature points. *Pattern Recognition*, 41:418 – 431, 2008.

T. Lindeberg. Feature detection with automatic scale selection. *IJCV*, 30(2):79–116, 1998.

J. Liu and M. Shah. Learning human actions via information maximization. In *CVPR*, 2008.

J. Liu, S. Ali, and M. Shah. Recognizing human actions using multiple features. In *CVPR*, 2008.

J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos "in the wild". In *CVPR*, 2009.

D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

W.-L. Lu and J. J. Little. Simultaneous tracking and action recognition using the pca-hog descriptor. In *CRV*, 2006.

B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, 1981.

M. Marszałek, I. Laptev, and C. Schmid. Actions in context. In *CVPR*, 2009.

P. Matikainen, M. Hebert, and R. Sukthankar. Trajectons: Action recognition through the motion analysis of tracked features. In *ICCV workshop on Video-oriented Object and Event Classification*, 2009.

P. Matikainen, M Hebert, and R. Sukthankar. Representing pairwise spatial and temporal relations for action recognition. In *ECCV*, 2010.

R. Medina-Carnicer, J.L. Garrido-Castro, E. Collantes-Estevez, and A. Martinez-Galisteo. Fast detection of marker pixels in video-based motion capture systems. *Pattern Recognition Letters*, 30:432 – 439, 2009.

R. Messing, C. Pal, and H. Kautz. Activity recognition using the velocity histories of tracked keypoints. In *ICCV*, 2009.

Microsoft. Project natal. http://www.xbox.com/projectnatal/, 2009.

K. Mikolajczyk and U. Hirofumi. Action recognition with motion-appearance vocabulary forest. In *CVPR*, 2008.

K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE T-PAMI*, 27:1615–1630, 2005.

K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In *ECCV*, 2004.

T. B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *CVIU*, 81:231–268, 2001.

T. B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *CVIU*, 104(2-3):90–126, 2006.

J. Niebles and L. Fei-Fei. A hierarchical model of shape and appearance for human action classification. In *CVPR*, 2007.

J. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. In *BMVC*, 2006.

J. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *IJCV*, 79:299–318, 2008.

E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In *ECCV*, 2006.

A. Oikonomopoulos, I. Patras, and M. Pantic. Spatio-temporal salient points for visual recognition of human actions. *IEEE Trans. Systems, Man, and Cybernetics, Part B*, 36 (3):710–719, 2006.

V. Parameswaran and R. Chellappa. View invariance for human action recognition. *IJCV*, 66:83–101, 2006.

K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572, 1901.

R. Polana and R. Nelson. Low level recognition of human motion. In *IEEE Workshop on Nonrigid and Articulate Motion*, 1994.

R. Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28:976–990, 2010.

L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, 1989.

D. Ramanan, D.A. Forsyth, and A. Zisserman. Tracking people by learning their appearance. *IEEE T-PAMI*, 29:65, 2007.

E. Ramasso, C. Panagiotakis, M. Rombaut, D. Pellerin, and G. Tziritas. Human shape-motion analysis in athletics videos for coarse to fine action/activity recognition using transferable belief model. *Electronic Letters on Computer Vision and Image Analysis*, 7:32–50, 2009.

M. Rodriguez, J. Ahmed, and M. Shah. Action mach: A spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, 2008.

G. Salton. *Automatic Information Organization and Retrieval.* McGraw Hill Text, 1968.

K. Schindler and L. van Gool. Action snippets: How many frames does human action recognition require? In *CVPR*, 2008.

C. Schüldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. In *ICPR*, 2004.

P. Scovanner, S. Ali, and M. Shah. A 3-dimensional SIFT descriptor and its application to action recognition. In *MULTIMEDIA*, 2007.

A. Senior. An introduction to automatic video surveillance. In *Protecting Privacy in Video Surveillance*. Springer, 2009.

J. Shi and C. Tomasi. Good features to track. In *CVPR*, 1994.

J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman. Discovering objects and their location in images. In *ICCV*, 2005.

J. Sullivan and S. Carlsson. Recognizing and tracking human action. In *ECCV*, 2002.

J. Sun, X. Wu, S. Yan, L.-F. Cheong, T.-S. Chua, and J. Li. Hierarchical spatio-temporal context modeling for action recognition. In *CVPR*, 2009.

P. Turaga, R. Chellappa, V.S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 18:1473–1488, 2008.

H. Uemura, S. Ishikawa, and K. Mikolajczyk. Feature tracking and motion compensation for action recognition. In *BMVC*, 2008.

R. Urtasun, D. J. Fleet, and P. Fua. Temporal motion models for monocular and multiview 3d human body tracking. *CVIU*, 104:157–177, 2006.

A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *ICCV*, 2009.

P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001.

H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *BMVC*, 2009.

L. Wang and D. Suter. Recognizing human activities from silhouettes: Motion subspace and factorial discriminative graphical model. In *CVPR*, 2007.

D. Weinland and E. Boyer. Action recognition using exemplar-based embedding. In *CVPR*, 2008.

D. Weinland, E. Boyer, and R. Ronfard. Action recognition from arbitrary views using 3D exemplars. In *ICCV*, 2007.

D. Weinland, R. Ronfard, and E. Boyer. A survey of vision-based methods for action representation, segmentation and recognition. Technical report, INRIA, 2010.

Wikipedia. Platonic solid — Wikipedia, the free encyclopedia, 2010. URL http://en.wikipedia.org/wiki/Platonic_solid. [Online; accessed 20-Mai-2010].

G. Willems, T. Tuytelaars, and L. Van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *ECCV*, 2008.

G. Willems, J. H. Becker, T. Tuytelaars, and L. van Gool. Exemplar-based action recognition in videos. In *BMVC*, 2009.

S.-F. Wong, T.-K. Kim, and R. Cipolla. Learning motion categories using both semantic and structural information. In *CVPR*, 2007.

S.F. Wong and R. Cipolla. Extracting spatio-temporal interest points using global information. In *ICCV*, 2007.

J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden markov model. In *CVPR*, 1992.

A. Yilmaz and M. Shah. Actions sketch: A novel action representation. In *CVPR*, 2005a.

A. Yilmaz and M. Shah. Recognizing human actions in videos acquired by uncalibrated moving cameras. In *ICCV*, 2005b.

J. Yuan, Z. Liu, and Y. Wu. Discriminative subvolume search for efficient action detection. In *CVPR*, 2009.

J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *IJCV*, 73(2): 213–238, 2007.

Z. Zhang, Y. Hu, S. Chan, and L.-T. Chia. Motion context: A new representation for human action recognition. In *ECCV*, 2008.

Abstract

This dissertation targets the recognition of human actions in realistic video data, such as movies. To this end, we develop state-of-the-art feature extraction algorithms that robustly encode video information for both, action classification and action localization.

In a first part, we study bag-of-features approaches for action classification. Recent approaches that use bag-of-features as representation have shown excellent results in the case of realistic video data. We, therefore, conduct an extensive comparison of existing methods for local feature detection and description. We, then, propose two new approaches to describe local features in videos. The first method extends the concept of histograms over gradient orientations to the spatio-temporal domain. The second method describes trajectories of local interest points detected spatially. Both descriptors are evaluated in a bag-of-features setup and show an improvement over the state-of-the-art for action classification.

In a second part, we investigate how human detection can help action recognition. Firstly, we develop an approach that combines human detection with a bag-of-features model. The performance is evaluated for action classification with varying resolutions of spatial layout information. Next, we explore the spatio-temporal localization of human actions in Hollywood movies. We extend a human tracking approach to work robustly on realistic video data. Furthermore we develop an action representation that is adapted to human tracks. Our experiments suggest that action localization benefits significantly from human detection. In addition, our system shows a large improvement over current state-of-the-art approaches.

Keywords: computer vision, action recognition, video, image, classification, local descriptors, bag-of-features, detection.

Résumé

Cette thèse s'intéresse à la reconnaissance des actions humaines dans des données vidéo réalistes, tels que les films. À cette fin, nous développons des algorithmes d'extraction de caractéristiques visuelles pour la classification et la localisation d'actions.

Dans une première partie, nous étudions des approches basées sur les sacs-de-mots pour la classification d'action. Dans le cas de vidéo réalistes, certains travaux récents qui utilisent le modèle sac-de-mots pour la représentation d'actions ont montré des résultats prometteurs. Par conséquent, nous effectuons une comparaison approfondie des méthodes existantes pour la détection et la description des caractéristiques locales. Ensuite, nous proposons deux nouvelles approches pour la descriptions des caractéristiques locales en vidéo. La première méthode étend le concept d'histogrammes sur les orientations de gradient dans le domaine spatio-temporel. La seconde méthode est basée sur des trajectoires de points d'intérêt détectés spatialement. Les deux descripteurs sont évalués avec une représentation par sac-de-mots et montrent une amélioration par rapport à l'état de l'art pour la classification d'actions.

Dans une seconde partie, nous examinons comment la détection de personnes peut contribuer à la reconnaissance d'actions. Tout d'abord, nous développons une approche qui combine la détection de personnes avec une représentation sac-de-mots. La performance est évaluée pour la classification d'actions à plusieurs niveaux d'échelle spatiale. Ensuite, nous explorons la localisation spatio-temporelle des actions humaines dans les films. Nous étendons une approche de suivi de personnes pour des vidéos réalistes. En outre, nous développons une représentation d'actions qui est adaptée aux détections de personnes. Nos expériences suggèrent que la détection de personnes améliore significativement la localisation d'actions. De plus, notre système montre une grande amélioration par rapport à l'état de l'art actuel.

Mots-clés : vision par ordinateur, reconnaissance d'actions, vidéo, image, classification, descripteurs locals, sac-de-mots, détection.