N°d'ordre NNT : 2017LYSEI060

THESE de DOCTORAT DE L'UNIVERSITE DE LYON
opérée au sein de
INSA Lyon

Ecole Doctorale N° 512
Informatique et Mathématiques

Spécialité/ discipline de doctorat :
Informatique

Soutenue publiquement le 07/07/2017, par :
Emre Dogan

Human Pose Estimation and Action Recognition by Multi-robot Systems

---

Devant le jury composé de :

Pellerin, Denis	PRU, Polytech Grenoble	Président
Fofi, David	PRU, Université de Bourgogne	Rapporteur
Vincent, Nicole	PRU, Université Paris Descartes	Rapporteure
Ducottet, Christophe	PRU, Université Jean Monnet	Examinateur
Teulière, Céline	MC, Université Clermont-Ferrand	Examinatrice
Baskurt, Atilla	PRU, INSA Lyon	Directeur de thèse
Wolf, Christian	MC/HDR, INSA Lyon	Co-directeur de thèse
Eren, Gonen	MC, Université Galatasaray	Co-directeur de thèse

Département FEDORA – INSA Lyon - Ecoles Doctorales – Quinquennal 2016-2020

SIGLE	ECOLE DOCTORALE	NOM ET COORDONNEES DU RESPONSABLE
CHIMIE	**CHIMIE DE LYON** **http://www.edchimie-lyon.fr** Sec : Renée EL MELHEM Bat Blaise Pascal 3e etage secretariat@edchimie-lyon.fr Insa : R. GOURDON	M. Stéphane DANIELE Institut de Recherches sur la Catalyse et l'Environnement de Lyon IRCELYON-UMR 5256 Équipe CDFA 2 avenue Albert Einstein 69626 Villeurbanne cedex directeur@edchimie-lyon.fr
E.E.A.	**ELECTRONIQUE, ELECTROTECHNIQUE, AUTOMATIQUE** **http://edeea.ec-lyon.fr** Sec : M.C. HAVGOUDOUKIAN Ecole-Doctorale.eea@ec-lyon.fr	M. Gérard SCORLETTI Ecole Centrale de Lyon 36 avenue Guy de Collongue 69134 ECULLY Tél : 04.72.18 60.97 Fax : 04 78 43 37 17 Gerard.scorletti@ec-lyon.fr
E2M2	**EVOLUTION, ECOSYSTEME, MICROBIOLOGIE, MODELISATION** **http://e2m2.universite-lyon.fr** Sec : Sylvie ROBERJOT Bât Atrium - UCB Lyon 1 04.72.44.83.62 Insa : H. CHARLES secretariat.e2m2@univ-lyon1.fr	M. Fabrice CORDEY CNRS UMR 5276 Lab. de géologie de Lyon Université Claude Bernard Lyon 1 Bât Géode 2 rue Raphaël Dubois 69622 VILLEURBANNE Cédex Tél : 06.07.53.89.13 cordey@univ-lyon1.fr
EDISS	**INTERDISCIPLINAIRE SCIENCES-SANTE** **http://www.ediss-lyon.fr** Sec : Sylvie ROBERJOT Bât Atrium - UCB Lyon 1 04.72.44.83.62 Insa : M. LAGARDE secretariat.ediss@univ-lyon1.fr	Mme Emmanuelle CANET-SOULAS INSERM U1060, CarMeN lab, Univ. Lyon 1 Bâtiment IMBL 11 avenue Jean Capelle INSA de Lyon 696621 Villeurbanne Tél : 04.72.68.49.09 Fax :04 72 68 49 16 Emmanuelle.canet@univ-lyon1.fr
INFOMATHS	**INFORMATIQUE ET MATHEMATIQUES** **http://infomaths.univ-lyon1.fr** Sec :Renée EL MELHEM Bat Blaise Pascal, 3e étage Tél : 04.72. 43. 80. 46 Fax : 04.72.43.16.87 infomaths@univ-lyon1.fr	M. Luca ZAMBONI Bâtiment Braconnier 43 Boulevard du 11 novembre 1918 69622 VILLEURBANNE Cedex Tél :04 26 23 45 52 zamboni@maths.univ-lyon1.fr
Matériaux	**MATERIAUX DE LYON** **http://ed34.universite-lyon.fr** Sec : Marion COMBE Tél:04-72-43-71-70 –Fax : 87.12 Bat. Direction ed.materiaux@insa-lyon.fr	M. Jean-Yves BUFFIERE INSA de Lyon MATEIS Bâtiment Saint Exupéry 7 avenue Jean Capelle 69621 VILLEURBANNE Cedex Tél : 04.72.43 71.70 Fax 04 72 43 85 28 Ed.materiaux@insa-lyon.fr
MEGA	**MECANIQUE,ENERGETIQUE,GENIE CIVIL,ACOUSTIQUE** **http://mega.universite-lyon.fr** Sec : Marion COMBE Tél:04-72-43-71-70 –Fax : 87.12 Bat. Direction mega@insa-lyon.fr	M. Philippe BOISSE INSA de Lyon Laboratoire LAMCOS Bâtiment Jacquard 25 bis avenue Jean Capelle 69621 VILLEURBANNE Cedex Tél : 04.72 .43.71.70 Fax : 04 72 43 72 37 Philippe.boisse@insa-lyon.fr
ScSo	**ScSo*** **http://recherche.univ-lyon2.fr/scso/** Sec : Viviane POLSINELLI Brigitte DUBOIS Insa : J.Y. TOUSSAINT Tél : 04 78 69 72 76 viviane.polsinelli@univ-lyon2.fr	M. Christian MONTES Université Lyon 2 86 rue Pasteur 69365 LYON Cedex 07 Christian.montes@univ-lyon2.fr

*ScSo : Histoire, Géographie, Aménagement, Urbanisme, Archéologie, Science politique, Sociologie, Anthropologie

Acknowledgements

I would like to express my appreciation and gratitude to my main advisor Atilla Baskurt. It was a pleasure and privilege to be able to work with him. His experience about research and his wisdom in general was truly invaluable for me in this journey. I am also immensely thankful to Christian Wolf, who made this thesis possible and guided me through the years, and put me back on track when necessary. I have always felt lucky to have him as an advisor and was inspired from his scientific stance; but most importantly, it was his genuine passion and excitement towards research that motivated me. I would also like to thank Gönen Eren, who is a friend as much as an advisor to me. He was valuable as an enabler and a facilitator, and his insights were priceless. Moreover, he inspired me with his out-of-the-box thinking and his serenity under stressful conditions.

I also want to thank Ozan Çağlayan, my dearest friend. Our discussions regarding research was certainly fruitful and his competency in programming saved me countless times. I would specially like to thank Teoman Naskali, who helped me to put things in perspective many times, and who was remarkably generous to share his personal computational resources in many occasions. Furthermore, I would like to express my sincere gratitude to Eric Lombardi for his technical and scientific support. It would be unfair not mentioning Barış Çelik and Özgün Pınarer; without them my time in Lyon would be extremely dull.

I also would like to sincerely thank the *Consortium* for providing mobility grants for my stays in Lyon, and specially Atilla Baskurt, Jean-Jacques Paul and Christophe Paoli who guided me through it.

Last but not the least, I would like to express my infinite gratitude to my parents, Canan and Binali. This thesis would be impossible without their endless support and compassion. I would like to particularly thank my father -who is now a retired academician himself- for opening my eyes and for encouraging me to leave the corporate world for the greater good: science. Finally, I am truly grateful to my significant other, Bilge, who endured the hardest parts of this adventure with me.

Résumé

Cette thèse s'intéresse à deux problématiques liées et complémentaires, à savoir l'estimation de la posture humaine et la reconnaissance des activités humaines. Il s'agit d'étapes importantes dans de nombreuses applications, tels que les interfaces informatiques humaines, les soins médicaux, la robotique, la surveillance et la sécurité, etc. Malgré les efforts continus dans ce domaine, ces problèmes ne sont toujours pas résolus, en particulier dans des environnements non-coopératifs. L'estimation de la posture et la reconnaissance d'activités posent de nombreux défis, comme les occultations , les variations de points de vues, de morphologies humaines et d'apparences physiques, les fonds complexes, la nature articulée du corps humain et la diversité des comportements des personnes. L'usage de la profondeur permet de gérer les problèmes liés à l'arrière-plan et à l'apparence. En revanche, son application est limitée à des faibles distances entre capteurs et objets d'intérêt. En conséquence, ces types de méthodes sont peu adaptées à des scénarios non coopératifs. Plus précisément, nous avons envisagé des scénarios de reconnaissance d'actions où la position du capteur visuel n'est pas fixée, et qui nécessitent une méthode invariante au point de vue.

Dans la première partie, nous nous sommes concentrés sur la reconnaissance d'actions complexes dans des vidéos. Nous avons exploré plusieurs méthodologies et avons introduit une représentation spatio-temporelle en 3D, qui décrit une séquence vidéo de manière invariante au point de vue. Plus précisément, nous avons caractérisé le mouvement de la personne pour une durée limitée en utilisant un capteur de profondeur et nous l'avons encodé de manière compacte pour représenter l'activité effectuée. Un descripteur de caractéristiques en 3D a ensuite été utilisé pour construire un dictionnaire, qui regroupe des caractéristiques communes. Les activités sont reconnuées à l'aide d'une approche de type "bag-of-words".

Pour la deuxième partie, notre objectif était l'estimation de posture articulée, une étape intermédiaire fréquemment utilisée pour la reconnaissance d'activités. Notre motivation était d'incorporer des informations obtenues à partir de plusieurs des vues, et de les fusionner. Nous avons proposé une extension du modèle de mélange de parties à une gestion de plusieurs vues. Nous avons démontré que les contraintes géométriques et de cohésion d'apparence sont particulièrement efficaces pour renforcer la cohérence entre les points de vue. Par ailleurs, notre approche est capable de gérer les auto-occultations et d'améliorer la robustesse.

Abstract

Estimating human pose and recognizing human activities are important steps in many applications, such as human computer interfaces (HCI), health care, smart conferencing, robotics, security surveillance etc. Despite the ongoing effort in the domain, these tasks remained unsolved in unconstrained and non cooperative environments in particular. Pose estimation and activity recognition face many challenges under these conditions such as occlusion or self occlusion, variations in clothing, background clutter, deformable nature of human body and diversity of human behaviors during activities. Using depth imagery has been a popular solution to address appearance and background related challenges, but it has restricted application area due to its hardware limitations and fails to handle remaining problems.

Specifically, we considered action recognition scenarios where the position of the recording device is not fixed, and consequently require a method which is not affected by the viewpoint. As a second problem, we tackled the human pose estimation task in particular settings where multiple visual sensors are available and allowed to collaborate. In this thesis, we addressed these two related problems separately.

In the first part, we focused on indoor action recognition from videos and we consider complex activities. To this end, we explored several methodologies and eventually introduced a 3D spatio-temporal representation for a video sequence that is viewpoint independent. More specifically, we captured the movement of the person over time using depth sensor and we encoded it in 3D to represent the performed action with a single structure. A 3D feature descriptor was employed afterwards to build a codebook and classify the actions with the bag-of-words approach.

As for the second part, we concentrated on articulated pose estimation, which is often an intermediate step for activity recognition. Our motivation was to incorporate information from multiple sources and views and fuse them early in the pipeline to overcome the problem of self-occlusion, and eventually obtain robust estimations. To achieve this, we proposed a multi-view flexible mixture of parts model inspired by the classical pictorial structures methodology. In addition to the single-view appearance of the human body and its kinematic priors, we demonstrated that geometrical constraints and appearance-consistency parameters are effective for boosting the coherence between the viewpoints in a multi-view setting.

Both methods that we proposed was evaluated on public benchmarks and showed that the use of view-independent representations and integrating information from multiple viewpoints improves the performance of action recognition and pose estimation tasks, respectively.

Keywords Activity recognition, articulated pose estimation, multi-view settings

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter introduces scientific context, topics covered and research challenges addressed in this manuscript. First, context of the research is introduced and application areas are discussed. Then, the main problem that this thesis addresses is presented on Section 1.2 and research goals are introduced. Contribution of this work towards the indicated problems are given in Section 1.3, and finally Section 1.4 describes the outline of the manuscript.

1.1 Context

Figuring out what is happening on an image or a video, or interpreting a scene is a non-trivial task for computers. Understanding an image is a long sought effort and many domains of computer vision such as object recognition, object tracking, activity recognition, pose estimation and others have been popular research topics for many years. These researches have a very large variety of application areas ranging from industrial automation to entertainment, and from health care to security surveillance. In this thesis, we focus on activity recognition and articulated pose estimation of people in presence of multiple cameras.

Recognizing human activities in images or videos is an important notion in many applications, such as human-computer interfaces, health care and dementia care[1], smart conferencing, robotics, security surveillance, advertisement and many more. Figure 1.1 depicts examples from activity two datasets that consist of several actions performed in a realistic kitchen environment and university environment.

Articulated human pose estimation is the task of predicting positions of limbs and other parts of a person given visual data, with respect to a reference system that is based on image or on real world. Estimating the pose is a building block in many vision tasks such as tracking, activity recognition and video indexing, and 2D pose estimation can be an input for 3D pose estimation task. Furthermore, it is more or less directly employed on industrial applications such as human-computer interaction and entertainment, tracking based security systems and motion capture systems as illustrated on Fig. 1.2.

The availability of cameras nowadays is remarkably higher compared to past. This *abundance* of

[1]Seventh framework programme by EU, Dementia Ambient Care: Multi-Sensing Monitoring for Intelligent Remote Management and Decision Support - `http://www.demcare.eu/`, last accessed on 20/03/2017

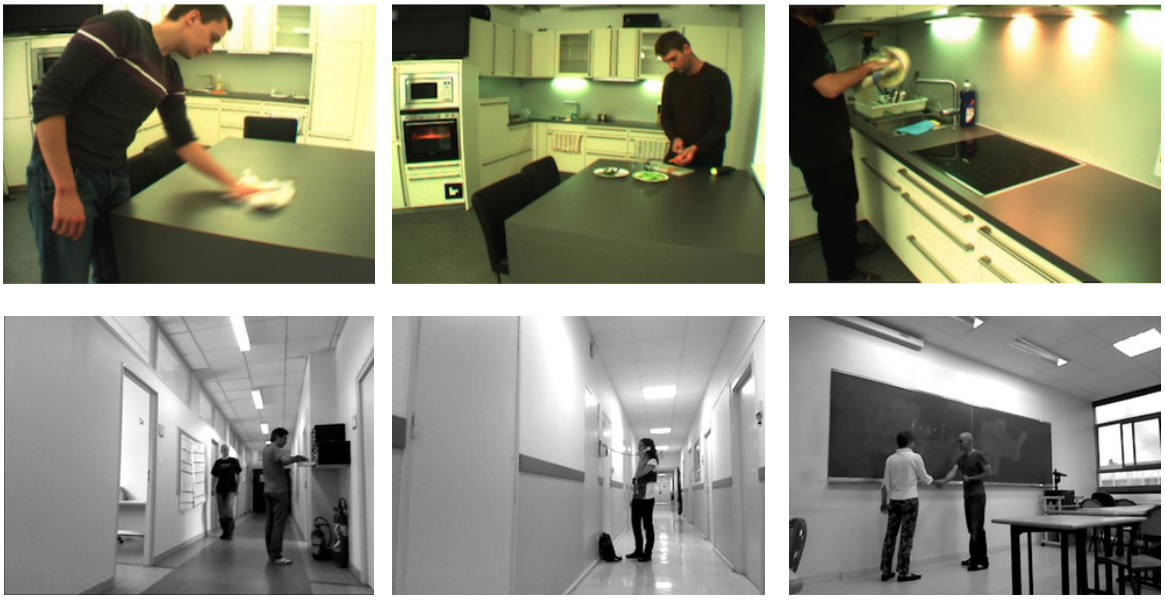


Fig. 1.1 *Top:* Sample images from The KIT Robo-Kitchen Activity Data Set [234], demonstrating various kitchen related activities in a household environment. *Bottom:* Samples from LIRIS Human Activities Dataset [327] showing everyday activities recorded in an uncontrolled environment.

Fig. 1.2 Examples of commercial markerless motion capture systems from MetaMotion[2](*left*) and OrganicMotion[3](*right*).

visual sensors can often be leveraged, for instance where multiple cameras are installed to observe the scene from different viewpoints. With an appropriate approach, information gathered from multiple sources can be fused, and produce collaboratively more reliable and precise information. To that end, the need for algorithms that can perform regardless of their view angle are exceptionally important.

In the next section, the scope of this research will be defined as well as the tackled problems. Additionally, the motivation of the thesis and research goals are presented.

[2]Company website - `http://www.organicmotion.com/motion-capture/`, last accessed on 21/03/2017
[3]Company website - `http://metamotion.com/`, last accessed on 21/03/2017

1.2 Problem and Objectives

For this thesis, we consider two separate but related problems for challenging and uncommon view angle geometries. Problems are defined in the following, and corresponding research goals are stated thereafter.

First, there are numerous methods for recognizing human activities from video sequences, but they are often tailored for specific scenarios and particular cases as discussed in literature review in Chapter 2, therefore their robustness against viewpoint changes are debatable. It is desirable to establish a technique where the position of the camera carries minimal or no impact on the recognition performance. This can either be achieved by building a classification scheme that can directly handle all types of video inputs; or by formulating an intermediate representation that can be view-invariant by its nature, that is, establishing an operation that takes video with any kind of viewpoint as input and outputs a standardized representation. The former clearly requires very large datasets with a high variance of camera angles and ought to be hard to achieve enough generalization even in that case, while the latter can be achieved with comparably smaller amount of data and a suitable approach for representation. Thus, our first motivation is to propose a representation that is robust against viewpoint variation, albeit uncomplicated for classification.

The second problem involves articulated pose estimation, which can be estimated in 2D or 3D and from RGB or depth data. Whereas the problem has been almost solved for easy instances, such as cooperative settings in close distance and depth data without occlusions, other realistic settings present a notable challenge, nevertheless. In particular, pose estimation from RGB input in non-cooperative settings is an ongoing effort and a very active research topic. We consider scenarios where multiple cameras are available and are able to cooperate, and we aim for a solution for these particular cases. Consequently, our second motivation is to propose a method that can leverage the visual data acquired from different viewpoints to achieve a reliable and accurate pose estimation. Furthermore, such a method can be exploited to its maximum if it is possible to evaluate on the fly the adequacy of available viewpoints to support the consistency for a part. We are additionally motivated to explore such a technique that will grant us control over the part based contributions of each viewpoint.

These two problems can be consolidated under the broader notion of *geometrical flexibility for computer vision tasks*.

1.3 Contributions

In the previous section, targeted problems and research objectives of this thesis were stated. In the following, main contributions and achievements will be briefly summarized with respect to the aforementioned problems.

View invariant representation for activities: In order to address the first problem mentioned in the previous section, we aim for a method which is capable of recognizing actions in a view invariant manner. Our goal is to establish a video representation that is able to encode what is happening on the scene, regardless of the placement of the camera. To that end, we exploit depth sensors which provide 3D

information about the scene. Discarding any appearance information, we focus on motion, which by definition conveys crucial information about human activities. The premise here, is that human activities can be recognized solely based on the motion information. Concerning the viewpoint invariance, we argue that actions often possess a dominant direction, and accordingly the acquired 3D motion information can be transformed into a standardized reference system, or a canonical orientation. We also advocate that this approach generally yields an analogous output for similar scenes, regardless of the view angle of the observer. Once the 3D motion information conforms to this recognized representation, it can be leveraged to extract various features, particularly ones that are appropriate for 3D data and can be eventually utilized to label the initial input with an action type.

To accomplish this goal, we first examine various ways to represent a video sequence and present the related literature in Chapter 2. Following the work that focus on motion history, i.e. representations that preserve information about the movement, we propose a volumetric data structure in Section 3.2, which encodes the history of motion in 3D space of the scene. We also propose various improvements and efforts to make this *volume motion templates* more robust. This template is then converted into a canonical representation with a geometric transformation, which grant us the desired view invariance. A particular feature extractor is then employed to determine underlying properties, as described in Section 3.4, and common machine learning techniques are finally used for classification of human activities.

Multi-view model for pose estimation: In response to the second problem that we stated earlier, our goal in the second part of this thesis is to establish an articulated pose estimation method that particularly focus on settings where multiple RGB cameras are allowed to cooperate. This objective compels us to propose an approach that enables to share the acquired information among the viewpoints. First, we identify a method that is naturally compatible for such information transfer. Specifically, we focus on a method that models the scene as an energy function and scores the eligibility of each possible location for body parts, i.e. higher the energy on a given point, higher the likelihood of this point being the body part that we seek. This approach is a common one in pose estimation, either with energy functions or probability maps, and is usually complemented with a body model (e.g. a kinematic tree) which impose a prior on the relative locations of each body part. The solution to the problem is then formulated as an optimization task, and the body part configuration that yields the maximum score is considered the body pose on the scene. Moreover, calculated score for a location can be transferred into other views, for instance via geometrical correspondence.

The advantage of using an energy function is that one can translate any supplementary data into extra score and influence the final pose in favor of this additional information. In our case, we leveraged appearance information of body parts and introduced additional scores to enforce the consistency between the views. In other words, we proposed a method that seeks a configuration of body parts from different viewpoints in the global search space, in which the poses most conform to each other. Furthermore, we point out that some view angles are not suitable to observe some body parts and in consequence may have a negative influence on the overall pose estimation. We introduce an online assessment technique that predicts the *fitness of a viewpoint* for each body part, and control the contribution of each viewpoint accordingly.

These contributions are explained and elaborated in Chapter 3 and Chapter 6, and they are evaluated with experiments in Section 3.5 and Chapter 7, respectively.

As an additional contribution, we implemented a baseline action recognition algorithm that utilize a spatio-temporal feature descriptor, namely histogram of 3D gradients [135]. This implementation, of which the details are given in Section 3.5.2, is used for evaluation of the state-of-the-art methods and assessment of the integrated quality and quantity measurements, which appears in the official publication of the LIRIS Human Activities Dataset [327].

1.4 Organization

The remainder of the manuscript is organized in two main parts that address the two problems mentioned in Section 1.2, and seven underlying chapters. First part, *View independent activity recognition*, concentrates on recognizing human activities in a view invariant manner and starts with Chapter 2 where background on activity recognition is presented along with an analysis of the relevant literature. Chapter 3 introduces our proposition to the recognition problem and Section 3.5 provides experimental results to demonstrate the capabilities of our contribution.

Second part is entitled *Multi-view pose estimation* and its main focus is pose estimation scenarios where multiple cameras are allowed to cooperate. State-of-the-art methods and different approaches are discussed in Chapter 4, which includes detailed summaries about part based models, deep learning methods, solutions for multi-view settings as well as 3D pose estimation specifics. Different approaches were explored beforehand to make a sound decision to follow through for pose estimation; this preliminary research steps and findings are explained and detailed in Chapter 5. Our proposition and pursuit to achieve a multi-view pose estimation technique is introduced and thoroughly detailed in Chapter 6. It is followed by extensive series of experiments in Chapter 7, where introduced method is evaluated and its capabilities are challenged in terms of performance.

Finally both parts are recapitulated and summarized, and the conclusion is given in Chapter 8.

View Independent Activity Recognition

Chapter 2

Background on Activity Recognition

2.1 Introduction

Action recognition, or activity recognition, is the name of the task whose goal is to determine the action or the activity of an individual from an observation. Recognizing human activities is an important step in many applications, such as human-computer interfaces (HCI), health care, smart conferencing, robotics, security surveillance and many more. In this part of the thesis, we target applications where robustness to changes in viewpoint are especially important, as for instance in settings involving moving cameras like mobile robotics.

This is usually a classification task, where an observation is tested against a model (or a database) of known activity categories to determine which activity (if any) is performed during the observation. There are various ways to carry out such task, with various modes of observing the scene. For instance, using a group of sensors is a widely used approach [11, 50, 139], where mobile devices such as smart phones or other wearable items are used to acquire data from various sensors such as accelerometer [177], light sensors, audio sensors, temperature sensors, direction sensors, GPS sensors and heart rate sensors. In some cases, sensor data is first interpreted to determine the context to rule out unlikely activities before performing the actual recognition task [133, 221]. Using wearable and mobile devices is a popular approach to determine the human activity and the reader is kindly invited to read this survey [144] for more details. Furthermore, audio cues are started to be used in action recognition in conjunction with visual data [179, 184, 331]. However, in this thesis we will focus on human activity recognition with computer vision techniques.

In vision-based activity recognition the task becomes labeling the image evidence, which is usually an image sequence, with one of the activity labels that are targeted beforehand. In Fig. 2.1, examples from several benchmark datasets can be observed. Each dataset has various action labels and offers different challenges. On the other hand, it is not always required that the image evidence is a complete video; sometimes poses gathered from some frames [23] or in some cases even local features of a single frame [166, 242] can be enough to recognize the action performed.

Commonly, the recognition process can be seen as a combination of two successive challenges: First finding an appropriate representation for the observations, then using these representations for classi-

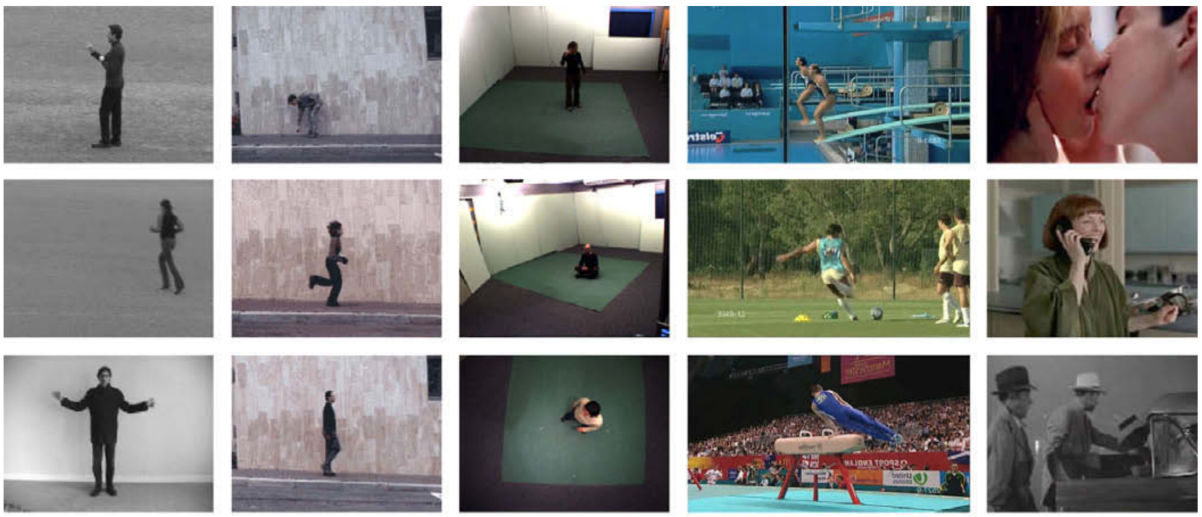


Fig. 2.1 Examples from various datasets: a) KTH [244], b) Weizmann [26], c) Inria XMAS [322], d) UCF sports [225], e) Hollywood human action [142], (Reprinted from [205]).

fication to determine what action is taking place on the monitored scene. The representation of the observation should ideally be invariant to human-related properties such as clothes and body attributes, viewpoint, occlusions and self occlusions and other environmental attributes such as lighting and background. Obtaining an ideal representation would allow to represent a short and heavy man walking on snow seen from distance and a tall slim women walking in an office corridor seen from a close distance in the same way. This arguably unrealistic example would be followed with a proper supervised learning and classification procedure to efficiently label any representation with one of the target activity classes.

Surveys on vision based activity and action recognition propose different taxonomies and different hierarchies between action analysis and recognition approaches [4, 205, 297, 305, 323]. For instance, methods can be first classified based on their *spatial* properties, and then by *temporal* properties, as done by [323]. In this thesis, we will initially focus on two kinds of approaches: Global (or holistic) methods where the action or the actor is represented as a whole; and its counterpart local methods, where a set of local features are considered to represent the activity. Details and examples are given in Section 2.2 for the former, and in Section 2.3 for the latter. Considering the recent popularity of convolutional neural networks and recurrent neural networks in vision tasks, Section 2.4 is dedicated to deep learning methods applied to activity recognition tasks. Unlike the common approaches that are mentioned in the previous paragraph, deep neural networks are known best for their capability the learn distinctive features automatically and therefore learn how to represent the images by themselves. Moreover, since Part 3.5.6 concentrates on this matter, methods that benefits from articulated human poses are particularly important for this thesis and will be reviewed in Section 2.5. Another essential topic for this manuscript is view invariance and settings with multiple viewpoints, thus Section 2.6 will examine related methods and approaches.

2.2 Global representations

An intuitive way to recognize human actions, is to represent the human body and the performed action as a whole; hence the alternative name *holistic representations*. These global models do not need to locate and recognize the types of body parts, but instead consider moving human body as a single object. Temporal analysis and representation of the video sequence is usually established in (i) an ordered manner, as a sequence of features extracted from successive frames; (ii) a holistic manner, as a single space-time structure; or (iii) an unstructured, statistics based manner where occurrences of events are considered. Commonly, global methods require as an input, or as a pre-processing result, a Region Of Interest (ROI); a bounding rectangle for the human body, from which some features are densely extracted to represent the activity (see Fig. 2.2 for an example). An early example of global model is for hand gesture recognition [58], where no features are computed but gesture images are matched with direct correlation.

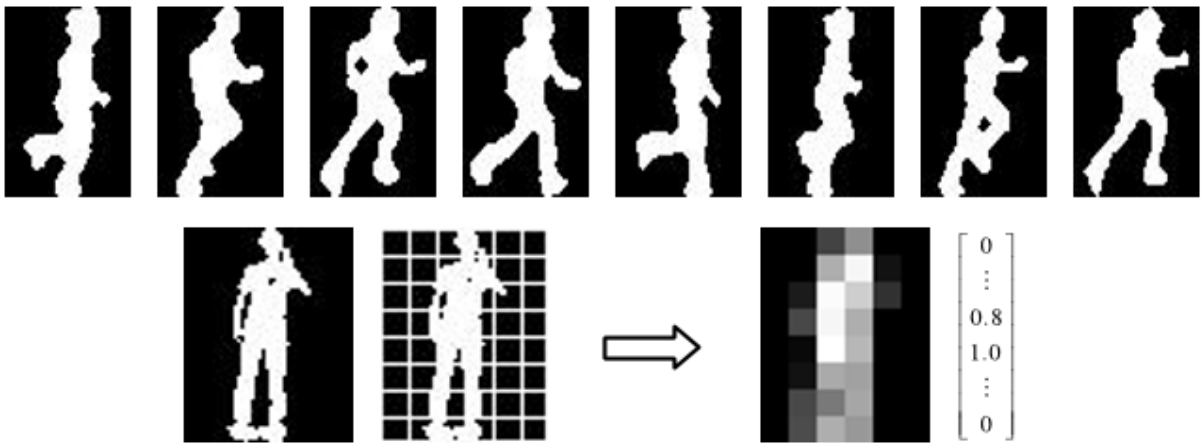


Fig. 2.2 Top: Obtained silhouettes within region of interest for running action. Bottom: Dense sampling of features from the region of interest (Reprinted from [314]).

Global models are generally simplistic in comparison to other types of approaches, which makes them easier to robustly compute. Although frequently they assume simple, uncluttered background as well as region of interest around the body as a starting point. Three main groups can be observed in global methods, for which we will give details in the following paragraphs.

Silhouettes and contours – are actually the outer-shape of the human body, which can be a very discernible attribute for a performed action [328]. Following [338], [314] takes binary silhouette images, quantize them into super-pixels where each pixel is enhanced with ratio of black and white pixels in its neighborhood. This method operates with Hidden Markov Models, whereas [44] uses bag of key silhouette-based poses for recognition. [28] integrates silhouettes over time to form *motion history images* (MHI) with decaying intensities in pixels where the motion is taking place, whereas [1] uses silhouettes to model trajectories on Riemannian shape manifolds. Similarly, [222] accumulates contours over the sequence to form *motion contours*. As well as [322], which takes MHIs to space-time volume objects, [26, 345] work in a similar fashion, directly with a 3D structured object. Another example of

contour-based method is [41], where activities are recognized through shape matching.

Since silhouettes and contours depend on background / foreground segmentation, some work addresses the background subtraction and attempt to improve the quality, for example using *chamfer distance* [71, 320] or shape-context descriptors [164, 272, 354].

When provided, silhouettes and contours are not sensitive to color, texture and contrast changes which makes them a robust source for feature extraction. Furthermore, they usually contain powerful indications about the performed action. However, they can not deal with self-occlusion and they heavily rely on a good background subtraction.

Optical flow features – have been utilized for quite a long time [115]. Basically, they reflect the pixel-wise movement over a given time interval and they have been used for action recognition for decades. Here, we will focus on use of optical flow features on a global scale. Early work uses flow magnitudes computed from a grid of non-overlapping bins [204], and *motion blobs* [52] which are clusters of optical flow fields and their motion, size and position information serves as features for the recognition task. [66] splits the optical flow field into its four components: Positive / negative and horizontal / vertical. These scalars fields of the optical flow vector are then separately matched, and was also utilized in [317]. [75, 132, 143] extend the famous Viola-Jones face detector to action recognition. Rectangular image features are replaced with spatio-temporal cubes which are computed over optical flow. [295], on the other hand, combines silhouette and optical flow features and achieve superior results.

Unlike silhouettes or contours, computing optical flow features does not require segmentation of human body, which liberates this family of methods from background subtraction issues that are mentioned before. However, the motion is calculated from the change of the pixel intensities over time, which ignores other possibilities that can cause differences in the image, such as light and contrast change between the frames.

Gradient features – are commonly used in human detection and often in pose estimation. Nonetheless, there are examples that employ gradient features in action recognition. For instance in [349], gradient fields are calculated in spatio-temporal space, and frames are represented with histograms of those gradients. These histograms are then used for action classification. In [290], an extension of histograms of oriented gradients (HOG) [55] are used with an emphasis on foreground edges by non-negative matrix factorization. HOG descriptors suffer from the curse of dimensionality in some cases, [162] employs principal component analysis (PCA) to reduce the number of dimensions used in HOG feature vectors. There are some works [142, 143] that propose combinations of gradient and optical flow features to achieve better recognition results.

An obvious advantage of using gradient features is that the independence from background subtraction. Also, unlike optical flow features, these features can detect non-moving parts. In some cases non-moving parts such as head can provide a strong cue about the action performed. In other cases though this might become a disadvantage, for the reason that a background object with strong gradient can be confused with a moving part. Furthermore, gradient features are responsive to material, color, lighting and texture, which makes them prone to false detections.

To summarize, global models for action recognition are generally much more simplistic, especially

compared to pose related methods. As a result, they are computationally less expensive. But they are particularly sensitive to viewpoint, or the view angle, as well as the body size and height of the human subject. To address this, one should instantiate a large number of different templates, or design appropriate features and matching techniques that can deal with such transformations. Moreover, global methods usually do not have a natural mechanism to address partial occlusions and self-occlusions. Finally, many of the aforementioned methods make the assumption of a given ROI centered to the person, with simple and uncluttered background. As a consequence, the advancement of global methods are firmly coupled with the progress in other computer vision fields, for instance human detection and tracking.

2.3 Local representations

An important class of action recognition methods are based on local representations of the actions. In this type of approaches spatial, temporal and spatio-temporal features (similar and akin to ones that we have seen in Section 2.2) are extracted *locally*. This means that extracted features are not linked by any means to image coordinates, body part locations, nor exhibit a relationship between them. Simply put, an action is represented with a number of occurrences of position-agnostic visual patterns. The same statistical approach is sometimes extended to temporal layer, ignoring the chronological order of those occurrences. This is the main difference of local representations compared to global representations where positions of extracted features are significant, or compared to pose related methods where feature extractions are closely related to body part configuration. Noticeably, these representations do not require an explicit body part detection or background subtraction to detect the body whatsoever.

In this family of methods, features can be calculated densely on a whole region [218], or sparsely [61, 140, 245, 325], centered around interest points. In the sparse case, one would need an interest point detector to assess which pixels are more *relevant* than others. An example of spatio-temporal feature computation can be seen in Fig. 2.3. Usually, after calculating the feature vectors, whether it is performed densely or sparsely, image sequence is represented as a Bag-of-Words (BoW) model. This approach requires *codeword*s, i.e. cluster centers of learned feature vectors. In test time, each computed feature vector from the image evidence is assigned to nearest cluster center, i.e. to a codeword, and image sequence is represented as a histogram that counts the occurrences of these codewords. The action recognition task is then reduced to classification of these BoW models, as seen in [244]. It is immediately seen that BoW models are robust against partial and self-occlusions, but they ignore spatial relationships between points which cause the lack of structure in the models.

Among the interest point detectors in 3D where third dimension is frequently the time dimension, Harris3D [140, 141], Cuboids [61], Hessian [325] detectors are the most common ones. Another one is maximally stable extremal regions (MSER) [168], and employed in [171, 172] along with other detectors in a complementary fashion. Popular feature descriptors in static images are scale invariant feature transform (SIFT) [161], HOG [55], gradient location and orientation histogram (GLOH) [170] and speeded up robust features (SURF) [17]. For spatio-temporal features their counterparts HOG3D[135], SIFT3D [245] and Extended SURF [325] or combination with other features such as Cuboids [61], HOG/HOF (histograms of optic flows) [142] and motion boundary histograms (MBH) [56] are the most popular

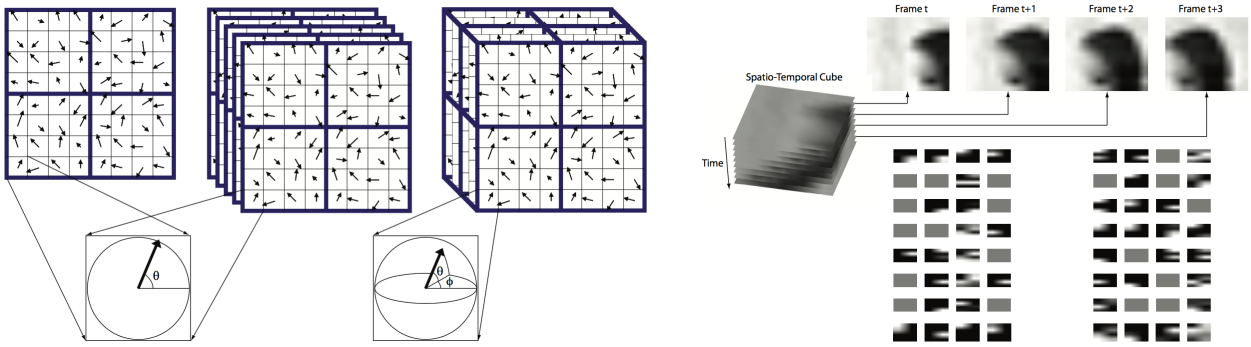


Fig. 2.3 *Left*: SIFT3D feature descriptor, illustrated as an extension of SIFT. *Right*: Computation of SIFT3D and descriptors before and after the reorientation. (Reprinted from [245])

ones. These interest point detectors and spatio-temporal features descriptors are thoroughly compared in [310].

Interest point based approaches do not require the person to be detected explicitly for feature extraction purposes, which is a handy advantage. Instead, it is sufficient that interest points are detected on consistent locations in resembling action sequences. As a drawback, detected interest points are almost always unordered and their size often vary. Therefore, geometric and temporal structures are very difficult to model, which further inspires the wide use of BoW models with local features.

Dense sampling of local action descriptors are utilized in [224], to use in action recognition through Hidden Markov Models. Dense sampling along trajectories have gained interest lately [308, 309]. Using optical flow fields, densely sampled points are tracked in multiple spatial scales to form trajectories. Then, support volumes are constructed around each trajectory, and a combination of HOG/HOF [142] and MBF [56] feature descriptors are employed to describe the motion in video.

In order to address the missing structure information in local representations several efforts are made. Graphical models with hidden variables are proposed to describe the relative positions of the local regions [81, 89, 182, 329], activity localization is encouraged based on spatio-temporal relationships of the interest points [235], or density of interest points were calculated in spatio-temporal context [332]. Similarly, [311] explores the relationship between the interest point by computing contextual densities of interest points. Another approach to cover structural information to local features is to use multiple levels of hierarchy between local spatio-temporal features [137, 281] and other hierarchical representations through pyramid matching [96, 146]. [148] also uses hierarchical features and feature descriptors are not handcrafted, but learned as a unsupervised manner. This conception is very similar to deep learning, and can be seen as a predecessor to modern methods seen in Section 2.4.

Other methods have been introduced to model spatial and spatio-temporal relationships which are ignored by BoW models. Examples include pairwise histograms [235, 287], space-time graph-matching [43, 286], and deformable parts models [291]. These models, originally designed for object detection and recognition, decompose an entity into different parts and learn filters for each part, as well as their geometric configuration: anchor positions with respect to the object center and deformation costs.

Local representation of action sequences is quite invariant to occlusions and robust against intra-class

variations, but BoW models often suffers from lack of discriminative power. To boost recognition capability, various improvements are proposed such as aiming for a vocabulary that is small but discriminative [157], or learning optimal codebooks for BoW construction [95, 122]. [120] applies dimensionality reduction with local spatio-temporal discriminant embedding, creates a mapping in the manifold between silhouettes of the same class, and temporal relations are modeled within the manifold subspace. Opting to use a transformed feature space is proposed in [254], where sparsely calculated features are transformed via Discrete wavelet transform (DWT), then modeled as BoW.

Some work focuses on spatially local features, but takes their temporal ordering into consideration [44, 153, 242]. For instance, [210] employs classical SIFT features, combines with Shape Context features to represent properties between the points and utilize a Semi-Markov model for inference. Tracking is also used for action recognition, where local interest points are tracked in spatio-temporal context to form trajectories and Hidden Markov Models are used for recognition [51]. Similarly, [169] tracks points using the famous Kanade-Lucas-Tomasi tracker and computes the velocity history for all tracked points during the image sequence. Features are also augmented with initial and final absolute position data of tracked points to include spatial information.

There are types of local features that does not require RGB images, but can be computed over a depth image, such as in [312], where *local occupancy patterns* are computed to describe *actionlet*s and an action is represented with an ensemble of actionlets.

In conclusion, local representations are suitable for uncooperated environments and relatively complex scenes where explicit models can be hard to recover. Because, local features and their occurrence based statistics are more flexible against occlusions and cluttered backgrounds as well as intra-class variations such as speed differences and body shape diversity. Yet, the lack of spatial and / or temporal structure affects the discriminative power of these methods even with dense sampling. As a result, local representations of actions tend to perform best in combination with more structural models. The reader is kindly invited to read further details on the recent survey [194].

2.4 Methods based on deep learning

This section is divided into two parts: First, a general definition of deep neural networks is given and they are compared to traditional computer vision methods, following examples from various types of computer vision tasks. Afterwards, use of deep learning methods in activity recognition will be discussed and deep learning based action recognition examples will be reviewed.

Deep neural networks

The traditional pattern recognition tasks rely on hand crafted features and extractors, whether it be a computer vision, speech recognition or natural language processing task. The classical workflow of the traditional approaches usually consists of engineered low level features such as HOG [55], SIFT [161] and HOF [16, 56]; mid-level features such as K-means, sparse coding or bag-of-words for sample representation; and a final classifier or a regressor that is trained in a supervised manner. Some argue

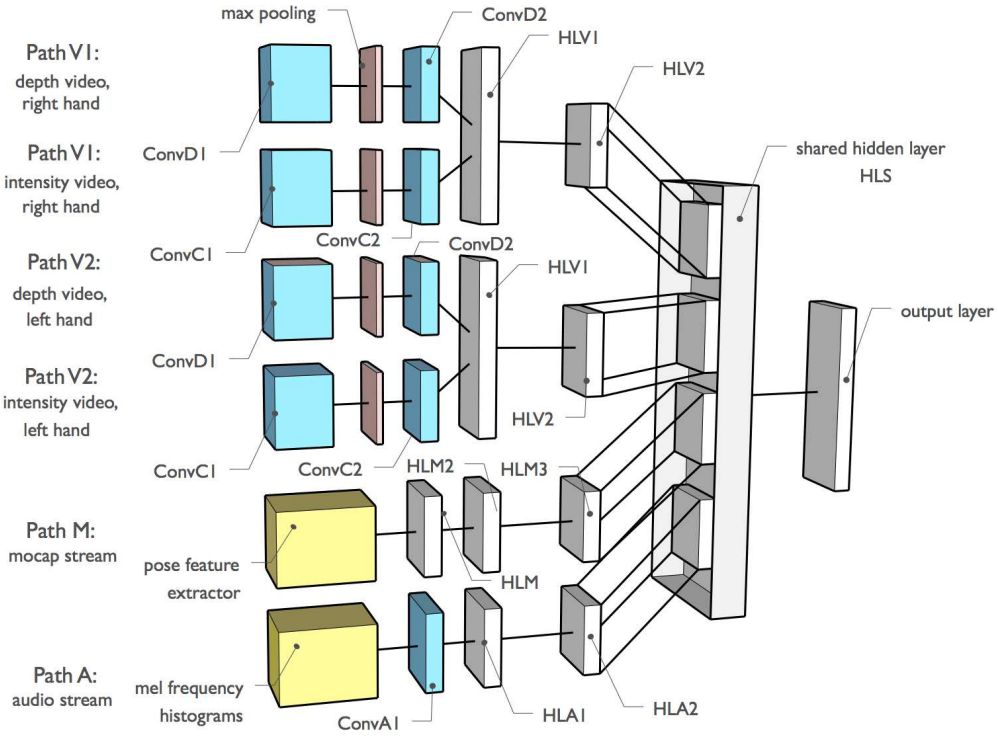


Fig. 2.4 Single-scale deep architecture of *ModDrop*, a deep learning method for multi-modal gesture recognition by Neverova et al. [176].

that almost every computer vision application up until 2012 is actually a "glorified linear classifier" or a "glorified template matching" [151], which are in fact derived from the old notion of perceptron [232].

Deep learning is a newly emerged term and it is often used interchangeably for deep neural networks, recurrent neural networks, convolutional neural networks (CNN), deep Boltzmann machines, so on and so forth. The term gained a remarkable attention since some deep learning based methods [59, 64, 128, 138, 147] proved their efficiency; especially when a large amount of training data and proper hardware, suitable graphical processing units (GPU) are available. The fast learning technique for deep belief nets that is introduced in [106] also played an important role. As of 2016, deep learning dominates computer vision domain, and at high level conferences such as CVPR it is extremely difficult to find papers which do *not* use deep neural networks. It is not exaggerated to state that deep learning has become a standard like linear algebra.

The main innovation behind deep learning is that it does not require explicit feature definitions, but rather captures the properties of the task at hand hierarchically by discovering low-level, mid-level and high-level features by themselves. Another characteristic is the existence of multiple stages of non-linear feature transformations, as seen in the example of Fig. 2.4. For instance, an object recognition task in a deep neural network would usually have several layers where pixels, edges, motifs, parts and finally objects are conceptualized hierarchically. Any neural network architecture that lacks the hierarchy of features is not considered as deep [151].

There are two common types of deep architectures: feed-forward deep neural networks where infor-

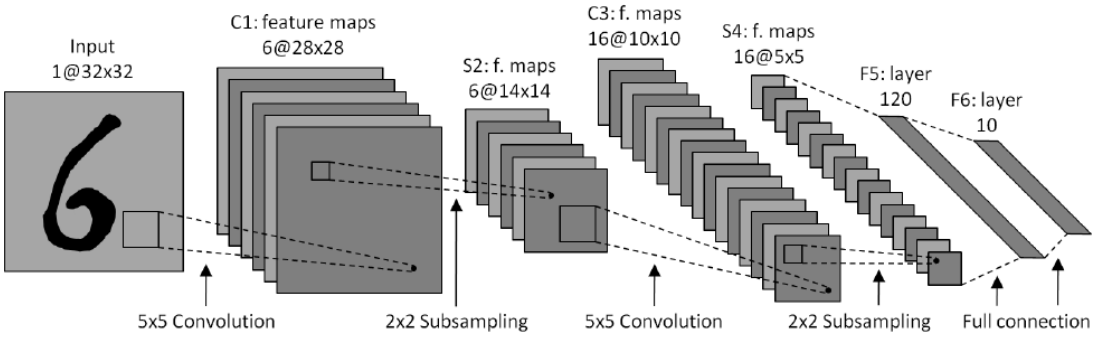


Fig. 2.5 Architecture of *LeNet-5*, one of the first convolutional neural networks introduced by LeCun et al. [150].

mation flow is directed forward only, without any cyclic connections as in Multi-layer Neural Networks, Convolutional Neural Networks (see Fig. 2.5), etc. Computationally, such networks perform calculations on a directed acyclic graph (DAG). Recurrent neural networks are the second family of deep neural network architecture, where recurrent connections are allowed. These directed cycles between the units form an internal state for the network and the network manifests temporal behavior. As a result, recurrent neural networks such as LSTM (*Long short term memory*) are very suitable for sequential tasks such as speech recognition and natural language processing. Furthermore, the training protocols for the deep neural networks can be purely supervised as in most computer vision task; or it can be unsupervised for each layer, with a supervised layer on top or at the end.

To briefly summarize how an object recognition task is carried out with deep learning approach, let us review the workflow of Convolutional Neural Networks. In training phase, an input image is first normalized and then convolved with an arbitrary number of randomly initialized kernels. Each convolution yields a *feature map*, that is the response of the full image to the given filter. Feature maps are then subject a to non-linear transformation, such as ReLUs (*Rectified Linear Unit*) which applies a non-saturating activation function of form $f(x) = max(0,x)$. It should be noted that non-linear functions are vital for expressiveness of the network, otherwise multiple layers of linear functions could have been replaced by a large equivalent single layer that is the linear combination of those. Following the non-linear transformation step, a pooling layer comes into action to aggregate each feature map over space or feature type to eventually subsample the image response to a smaller map. This four step 'normalization - filtering - non-linear transformation - pooling' scheme is repeated depending on the architecture of the network, and each time with a smaller patch to work on. This "shrinkage" of the patches ensures the multi-level and hierarchical feature extraction that is essential to the deep learning standard, since every time the kernel encodes a set of image patches with a different scale. Just before the final layers, there are usually (but not necessarily) a few fully connected layers where the high-level reasoning in between the different types of filters is taking place. At the final stage, a linear classifier is typically employed along with a soft-max function to estimate probabilities for each target class, often used in conjunction with the cross-entropy loss. The error of the output layer is calculated using a loss function given the ground truth. The error is then back-propagated and according to the gradient descent, all the weights are updated proportionally to their contribution to the calculated error. Note that the coefficients

of the kernels that lie in the filtering steps are also considered as weights, therefore they are subject to learning as well. This is considered the key aspect of the deep learning [149], layers of features that are expressed with kernels are not designed by engineers but they are *learned*. The forward & backward scheme is repeated until convergence over the training data that is usually split into mini-batches. Size of the kernels, the amount of stride and other hyper-parameters are usually optimized over a hold-out set. Once the model is ready, it can be used for testing purposes where the test sample is feed to the network and probabilities for output labels are obtained.

Action recognition with deep learning

Automatic learning of hierarchical representations, also known as deep learning, has been successfully applied to numerous problems in recent years, as discussed earlier. Impressive results were obtained for various fields such as image classification [138], object detection [90], video classification [129] and gesture recognition [176].

Unlike object recognition or image classification tasks, in action classification there usually is a time dimension which needs to be considered. That is either established with recurrent neural networks, which are specialized to serial inputs such as natural language processing tasks; or one can feed a convolutional neural network with data that contains temporal information. There are four main modes to integrate information within CNNs: (i) *Single frame*, where all action information is deduced from a single frame; (ii) *Late fusion*, where two frames with an offset is feed to a pair of single frame networks, and fused at the fully connected layer; (iii) *Early fusion*, where input layer accepts a set of frames as a spatio-temporal volume and following layers operates on these volumes; and (iv) *Slow fusion*, which is a balanced version of (ii) and (iii), where a set of frames are processed in parallel and computed features are fused slowly over the layers. Simplified architectures is illustrated in Figure 2.6.

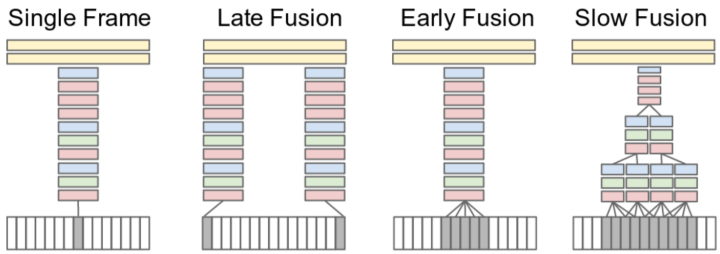


Fig. 2.6 Simplified illustration of integrating temporal information into convolutional neural networks. Red, green, blue and yellow boxes are convolutional, normalization, pooling and fully connected layers, respectively. (Reprinted from [129]).

A natural way to employ deep neural networks is to learn spatio-temporal features instead of engineering them, as proposed in [289] with convolutional neural networks. These features are commonly possess a multi-level hierarchy, unlike the global ones that we have seen in Section 2.2 they are more robust to viewpoint changes, and in contrast to the ones we have seen in Section 2.3 they manifest structural properties. Following this insight, [148] proposes to learn spatio-temporal features utilizing independent subspace analysis, while [13] uses a convolutional neural network (see Fig. 2.7) first to

learn spatio-temporal features, then employs a recurrent neural network to learn the temporal structure of image sequences. A similar scheme is adopted in [14], first a convolutional sparse auto-encoder to learn and extract spatio-temporal features, then a *Long Short-Term Memory Recurrent Neural Network* to benefit from temporal evolution of the learned features for classification purposes. In an analogous manner, end-to-end trainable recurrent convolutional neural networks are utilized to process video frames as well [63]. [344] proposes a new variant of long short-term memory recurrent neural network to label multiple and simultaneous actions in videos. [119] utilized a deep 3D convolutional neural network to extract motion features from a series of image frames, and employed a linear classifier for action classification. On the other hand, [129] demonstrated that a single frame as an input to a deep neural network can perform equally good as multiple frame input.

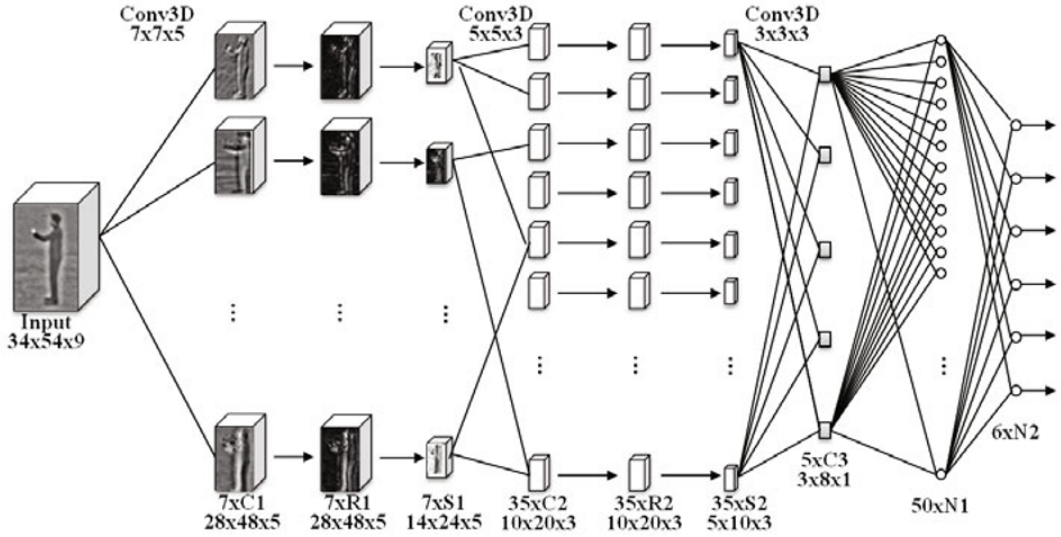


Fig. 2.7 A 3D convolutional neural network architecture to extract spatio-temporal features. (Reprinted from [13]).

Another alternative of providing temporal data to convolutional neural networks is exploiting handcrafted features that contains motion information, for example [268] designed an architecture with two CNNs, and uses raw RGB data for one and optical flow feature map for the other one and learns spatio-temporal features automatically. Later on, this approach was extended with action tubes for the localization of the action [94, 236]. On the other hand, [250] exploits both RGB and depth data and uses a convolutional *deep shared-specific component analysis* network to learn modality-specific features, as well as relations between features of different modalities.

The amount of information contained in videos is very large, considering the time dimension that multiplies the amount of data from each frame. Recent approaches resort to *attention* mechanisms in order to dynamically select relevant portions of the video with a recurrent trainable mapping [15].

All of the aforementioned method are not explicitly designed for view independence. Being heavily data-driven, without a doubt they learn features that are view-invariant to some extent, but it might be insufficient in particular cases. To that end, a deep network for cross-view action recognition is proposed

in [212], where a set of non-linear transformations are learned from multiple source views, to be able to transform new observations to a single canonical view, but the transform model in this work is learned from handcrafted features. On a follow up work, [213] follows a more generative way and creates a view-invariant representation of human pose with deep convolutional neural network model.

Advancement of deep learning methods and growth of the efforts put into this approach is undoubtedly exciting, since the performance of introduced methods are either very successful or very promising at worst case. It would not be unfair to speculate that all of the prominent research teams are already investing in deep learning in terms of both research effort and hardware. Considering the momentum it gained, deep learning approaches seem to continue to be leading paradigm in almost all computer vision tasks, including activity recognition.

2.5 Pose related methods

A natural way of recognizing activities is through articulated pose, i.e. skeletons, stick figures or other kinematic joint models of the human body. Body pose and part locations are particularly informative for performed actions if they can be estimated correctly, as discussed in [342]. In cooperative environments, articulated 2D and 3D pose can be estimated. Especially with depth imaging and using random forests [259], real-time estimations are possible. However, action recognition is often required outside of these cooperative settings.

In this section, the methods that recognize the action based on estimated pose will be reviewed. Spatial structure of actions are described with respect to the human posture. First subgroup of methods work with 2D poses that are extracted from RGB images. Second subgroup uses 3D pose, either directly provided through MoCap or depth data, or calculated and lifted from 2D images. In both cases, the main idea is to track the movement of the joints through time to calculate a set of discernible features. Some methods aim for more complex features, where the spatial neighborhood of the joint or pairwise relation to connected joints is considered as an information source.

Early methods in action recognition based on 2D poses worked with stick figure representations [99, 183]. Other works followed with coarse representations of 2D human body that tracked blobs and patches, for instance head and hand trajectories [35, 217, 278], set of body parts [163] or complete body trajectories [36, 337]. Tracking of body parts are often used in recognition of actions [198, 356]. Unlike the mentioned methods, [116] proposed a bag-of-postures approach, where occurrences of key poses are counted and action sequences are represented as histograms of posture occurrences. [192] estimates pose of two people in 2D, then utilizes those poses to recognize the interaction through hierarchical Bayesian network. Some methods explored extensions and proposed combining displacement-based features with appearance based features, for instance with optical flow and shape features [174]. Appearance model was also exploited in [315], where appearance based features were extracted from a single characteristic keyframe. To avoid explicit pose estimation, which sometimes is even more difficult than the action recognition task itself, [166] introduces an appearance based *poselet activation vector*. This descriptor is an implicit representation of the underlying stick figure, and action labels can directly be inferred from this descriptor.

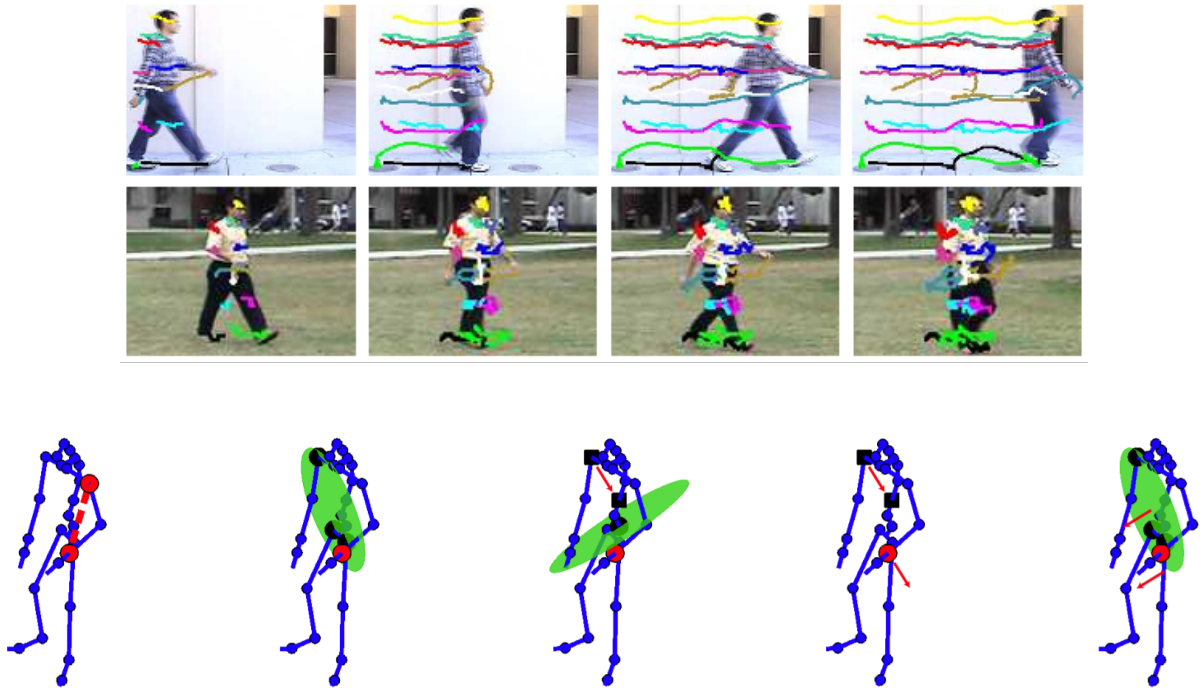


Fig. 2.8 *Top*: Joint trajectories in 2D RGB image sequence. (Reprinted from [346]). *Bottom*: Inter-joint distance, joint to plane distance and other features calculated from 3D kinematic model. (Reprinted from [342])

Most of the methods that work with 3D articulated poses strictly divide the action recognition tasks into successive stages: First one is the 3D estimation of the human posture and the second one is recognizing the action from the movements and displacements of those 3D joints. 3D body models that consist of cylindrical primitives date back to late 1970s [167], and similar 3D representations were exploited in other works [109, 228]. Analogous models are introduced with various improvements, such as flexibility with super-quadrics [88] and textured splines as an appearance cue [97]. Tracking of body parts in 3D provides a view-invariant and distinguishable cues for recognition of activities, as demonstrated with motion capture markers in [39]. Trajectories of significant body parts, such as head and hand, are utilized in [35, 38, 326]. 3D joints are arguably very convenient and promising for action recognition, that 2D observations such as tracked patches are lifted to 3D [117, 216], or disparity maps from stereo cameras are used to project pose estimations to point clouds [102]. Recent work also work with motion trajectories of 3D joints, while the learning is performed on Grassmann manifold [270].

A series of articulated 3D poses extracted from each frame is practical, since it provides motion information of the subject. Furthermore, it can be exploited to compute unary and pairwise features which may serve in recognition of actions. For instance [342] computes joint-joint and joint-plane distances to represent an action. [312] introduces *actionlets*, which are expressed via local depth-appearance based descriptors, called local occupancy patterns. Histogram of Oriented Joint positions are proposed in [333] and is extended with 4D normals in spatio-temporal cells [187]. In [348], 3D pose descriptor is augmented with speed and acceleration of the human body to become *moving pose descriptor*, whereas combinations of three body parts are selected to calculate fundamental ratios in [256]. It should be noted

that it is possible to infer 3D body poses from a series of 2D images as demonstrated in [175], where a combination of features is employed to represent action as a set of action primitives.

To conclude, classifying actions from 2D or 3D pose is very common with various techniques of inference. It is a structured approach, and invariant to view angle in case of 3D. However, it is obvious that pose related action recognition methods heavily depend on the performance of pose estimation. Pose estimation itself is an active field research and is only solved for cooperative settings. There are still many challenges to overcome, despite the relatively mature body of knowledge which we review in Chapter 4. More details and datasets can be found in related survey [65].

2.6 View independence

All robust vision techniques rely on certain invariances in order to work in realistic conditions, which include invariances towards viewpoints, size and morphology of subjects, changes in acquisition conditions etc. Viewpoint independence is particularly important in this context, and various methods have targeted this goal with different approaches. In this section, the taxonomy used to review view-independent action recognition methodologies is inspired from [131], where three families are proposed based on different strategies: We will first review methods with intrinsic invariance, then we will study methods that involve transformation to canonical representations, followed by methods that require exhaustive search.

View invariant approaches focus on features that do not explicitly require a transformation, but grasps various attributes of the action regardless of the recording angle; as well as appropriate matching techniques for such features.

In 2D context, simplest way of acquiring view-invariance against translation and scale variations would be histogram-based methods, arguably. As seen in Section 2.3, these methods do not extract features from a fixed grid of a given coordinate system, but collect features independent from pixel positions and accumulate them to store feature occurrences [349]. Moreover, point correspondences can be established between two observations, either by known camera parameters or matching enough landmarks on human body with SURF [17] or similar detectors. For instance epipolar geometry, which is thoroughly explained in Section 4.4, is often used in action recognition methods [98, 255, 256, 283, 345, 346]. Other methods proposed techniques, apart from epipolar geometry, to determine whether two points are corresponding, such as matrix factorization and rank constraints [217, 246]. [274] introduced a multi-chain structured latent conditional model that formulates the underlying structure of the multi-view image data. Geometrically invariant features that do not change under geometric transformations [190], or three layered silhouette-based features robust against camera rotation [53] are also introduced as view-invariant methods. Another view-invariant proposition is based on relative change between the frames [125], and it was demonstrated that these frame-to-frame changes are reasonably steady against view angle variations.

3D trajectories of parts can be very useful for extracting view-invariant features such as shift invariant velocities in cartesian or polar coordinates, as reviewed in [38]. Actions can be represented in a view-invariant manner with voxel reconstruction and cylindrical 3D histograms [200], with 3D shape-context and spherical harmonics [112], or with Fourier coefficients in cylindrical coordinates [298, 322]. Holte

et al. fuse RGB and depth data from a consumer sensor, compute 3D optical flow features and finally transform them into a view-invariant representations using a spherical coordinate system [111]. In [333], histograms of 3D joint locations are represented in 3D spherical coordinates to assure viewpoint invariance, whereas [154] uses densely extracted Hankelet features that are view-invariant. [158] proposes to use high level features called *bilingual words* and accumulates them to a BoW-like representation. Recently, [211] proposes a spatio-temporal feature that is invariant to viewpoint changes, called *histogram of oriented principal components* (HOPC).

View dependent information are discarded in view-invariant methods, which is good in terms of efficiency compared to exhaustive search on all possible views and also stable under changes in camera angle. On the other hand, removing view dependent information will mostly result in diminishing the discriminative power of the features and approaches.

Unlike the view invariant methods, where the presentation itself or the extracted features are view invariant by nature, some methods require a common pre-processing step, usually to avoid observational differences due to viewpoint or scale etc. Here, the idea is to find a canonical viewpoint, so that the actions are manifesting with similar visual properties. This however, requires first finding transformation from canonical view to current view, then adjusting the current observation with respect to the calculated transformation. If the transformation is estimated correctly, this operation compensates for global variations in body size, scale, translation, and body orientation in case of 3D. Further matching techniques are then executed on this transformed representation.

Based on the knowledge of camera parameters and ground homography, [226] proposes to deduce the 3D orientation of a person using its walking direction in 2D. The silhouette of person is then adjusted to a canonical view frame based on the estimated rotation of the body in 3D, then matched to a set of known canonical silhouettes. Prior based on walking direction to infer 3D rotation is also used in [29, 54, 198, 356], and in [227] this cue is called *dominant motion orientation*. Similarly, [251] proposes to use volumetric intersection of the visual hulls for transformation to canonical orientation. In most cases, the torso is used as a reference to represent relative orientation of body parts, and body parts are corrected with respect to this reference point.

Body orientation is a very important information in activity recognition, as it is in pose estimation and other computer vision tasks. It is easy to calculate if strong cues or a reconstructed 3D body model are available. Main disadvantage of transformation based methods is that the following matching steps heavily depend on the transformed representation and the calculation of canonical orientation. As a result, they are prone to perform poorly if any of these two operation fails.

Apart from finding a transformation as done by aforementioned techniques, or disposing of transformation dependent information as done by view-invariant approaches, a third option to achieve view independency is doing an exhaustive search over all possible transformations. In 2D context, this is commonly carried out in setups with multiple cameras that record simultaneously [5, 28, 185]. As a data driven method, [166] learns multiple patches for each distinctive body part from different view angles and uses them to estimate 3D orientation of a part from a 2D image. [355] builds an infinite number of *virtual views*, then utilizes a virtual view kernel to measure similarities and infer view angle. If a 3D model of the body is available along with the camera parameters however, one can render any 2D view

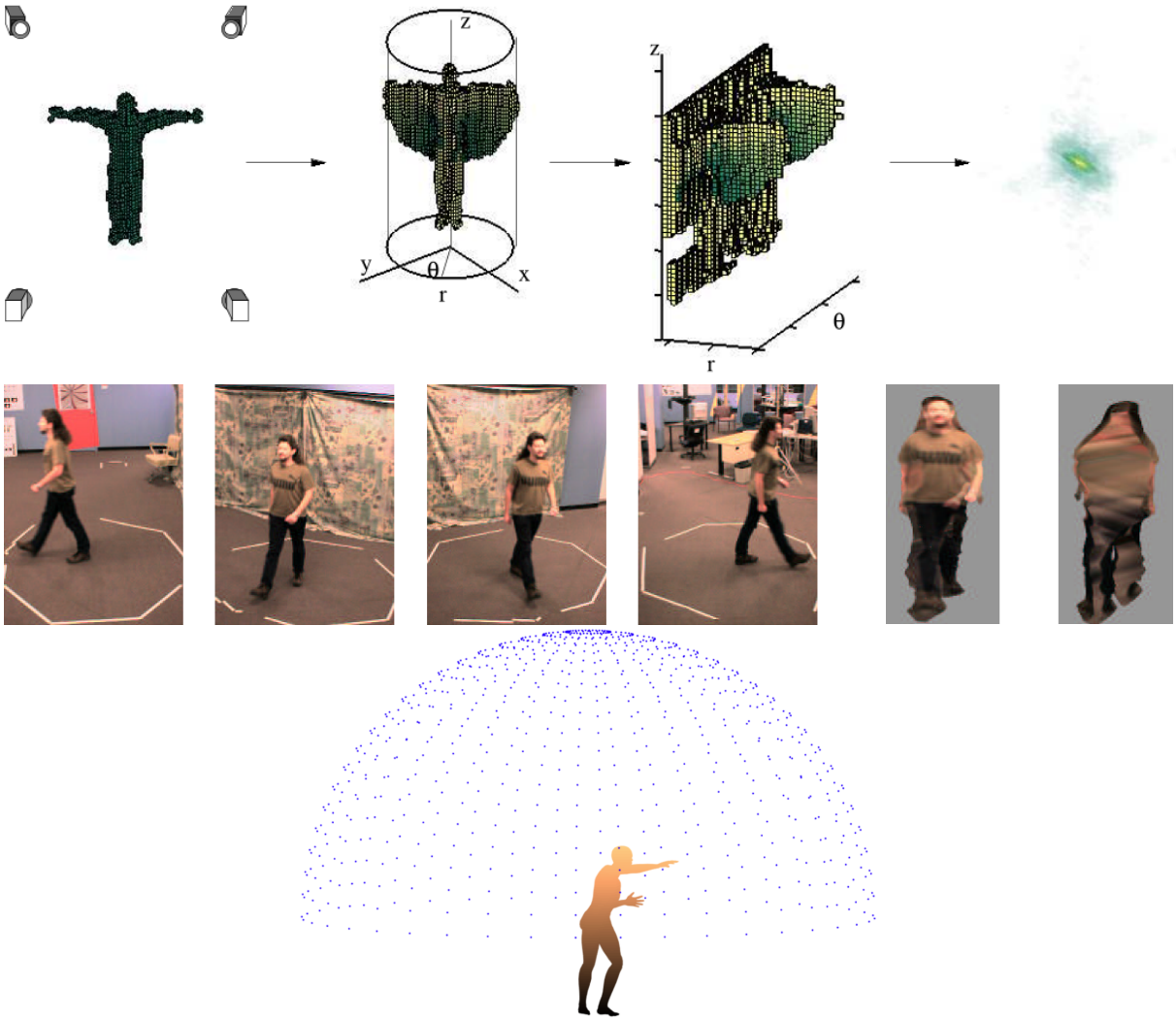


Fig. 2.9 *Top*: Successive visual hulls are accumulated to *motion history volume* objects, then view-invariant features (Fourier coefficients) are extracted in spherical coordinates. (Reprinted from [322]).
Middle: Images from various angles (left) are transformed into canonical representation (right). (Reprinted from [251]).
Bottom: An exhaustive search method, where each dot in the hemisphere corresponds to a generated virtual camera. (Reprinted from [342])

by projecting this 3D model. Many proposition are made in such a generative manner, for instance with synthetic 3D key-poses [126, 164, 174, 321], bag of key-poses [44] and interpolation of key-poses [175]. For gait analysis purposes, 3D poses can be augmented with volumetric primitives and 3D temporal motion models can be fit [301], or 3D kinematic models can be matched to 2D silhouettes [24]. It is also possible to find an analytic function that transforms the 3D representation of the human body to a 2D silhouette [275]. Or inversely, 2D features can be computed first, than they can be back-projected to *4D action shapes* [339]. Further multi-view methods and techniques can be found in related surveys [113, 114]. Recently, [213] proposed a deep-learning approach to use synthetic 3D human models with

a large number of potential camera angles.

Modern computers made the computation efforts required for exhaustive search affordable and acceptable. Methods based on exhaustive search does not require the determination of body orientation, or other transformations which makes them a reliable way to address view independence. Although, without multi-view datasets and strong priors it is almost impossible to accomplish decent recognition of actions using view-invariant features.

2.7 Conclusion

In this chapter, several type of approaches for activity recognition were reviewed. The motivation, the problem at hand and general solutions to this problem, both in computer vision field and otherwise, are briefly given in Section 2.1. First, global representations for actions are studied in Section 2.2 along with several examples. Their advantages and disadvantages were assessed and it was found out that they are prone to occlusions and viewpoint changes, and often require a region of interest detection beforehand. Following, local representations and bag-of-words approaches were examined closely on Section 2.3. Various types of features that are extracted locally were reviewed, as well as classification techniques that build upon those. It was evaluated that this family of methods are robust against cluttered backgrounds and occlusions, however the lack of structure causes the discriminative power of these methods to decrease. Afterwards, deep learning approaches were presented generally with an emphasize on action recognition task in Section 2.4. Despite the short history of these methods, they seem quite successful and promising in action recognition and video labeling tasks. We then moved to methods that exploit pose estimations as an intermediate step to activity recognition in Section 2.5. These methods are well structured and are robust against viewpoints changes if 3D posture is available. On the downside, they strongly depend on pose estimation method and will most likely fail in case of poorly estimated posture. Finally, we reviewed methods that tackle view dependence problem which is particularly interesting for this thesis. Three main strategies namely view normalization, view-invariance and exhaustive search were inspected and example works were provided.

An arguable side observation would be that, activity recognition method groups become mainstream for a time period, then slowly lose their popularity. For instance, in late 1990s global methods were considered most promising, whilst local features were blossomed during the 2000 decade. Pose related methods seem to have two peaks, once after the popularization of Pictorial Structures [78] and once after the Shotton's seminal depth-based pose estimation [258]; which confirms their strong dependency to pose estimation methods. View invariance can be considered as a parallel field of study, where extensions to single-view methods and ideas of view independence constantly proposed. One thing is almost certain though, convolutional neural networks and recurrent neural networks have become the de facto standard for almost all vision tasks after 2012.

The organization of the remainder of this part of the thesis is as follows: In Chapter 3 our proposition for the vision based action recognition task will be presented, first by setting our work into broader context of the state-of-the-art and then by suggesting our mathematical rationale. Then in Section 3.5 we will challenge our method with LIRIS Human Activities dataset, will evaluate results and assess the

success of our proposition, and finally conclude.

Chapter 3

View Independent Activity Recognition

3.1 Introduction and Overview

In this chapter, a framework is proposed to tackle the action recognition problem in a view independent manner. The motivation of this work is to recognize complex activities that are performed by one or more people, in uncooperative environments in a variety from small rooms to long and large hallways, and possibly outdoors. It should be emphasized that we consider scenarios where viewpoint angles are unknown and they are expected to change frequently. For instance, a mobile robot equipped with a camera system is moving in a building while recording videos, would be a meaningful example for our motivation. Without a doubt, such a scenario raises up the necessity to address *ego-motion* issues, i.e. determining the changes in the image due to displacement of the camera and to distinguish these changes from the actual motion that is related to the activity in scene. However, since the compensation for the ego-motion is achieved in literature several times [87, 100, 208, 247, 282], we assume it to be solved and leave it out of the scope of this thesis. Instead, we focus on ways to establish view independence on account of the targeted scenarios, in which the camera can capture the activity from any angle.

The proposed method takes a global approach (see Section 2.2) on temporal domain inspired by [28] and its descendants. In order to constitute view independence, we follow a classical viewpoint normalization scheme to canonical orientation (see Section 2.6). Once we achieve a intermediate representation that is invariant to viewpoint, our pipeline follows a local approach (see Section 2.3) for feature extraction. As a result, it can be said that the proposed method is a combination local and global activity recognition methods with view-independence properties: Temporal data is collected in a holistic manner and has structure in that sense, whereas the feature extraction is carried out locally, making the model robust against occlusions.

The outline of our method is illustrated in Figure 3.1. First, people are detected in the scene and tracked, resulting in a sequence of bounding boxes, or *tracklets* (see Section 3.2.1). Since we also consider activities that may occur between several people, bounding boxes of nearby people are combined to create larger candidate bounding boxes so that they include all actors involved for an activity. Then, a robust variant of volume motion templates (VMT) is computed for this tracklet (see Sections 3.2.2 and 3.2.3). Viewpoint independence is achieved through a rotation with respect to a canonical orienta-

tion, which results in representations where all observations are normalized and equivalent in terms of viewpoint angle. A spatio-temporal 3D descriptor based on histograms of 3D gradients is then densely extracted and pooled into a *Bag-of-Words* model (see Section 3.3). Finally an SVM model is trained from these features, and activities are recognized through classification. The rest of the chapter is organized accordingly.

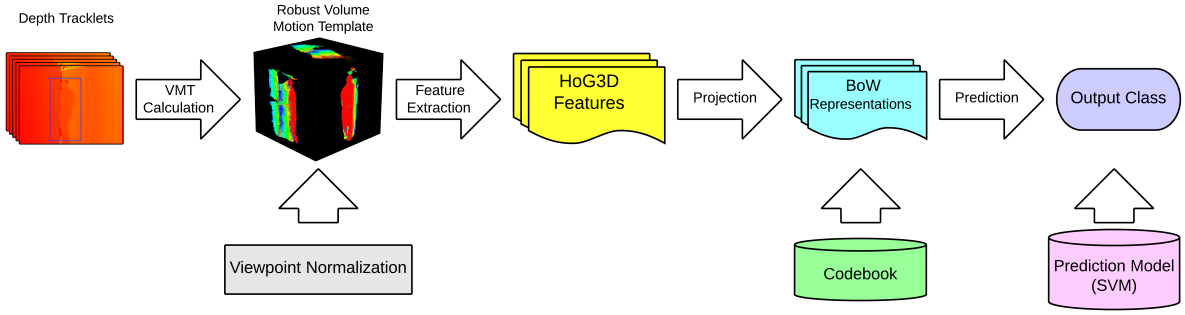


Fig. 3.1 Overview of the proposed method: First, tracklets on depth images are accumulated to compute robust volume motion templates. These templates are then normalized to a canonical orientation, and 3D features are extracted. Using a codebook that is previously learned during training, these features are pooled into a bag-of-words model and the action is finally predicted using SVM classification.

3.2 Robust Volume Motion Templates

In this section, robust volume motion templates are described in details. First, the process to obtain depth tracklets, which are the designated input data for robust volume motion templates, is specified in Section 3.2.1 Then, the formal definition of volume motion template is given in Section 3.2.2. Finally, robust volume motion template is proposed in Section 3.2.3 and the rationale behind it is compared to its ancestor.

3.2.1 Creating tracklets

We create tracklets by employing a human detection method by Ni et al. [181]. People are detected with the Dalal and Triggs detector [55] employing HoG features and linear SVM on grayscale images. Bounding boxes are then transferred to depth images and false positives are filtered using features from the depth image, based on two constraints. The first one is that the ratio of the area to median depth should be within a given range, i.e. $r_l \leq \frac{\text{Area}(x)}{d_m(x)} \leq r_u$, where Area($x$) is the area of a detection x, $d_m(x)$ is the median depth value for that detection and r_l, r_u are the learned lower and upper thresholds, respectively. The second constraint assumes the person to be in the foreground, by comparing the depth values of the detected region against the narrow stripes on both sides of the detection box. Formally, $d_m(x)<d_m(lb_x), d_m(x)<d_m(rb_x)$, where $d_m(lb_x)$ and $d_m(rb_x)$ are the median depth values for the aforementioned striped regions. These constraints are illustrated in Figure 3.2. After the elimination of non-conforming detection candidates, the remaining per-frame detections are matched in consecutive frames with a distance threshold and finally merged into tracklets.

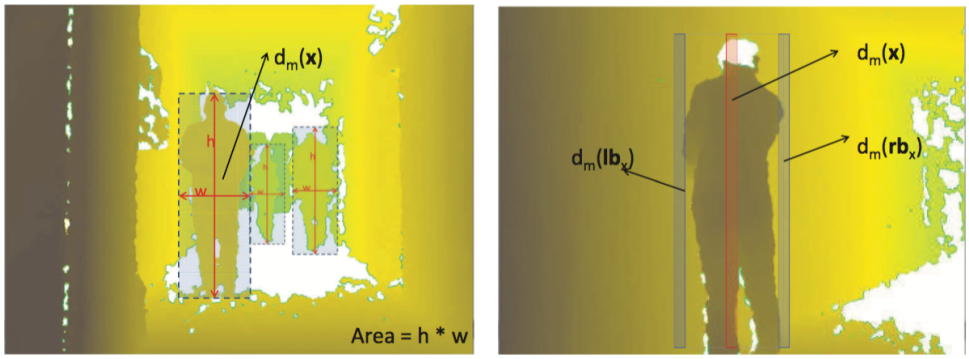


Fig. 3.2 In order to remove false human detections on grayscale images, features from depth frames are used. *Left*, calculation of area and the median depth value $d_m(x)$ of the detection is shown. *Right*, calculation of median depth values for detection as well as side stripes are illustrated. (Reprinted from [181]).

3.2.2 Volume Motion Templates

Volume motion templates (VMT) [227] are an extension of motion history images (MHI) [28] to depth videos (originally obtained from disparity maps of a stereo RGB cameras) whose goal is to describe the motion history of a scene. In a 3D cube calculated for a given time window, recent movement is represented with higher intensity voxels, while intensity of earlier movement decays and finally disappears. Consequently, the motion history of the observation is encoded in a 3D fashion using voxel intensities, i.e. fading traces of moving objects along the movement trajectory as illustrated in Fig. 3.3b.

A VMT is computed for a given time window $[t, t+w]$ where $w+1$ is the number of frames, capturing the motion information for that time window. From a depth image Z_t, a human silhouette is extracted by any form of background subtraction giving a binary image S_t, which is a binary 2D matrix that describes the actor on scene. Then a binary volume object O_t is calculated in 3D space for every frame t in the window as follows:

$$O_t(x,y,z) = \begin{cases} 1 & \text{if } S_t(x,y) = 1 \text{ and } Z_t(x,y) = z \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

Thus, the binary volume object O_t can be seen as the back-projection of the silhouette to 3D space, according to depth values Z_t at hand, which signifies the position of the person with binary voxels in 3D space for the time instance t. Repeating the computation for each frame in the time window, $w+1$ binary volume objects are obtained.

This sequence of binary volume objects expresses the 3D position of the silhouette of the person through time. In order to describe the motion, these simultaneous binary volume objects should be observed chronologically. To that end, consecutive binary volume objects are subtracted one by one:

$$\sigma_t(x,y,z) = |O_t(x,y,z) - O_{t-1}(x,y,z)| \tag{3.2}$$

which results in a series of volume object differences from σ_{t+1} to σ_{t+w}, where each one specifies the inter-frame motion information.

A VMT is then constructed by defining the intensity of each voxel as the "recentness of motion" at that position. That is, newly appeared voxels are known due to volume object differences σ_t and they are set to maximum intensity I_{max} (for instance 255 in 8 bit images). Voxels with no changes are considered immobile locations, and they are subject to fade away. Again, a VMT is calculated for each t, which accumulates the historical data from the beginning of the sequence, and last VMT of the time window represents the historical information about motion for the complete window. More formally,

$$V_t(x,y,z) = \begin{cases} I_{max} & \text{if } \sigma_t(x,y,z) = 1 \\ \max(0, V_{t-1}(x,y,z) - \eta\mu_t) & \text{otherwise} \end{cases} \tag{3.3}$$

where μ_t is the magnitude of motion at time t, which signifies the amount of the motion given the time instance. η on the other hand is attenuating constant:

$$\mu_t = \iiint \sigma_t(x,y,z)dxdydz \quad , \quad \eta = \frac{I_{max} - 1}{\sum_{t=1}^{T} \sum \mu_t} \tag{3.4}$$

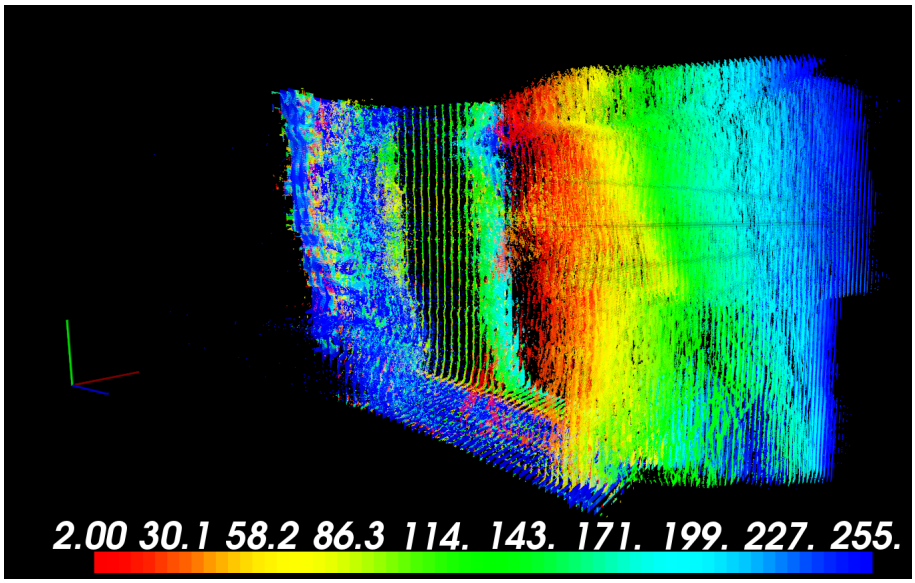
$\eta\mu_t$ can be interpreted as the *disappearing rate*, and it is dynamic for a time window in order to ensure that the VMT captures as much of information from the scene as possible. Without this dynamic mechanism, in cases where there are small amount of motion, voxels that are mobile in the beginning of the time window but immobile shortly after would tend to fade away very quickly; which would result in missing information about a motion on the final VMT object. In other words, disappearing rate is proportional to the total amount of motion during the observation to guarantee the maximum information is encoded in the final VMT representation.

3.2.3 Robust VMT

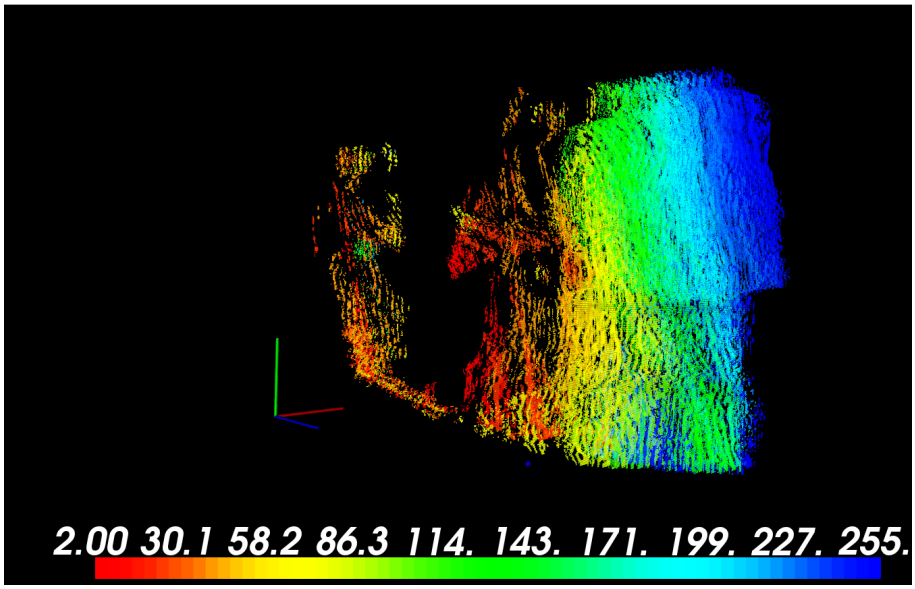
We present a robust variant of volume motion templates in this section, which we call *Robust VMT*. Instead of using stereo camera system and a disparity map, we directly calculate the binary volume object from frames acquired by a depth sensor. The robustness is achieved by three major differences from the original volume motion template. First, we address the noise that is caused by the acquisition device; specifically we address the artifacts produced by the imperfection of the depth sensor which manifest themselves as small but erratic movements between frames. Second, since we do not rely on background subtraction, the double-counting issue emerges where we need to identify the "shadow" of the moving voxel in the background which in fact needs to be ignored when considering motion. Finally, we employ an additional filter to eliminate isolated points that are rather unlikely to be reckoned as relevant for activity. In the following paragraphs, these differences will be explained in details.

We calculate binary objects O_t directly from the depth image Z_t, and since the camera and the background are stationary for most cases, voxels that correspond to background are quickly faded away due to lack of movement. The motion information is then attained without need of background subtraction.

(a) Every 5th frame from the video sequence.



(b) Standard VMT [227].



(c) Proposed robust VMT.

Fig. 3.3 Comparison of standard and robust VMTs. Please not that the VMTs are manually rotated to emphasize the differences in z axis.q (Best viewed in color).

Thus, the binary object is calculated as:

$$O_t(x,y,z) = \begin{cases} 1 & \text{if } Z_t(x,y) = z \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

We introduce a new robust filter that tackles the noise caused by the lack of precision of the depth sensor at hand. Even for scenes where there is no movement at all, there is usually a slight noise in depth images as we observed in our experiments. Put another way, a point in scene may appear on marginally different coordinates within the successive O_t objects. Although it seems like a negligible glitch to the eye, it translates as a strong misinformation onto VMT. Furthermore, the variation appears to be proportional to depth, being more severe in farther points. To that end, we perform a robust difference operation including a local neighborhood search and ignoring insignificant displacements:

$$\sigma_t(x,y,z) = \min_{\substack{x' \in \{x \pm \Delta_x\} \\ y' \in \{y \pm \Delta_y\} \\ z' \in \{z \pm \Delta_z(z)\}}} |O_t(x,y,z) - O_{t-1}(x',y',z')| \tag{3.6}$$

where Δ_x, Δ_y and $\Delta_z(z)$ are parameters that define the size of the search space. Δ_x and Δ_y are fixed to form small neighborhood regions, whereas $\Delta_z(z)$ is adaptive and depends on z. We set it as a monotonically increasing function whose values have been defined from estimations of local depth variances at specific intervals of absolute depth.

As stated earlier, our method does not rely on background estimation and this makes it less dependent on any noise from this error prone background estimation process. However, when subtracting successive volume objects O_t, classically done by Eq. (3.2), differences in depth now directly translate into detected motion without being masked by the background subtraction process. In particular, an object moving before background will translate into two different pixels in motion in the binary motion object σ_t for given coordinates (x,y): a double-counting issue due to an appearance in foreground at one pixel and a disappearance in background for the neighboring pixel. The variation in background is not the actual motion and therefore must be eliminated from the differences σ_t. We therefore change the difference process in order to eliminate the change in background. Formally,

$$\sigma'_t(x,y,z) = \begin{cases} \sigma_t(x,y,z) & \text{if } \nexists\, z' < z : \sigma_t(x,y,z') > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.7}$$

where the difference σ_t is defined as in (3.6). So to put it simply, for a given 2D coordinate (x,y) all non-zero z coordinates are checked, and only the closest one to the depth sensor is kept while others are discarded.

As our VMTs are calculated on tracklets of varying window sizes and not on full frames, we modify the calculation of the magnitude of motion so that it is compatible and suitable for spatio-temporal

volumes of different sizes. Specifically,

$$\mu_t = \frac{1}{V} \iiint \sigma_t(x,y,z)dxdydz \tag{3.8}$$

where V is space-time volume of the difference object σ_t. Additionally, each robust VMT is further normalized by scaling along the z axis into $[0, 2000]$.

To eliminate additional outliers (i.e. isolated points that are very likely to be irrelevant to the observed activity), a *Statistical Outlier Removal* [233] filter is applied to the resulting VMT. This filter computes the mean distance between each point and its corresponding k-nearest neighbors, then assumes that the resulting distribution should be a Gaussian, and finally considers points having mean distances outside a defined interval as outliers. Such points are removed from our VMT to obtain a more clean and compact representation of the activity.

Figure 3.3 shows an example of standard VMT (top) and the proposed robust version (bottom) of a walking person. Please note that high intensity points which signifies recent motion are colored to the blue end of the spectrum, while lowest intensity points are shown in red. The standard version produces motion in static background areas which is extremely noisy, but these artifacts are almost completely avoided by our robust version.

3.3 View Invariance

Let us recall our motivation about achieving a viewpoint independent representation of the activity. As previously seen in Section 2.6 one of the simplest way to achieve this, is to transform the obtained representation to a canonical view, so that all representations and features extracted from them are comparable. In earlier work as stated in the cited section, the walking direction is used as a reference vector for a canonical view. The authors of [227] are following a similar approach and they consider the *dominant motion orientation* for a time window, which signifies the average direction and magnitude of the all movement vectors present in the scene. To that end, *moment vectors* of first and last volume objects i.e. O_t and O_{t+w} of the time windows are calculated as follows:

$$m(O_t) = \frac{1}{\sum_x \sum_y \sum_z O_t(x,y,z)} \sum_x \sum_y \sum_z O_t(x,y,z)\overrightarrow{(x,y,z)} \tag{3.9}$$

where $\overrightarrow{(x,y,z)}$ simply indicates that the result is a 3D vector. And the dominant motion vector is defined as:

$$\vec{\delta}_t = m(O_{t+w}(x,y,z)) - m(O_t(x,y,z)) \tag{3.10}$$

Given the dominant motion vector $\vec{\delta}_t$, and its projection to x, y and z axes $\Delta_t\vec{(x)}$, $\Delta_t\vec{(y)}$ and $\Delta_t\vec{(z)}$ respectively, the corrective angles are calculated as given below:

$$\alpha = \cos^{-1}\left(-\operatorname{sgn}(y)\operatorname{sgn}(z) \frac{||\vec{\Delta}_t(y)||}{||\vec{\Delta}_t(y) - \vec{\Delta}_t(z)||} \right) \tag{3.11}$$

$$\beta = \frac{2}{\pi} - \cos^{-1}\left(-\text{sgn}(z)\text{sgn}(x)\frac{||\vec{\Delta}_t(z)||}{||\vec{\Delta}_t(z) - \vec{\Delta}_t(x)||}\right) \tag{3.12}$$

$$\gamma = \cos^{-1}\left(-\text{sgn}(x)\text{sgn}(y)\frac{||\vec{\Delta}_t(x)||}{||\vec{\Delta}_t(x) - \vec{\Delta}_t(y)||}\right) \tag{3.13}$$

where *sgn()* is a function that indicates the sign of the input.

In [227] obtained 3D volume object is rotated according to α, β and γ, then the obtained object is projected into a 2D representation, as called *Projected motion template* (PMT). We argue that this projection operation may cause loss of information to some extent and avoid the loss by keeping the motion information in 3D space. Instead, we rotate the VMT w.r.t. canonical orientation using the basic rotation matrices (see Equations 3.14, 3.15 and 3.16), resulting in a viewpoint invariant 3D representation from which 3D features can be extracted. The type of the features we employed in our work and their usage for activity classification is explained in the next section.

Figure 3.4 illustrates the results of transformation to canonical orientation in several examples. Left, middle and right columns illustrate grayscale sample from video sequence, computed VMT object and transformed VMT w.r.t. computed canonical orientation, respectively. For the samples that involve walking (e.g. first four rows) it is clear that all VMT objects are rotated so that the walking action is performed from right to left (blue pixels signify most recent movement, while reds signify the oldest). Therefore, all walking motion is encoded in an analogous manner regardless of the view angle. Remaining samples on the last two rows are shown to provide examples with less obvious cases, where the performed action is more complicated and less linear. Nevertheless, the transformation to w.r.t. canonical orientation provides a standardized reference frame even for these cases.

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \tag{3.14}$$

$$R_y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & -\sin(\beta) \\ 0 & 1 & 0 \\ \sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \tag{3.15}$$

$$R_z(\gamma) = \begin{bmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.16}$$

3.4 Feature Extraction and Classification

In this section the activity recognition task is finalized with extraction of features from the spatio-temporal video sequence representation at hand, and the use of these features for the classification task.

There are several feature descriptor options in literature, as previously seen in Section 2.3, that are

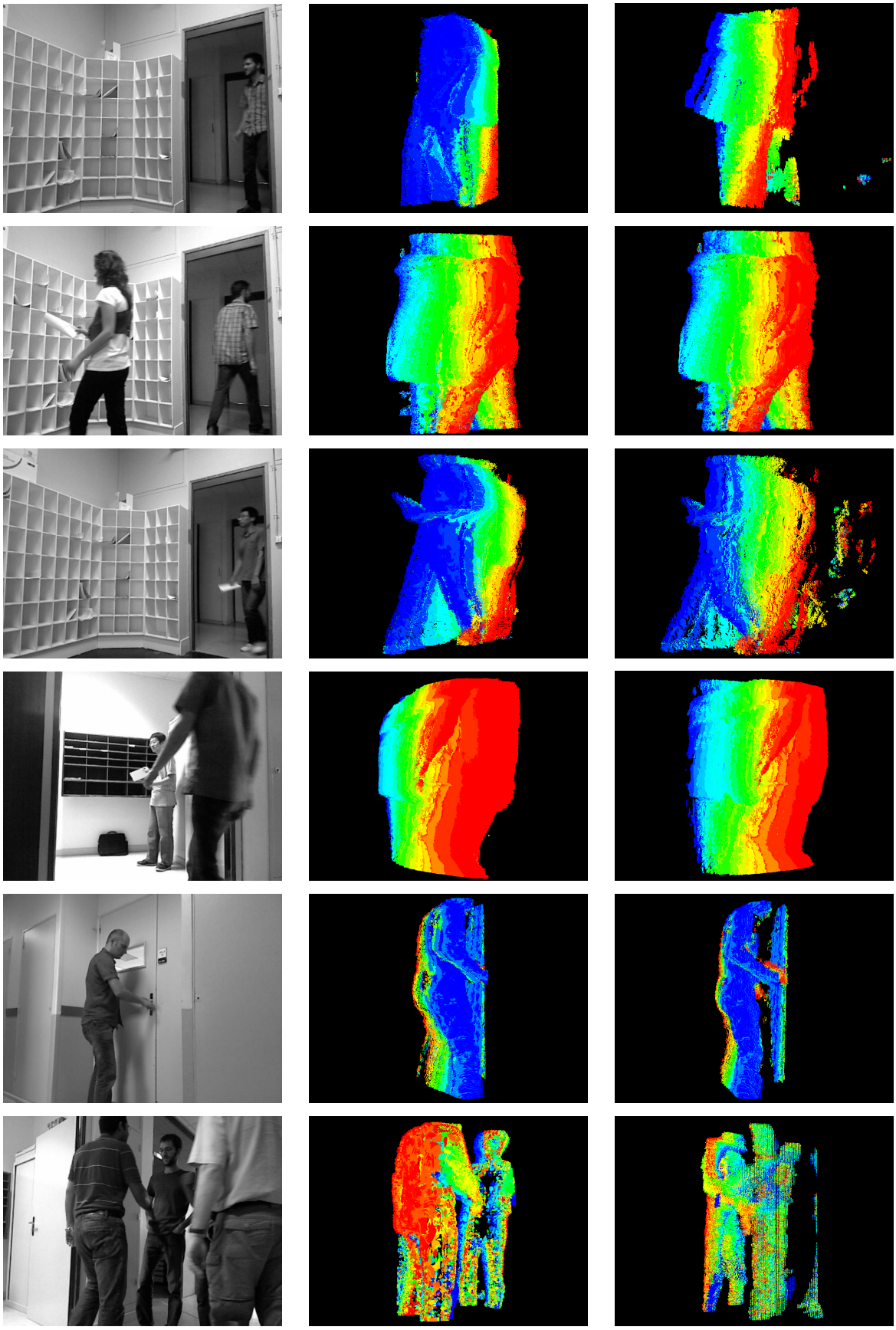


Fig. 3.4 Examples that illustrate the transformation w.r.t. canonical orientation. *Left column:* Sample frame from video sequence. *Middle column:* Computed VMT. *Right column:* Robust VMT which is rotated according to dominant motion vector. Blue pixels signifiy most recent motion, while reds signify the oldest. (Best viewed in color.)

designated to use on 3D data. Most of these methods are executed on raw image data which are usually a spatio-temporal stack of consecutive RGB images, or on integral videos which are frequently used for decreasing the computational cost. Therefore, there are components of these descriptors that encodes motion information via optical flow to describe motion within the observed sequence, either solely motion information or accompanied with appearance information. Please note that our intermediate representation, robust VMT, already contains information about motion in the scene. In particular, the motion information is translated into differences in intensities, which implies that an appearance based feature descriptor is much more suitable for conveying information about the motion. To that end, the HOG3D descriptor appears to be a reliable alternative due to its capabilities to express the differences in intensities as well as the orientations of these differences. Thus, we extract features from VMTs using a method based on HOG3D descriptors.

In Section 3.4.1 the HOG3D descriptor is explained in details, and in Section 3.4.2 the recognition scheme using these features is reported.

3.4.1 HOG3D

HOG3D is a local descriptor which is a generalization of well known HOG descriptor [55] to 3D spatio-temporal volumes, introduced in [135]. Designated input data for HOG3D descriptor is stacked consecutive images taken from a video, or most likely 3D regions that is cut out of such volumes.

Each given point is described with multiple 3D gradient vectors that are computed inside a support region. The computation scheme works on three folds: The support region is divided into *cells*, and cells are divided into *sub-blocks*. In each sub-block, mean 3D gradient vector is calculated. Calculation of mean gradient is actually straightforward, but considering the number of sub-blocks in a support region, the total number of support regions and the necessity to handling different scales with a pyramid scheme, it quickly becomes cumbersome in terms of computation. This gradient vector is then projected on a regular polyhedron with congruent faces, i.e. polyhedrons where every face has a counterpart on the opposite side. Each face of the polyhedron is considered as a bin for the histogram, thus the mean gradient vector is quantized and conveys information about a single sub-block. To roll up, these quantizations for each sub-block are accumulated into a histogram to portray the gradient data of a single cell. Such histograms gathered from all cells within the support region are collected and simply concatenated to one final feature descriptor for the point that was given as an input. Overview for this procedure is illustrated in Figure 3.5. These support regions are determined either by an interest point detector or by dense sampling.

Let us recall that robust VMT contains information about motion and nothing else from the observation. This indicates that only movement is encoded in VMT and allows us to assume that every voxel in our robust VMT is conveying relevant information about the observed activity. Consequently, we calculate the descriptor on robust VMT objects by dense sampling, unlike [135]. This is also the main difference from the approach in [135], the fact that the input data is very different in nature and instead of stacking RGB frames to form an integral video we utilize robust VMTs that contains motion history information in 3D.

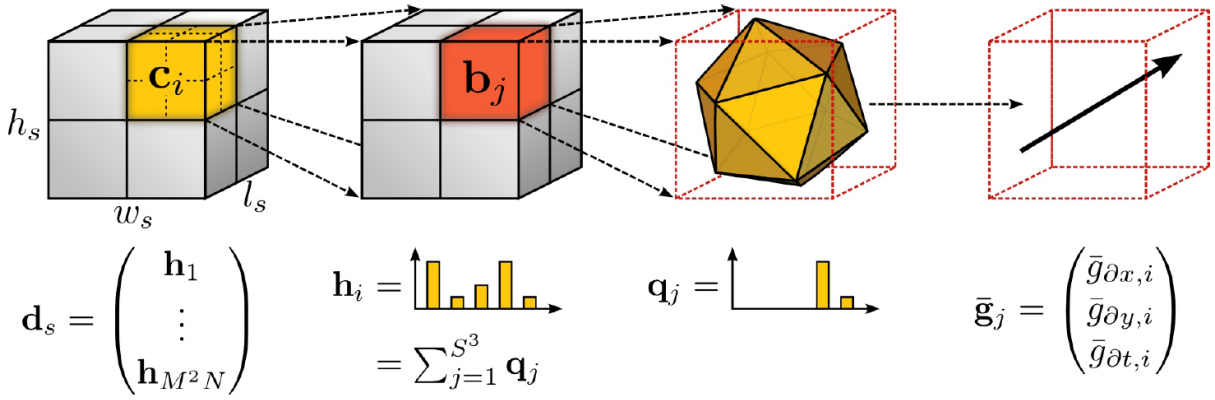


Fig. 3.5 Hierarchical overview of HOG3D feature extraction, see text for details of operations on each level. (Reprinted from [135]).

A note about the way HOG3D descriptors are calculated in this VMT context, is that our robust VMT objects are sparse point clouds in practice, which makes gradient computation difficult because of the gaps between the points. We solve this by trilinear interpolation of gaps, using a *maximum gap length* threshold which makes the decision of whether points of same intensity values should be interpolated or not.

Finally, each set of histograms portrays data about a single point as explained above. Since dense sampling is utilized to determine the points of which features should be extracted, we end up with a large number of feature vectors for a tracklet. The procedure to exploit these features vectors in order to recognize the observed action is described in Section 3.4.2.

3.4.2 Classification via Bag of Words

It is clarified in Section 3.4.1 that HOG3D features are extracted locally and each densely sampled point is described with a set of histograms where each bin is a face of polyhedron that is used for quantization of the mean gradient vector computed within a sub-block. Consequently, a large collection of locally extracted feature vectors are obtained for a single tracklet.

Following the classical Bag-of-words approach, a codebook is built using k-means clustering. Specifically, all high-dimensional feature vectors from training set are considered and clustered in a high-dimensional space. Cluster centers are assigned as codewords of the codebook. Once the codebook is ready, each tracklet is processed and computed feature vectors are assigned to a closest cluster center, i.e. codeword. As a result, the tracklet is tranformed to a single histogram where each bin is a codeword and occurrences of these codewords are counted. Simply put, each tracklet is represented by a *bag* (i.e. histogram) of *words* (i.e. codewords) where it can be observed how many times a codeword appears.

Thus, each tracklet is represented by a Bag-of-words model calculated by projecting densely sampled features on a codebook obtained by clustering. In order to recognize activities in tracklets, a prediction model should be utilized. We employ Support vector machines (SVM) model, which is a supervised learning model. We train a non-linear SVM model, particularly a Gaussian radial basis function with

a kernel of type $K(x,x') = exp(-\gamma||\vec{x} - \vec{x'}||^2)$, where x and x' are samples and γ is a positive hyper-parameter that is learned during validation.

The activities have different lengths in terms of time, even same class of activities can be of variable duration depending on the person who performs it. To that end, we divide tracklets into runs of sliding temporal windows of length $T = 40$ frames and 50% overlap. Temporal windows are classified individually in our recognition scheme, and classification results are integrated over tracklets through majority voting. The label with highest number of votes is then assigned to the tracklet as the final recognition result.

The evaluation of the proposed method is given in Section 3.5, where it is tested on a challenging activity dataset and compared to a well known baseline method.

3.5 Experiments

In this chapter we evaluate the method we proposed in Chapter 3 and conduct experiments to demonstrate the extent of its recognition capabilities. To that end, we selected a challenging dataset, which is recorded in an uncontrolled environment and contains sequences of interacting people. It should also be noted beforehand that this dataset is recorded from the perspective of a mobile robot, which conforms appropriately with our intention to test our method particularly in terms of viewpoint invariance.

Following sections will first thoroughly present the selected dataset and the particular subset that is most suitable to the proposed method. After that, details concerning the training procedure are disclosed, as well as the evaluation specifications and metrics. Then experiment results are stated, along with a comparison to baseline method to indicate the amount of improvement committed by our contribution. Finally, the activity recognition part is concluded with Section 3.5.6.

3.5.1 LIRIS Human Activities Dataset

LIRIS Human Activities Dataset [327] is a publicly available dataset, which was originally created for the Human Activity Recognition and Localization (HARL) 2012 competition of 21st International Conference on Pattern Recognition (ICPR). The dataset consists of video sequences that are recorded by a commercial depth camera (Microsoft Kinect™) which is mounted on a mobile robot, as depicted in Fig. 3.6. It contains not only simultaneously occurring complex and realistic actions, but also people-people and people-object interactions. Data is offered in the standard format for the depth camera, which contains two sequences of 640×480 grayscale images and 640×480 depth images for each sequence. Ground truth annotations are provided along with the dataset, formatted as sequences of bounding box coordinates and class labels. For convenience, calibration parameters between the depth sensor and grayscale sensor of the depth camera are provided, which may prove useful for methods that require spatial correspondence between depth and grayscale images. The creators of the dataset also provide RGB images recorded via a camcorder, but these are not used in our experiments.

Conforming to the operational prerequisites of the employed depth camera, all video sequences are recorded indoor. Specifically, the recordings take place in a real-life university environment (INSA Lyon) which includes difficult settings such as narrow hallways and rooms of various sizes. There are ten classes of action: Discussion between two or more people (`DI`), give an object to another person (`GI`), put / take an object into / from a box / desk (`BO`), enter / leave a room (pass through a door) without unlocking (`EN`), try to enter a room (unsuccessfully) (`ET`), unlock and enter (or leave) a room (`LO`), leave baggage unattended (`UB`), handshake (`HS`), typing on a keyboard (`KB`) and telephone conversation (`TE`). Some of the samples are illustrated in Fig. 3.7. All actions were performed by a group of 21 people of both genders, with varying outfit and personal appearance. In total, dataset offers 305 actions for training and validation, and 156 actions for test purposes. However the distribution of available videos are remarkably unbalanced in terms of classes, as there are 110 samples of `EN` whereas there are only 22 for `UB`.

[1] VOIR - Vision and Observation In Robotics, `http://liris.cnrs.fr/voir/wiki/doku.php`, last accessed on 30/03/2017

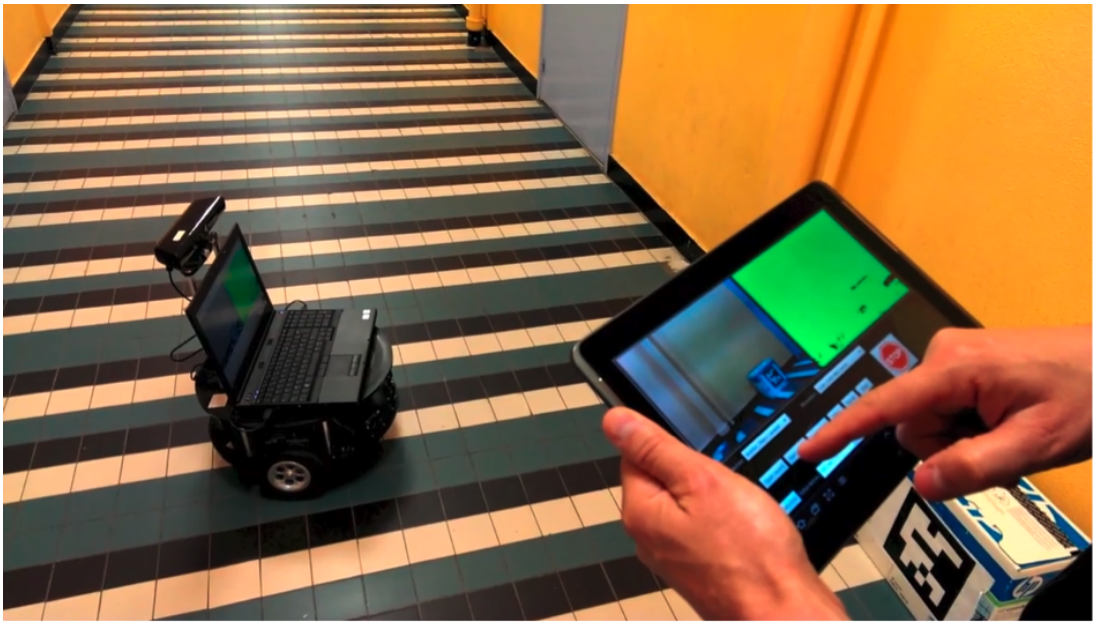


Fig. 3.6 Mobile robot with a mounted commercial depth camera, which is used for recording of the dataset. Part of *VOIR*[1] platform of LIRIS laboratory.

The video sequences were recorded from various viewpoints, and for some of the sequences the mobile robot is actually in motion during an activity. Various viewpoints prevent learning actions from potential background features. The LIRIS Human Activities dataset is considered more challenging compared to other activity recognition datasets due to its notably high intra-class variation. For instance, subjects performing `DI` may be sitting or standing; likewise, `EN` may involve opening the door or not. Moreover, there is a remarkable amount of similarity between some action classes: `LO` and `ET` actions include standing in front of a door for a period of time while trying the key to the lock and thus, are almost identical except for the last part. Similarly, both `BO` and `UB` actions may involve putting down an object and both result in a similar display, but by definition a baggage becomes "unattended" only after a while. As a result, recognizing some actions requires either a long duration of observation, or additional steps as object of context recognition. On the other hand the mounted camera can be considered firmly fixed with respect to the mobile robot, and there is negligible or no tilt / yaw / pitch movement. In other words, camera is at a constant height and always parallel to the floor.

Our method has a limitation that is discussed in Section 8.2, which makes Volume Motion Template (VMT) calculation difficult and inefficient for cases with a moving camera. For the sake of this experiment session, we simply work on a subset of the dataset which conforms with our requirements and we only consider the actions that are fully recorded in a stationary manner, excluding the ones that contains motion of the camera. This shortcoming may be addressed with compensation for the ego-motion, as described in Section 8.3, which will enable VMT calculation for such cases.

Fig. 3.7 Several samples from LIRIS Human Activities dataset, including following activities in each row: Handshake, unlock and enter, discussion, enter/leave room, telephone conversation, put/take an object.

3.5.2 Baseline Implementation for the Dataset

As mentioned earlier, the LIRIS Human Activities dataset was initially created for the HARL competition in 2012, where over 70 teams attended worldwide. The dataset is a challenging one, and the goal is not only classification of activities but also localizing them in time and space. 4 teams accomplished the task at hand and submitted their results. In the publication associated to the database [327] an elaborate evaluation metric is presented, which is also explained in Section 3.5.4. We applied the metric on as many methods as possible to demonstrate the the capabilities of the evaluation metric itself, and to position the participants' methods with respect to known baselines. To that end, two additional baseline methods (dubbed *Method A* and *Method B* in the article) to the original four submissions were also included to observe the introduced metric in action, which consequently provide comparison of results among diversified approaches.

Here we will focus on *Method A*, which we implemented solely for this purpose, and we will omit the details concerning all other methods which can be found in the original publication. *Method A* is based on the spatio-temporal feature descriptor [135] that we described in Section 3.4.1. Basically, the method consists of a pre-processing step where tracklets are obtained as described in [181], a HOG3D feature extraction step applied on depth tracklets, and a classification step where bag-of-words approach is applied to classify the activities. A primitive version of the feature extraction code is available online[2], but it fails to meet the requirements for this set of experiments. Based on this C++ implementation, we almost re-implemented the whole pipeline from scratch and supplemented the program with extra features such as new input formats to support LIRIS Human Activities dataset depth images as well as image conversion steps, new stream readers to transform XML tracklets into *track-file* format required by the feature extractor, experimental gradient computation functions, algorithms to handle dense sampling in space-time sliding windows settings, launcher programs that executes multiple instances of the implementation on multiple cores to reduce the total execution time and countless other helper tools to ease the heavy data load involved in the experiment procedure. All written code, helper scripts and used third party libraries are available on a public repository[3], which also includes the code regarding the VMT calculation and corresponding helper functions.

Once the HOG3D features were extracted from the sliding windows with 50% overlap, we employed k-means clustering to build a codebook to use bag-of-words method and represented video sequences as a collection of *codewords*. These are then used to train a support vector machine prediction model which is used to finally classify test samples. Details regarding the codebook building and classification procedure is given in the next section for VMTs, which was essentially similar for this set of experiments. As a matter of fact, some of the design decisions for the VMT experiments were actually taken based on the hands-on experience and empirical results that we obtained during the experiments of baseline method. Details on these methods (as well as on the HARL 2012 competition) can be found on our journal paper [327].

[2]Tool for computing 3D descriptors in videos: `https://lear.inrialpes.fr/people/klaeser/software_3d_video_descriptor`, last accessed on 24/03/2017.

[3]GitHub repository that contains code for the first part of the manuscript: `https://github.com/emredog/vmt`, last accessed on 24/03/2017.

Fig. 3.8 VMT objects are interpolated to fill the gaps between the voxels, in order to ease the feature extraction step. *Left:* before interpolation, *Right:* after interpolation.

3.5.3 Training

Video sequences consist of various actions, which have diverse lengths even for same activity class and often occur simultaneously. Working with tracklets overcomes this problem and enables us to focus only on the performed action. Furthermore, we divide the tracklets into smaller ones of T frames with 50% overlap, which produces a small number of tracklets for actions that take a small amount of time to complete, and larger number of tracklets for actions that last longer. This may seem like an approach where a single tracklet might be missing some important proportion of the whole activity sequence, but since we employ a *local* feature descriptor, all features accumulate to describe the bigger picture. It should also be noted that in order to be able to detect activities (as opposed to pure classification), we also included a *No-Action* class whose training examples are selected through bootstrapping. In our experiments we set T to 40 frames, which translates to ~1800 training samples for training and validation purposes after balancing.

For each depth tracklet, a robust VMT was calculated as described in Section 3.2.3 where noise, double counting issue and outliers were addressed according to corresponding descriptions. Dominant motion vector and complementary corrective angles were calculated respectively as indicated in Section 3.3. The obtained VMT, which is a 3D point cloud, was rotated in agreement with canonical orientation. Let us recall that the mobile robot of the used dataset translates itself only in XZ plane and allowed to be rotated around its own y axis, therefore it is sufficient to rotate VMTs around y axis for viewpoint normalization. It should also be noted that our robust VMT objects are sparse point clouds, which makes feature extraction, i.e. gradient computation, difficult due to fictitious zero values in between. We solve this by trilinear interpolation of gaps as illustrated in Fig. 3.8, using an arbitrary *maximum gap length* threshold.

We applied the HOG3D feature extractor described in Section 3.4.1 to all obtained robust VMT objects with dense sampling, which enable us to represent each VMT object as a set of 80 dimensional feature vectors, each of which is a collection of histograms. In order to employ a bag-of-words technique for classification task, we first needed to build a *codebook*. A classical k-means [159] implementation were used for clustering the feature vectors with $k = 4000$, and was executed for 100 iterations. Cluster centers were appointed as *codewords* to assemble our codebook. Each feature vector within a VMT ob-

ject was assigned to closest cluster center, or codeword, where we used Euclidean distance as a proximity measure. Eventually, this allowed us to describe a VMT object as a 'collection of codewords', hence the term bag-of-words.

For the classification task, we trained a support vector machine (SVM) prediction model with a radial basis function kernel. We employed a publicly available variant called *nu-SVM* [45, 243] and learned its hyper-parameters via 10-fold cross-validation as follows: $\nu = 0.18$ and $\gamma = 0.008$.

3.5.4 Evaluation

In our experiments, we employed a classical confusion matrix which is calculated according to the constraints described in the official publication of the dataset [327]. We evaluate our method primarily based on this indicator, since confusion matrices provide a rapid feedback, and the amount of correct detections can be seen along with false positives and false negatives. For actions that span multiple tracklets of T frames, which is the case for most samples, our algorithm will produce multiple labels. In those cases a simple majority vote was performed to identify the performed activity.

Additionally, as a performance metric we refer to the official metric of the dataset which is described originally in [327] in details. The objective of this metric is to measure the similarity between the ground truth action labels and the detected action labels while taking the spatial and temporal accuracy into account. Naturally, the defined metric should penalize two types error: Missing detections, i.e. deficient detections which temporally or spatially overlook an action; and excessive detections which label locations or time frames that do not contain any action. The metric is intended to address both quantitative and qualitative evaluation of any tested method; it should be capable to tell how many actions have been detected correctly as well as false positives, and how good is the detection quality.

To fulfill these objectives, two spatial and two temporal thresholds are introduced. This thresholds directly controls the quality of a detection, such that a detection is considered a match, only if it meets the quality requirements imposed by these thresholds. This is complemented with plots, where scores are shown as a function of these thresholds. Combined, these two adequately illustrate the dependence of quantity on quality. In the following, the formal definition of the metric is given.

Basically, it consists of variants of classical Precision, Recall and F-Score metrics which are modified to include spatial and temporal thresholds acting as matching quality criteria. For set of ground truth annotations G and set of detections D, recall and precision are calculated as follows:

$$Recall(G,D) = \frac{\sum_v \sum_a IsMatched(G^{v,a}, BestMatch(G^{v,a}, D^v))}{\sum_v |G^v|} \tag{3.17}$$

$$Precision(G,D) = \frac{\sum_v \sum_a IsMatched(BestMatch(G^{v,a}, D^v), D^{v,a})}{\sum_v |D^v|} \tag{3.18}$$

where superscripts v and a denotes the indices for video and action, respectively. *BestMatch* function works as a selector, and formally, given a single action $X^{v,a}$ it picks the best possible match from a list of

actions Y^v:

$$BestMatch(X^{v,a}, Y^v) = \arg \max_{a'=1\ldots|Y^v|} \frac{2 \cdot Area(X^{v,a} \cap Y^{v,a'})}{Area(X^{v,a}) + Area(Y^{v,a'})} \tag{3.19}$$

For a ground truth annotation $g \in G$ and a detected action $d \in D$, the binary indicator of correct match is given as follows:

$$IsMatched(g,d) = \begin{cases} 1 \quad \text{if} & \frac{NoFrames(g \cap d)}{NoFrames(g)} > t_{tr} \quad \text{and} \quad \frac{NoFrames(g \cap d)}{NoFrames(d)} > t_{tp} \quad \text{and} \\ & \frac{Area(g \cap d)}{Area(g|_d)} > t_{sr} \quad \text{and} \quad \frac{Area(g \cap d)}{Area(d|_g)} > t_{sp} \quad \text{and} \\ & Class(d) = Class(g) \\ 0 \quad \text{else} \end{cases} \tag{3.20}$$

where $g|_d$ denotes the set of bounding boxes of the ground truth action g restricted to uniquely the overlapping frames of detected action d, and $d|_g$ vice versa. The four thresholds t_{tr} and t_{tp}, t_{sr}, t_{sp} are temporal recall and temporal precision, spatial recall, spatial precision, respectively and they constitute the matching quality criteria. Spatial ones enforce the amount of bounding box overlap, whereas the temporal ones enforce frame overlap, in percentage with respect to ground truth annotations. Specifically, these thresholds control the following:

Temporal Recall Threshold (TRT): Minimum quantity of frames that are common in detected action and ground truth action, with respect to number of frames in the ground truth. In other words, this threshold enforces that a sufficiently long duration of the action is detected.

Temporal Precision Threshold (TPT): Minimum proportion of frames that are common in detected action and ground truth action, to the number of frames in the detected action. In other words, this threshold enforces that duration of detection surplus should be sufficiently small.

Spatial Recall Threshold (SRT): Minimum quantity of overlapping area between the detected bounding box and the ground truth annotation, with respect to the size of ground truth annotation. In other words, this threshold enforces that a sufficiently large portion of the ground truth boxes are correctly found.

Spatial Precision Threshold (SPT): Minimum proportion of overlapping area between the detected bounding box and the ground truth annotation, to the size of detected bounding box. In other words, this threshold enforces that the amount of detection surplus should be sufficiently small.

Although F-score is calculated classically, as given in Eq. 3.21, please note that it depends on the thresholds that impose on the *IsMatched* function. Therefore they can be considered as F-score parameters, influencing on distinct facets of the matching quality: $F(t_{tr}, t_{tp}, t_{sr}, t_{sp})$. Four measures can be derived from here, to assess the detection quality from different quality aspects while keeping other thresholds on a fixed low value, such as $\varepsilon = 0.1$, as shown in Eq. 3.22. In practice, integration of matching quality is calculated numerically with N equidistant samples between 0 and 1; for the reported results, we took

$N = 50$.

$$F = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{3.21}$$

$$\begin{aligned} I_{tr} &= \int_0^1 F(u_{tr}, \varepsilon, \varepsilon, \varepsilon) \mathrm{d}u_{tr} \\ I_{tp} &= \int_0^1 F(\varepsilon, u_{tp}, \varepsilon, \varepsilon) \mathrm{d}u_{tp} \\ I_{sr} &= \int_0^1 F(\varepsilon, \varepsilon, u_{sr}, \varepsilon) \mathrm{d}u_{sr} \\ I_{sp} &= \int_0^1 F(\varepsilon, \varepsilon, \varepsilon, u_{sp}) \mathrm{d}u_{sp} \end{aligned} \tag{3.22}$$

Finally, theses measures are consolidated into one encompassing measure for ranking convenience by simply calculating the average:

$$\text{CombinedPerformance} = \frac{1}{4}(I_{tr} + I_{tp} + I_{sr} + I_{sp}) \tag{3.23}$$

3.5.5 Results

The test set contains 29 test videos with a total of 375 tracklets, which all conform to the aforementioned requirements regarding the camera movement. We compared the proposed method to the method described in [135], which is based on 3D gradients without VMTs. The rest of the classification scheme is identical. Please note that we are unable to compare our experimental results to competition participants that are reported in [327], due to restrictions regarding the stationary camera.

Table 3.1 gives confusion matrices for the baseline method as well as for the proposed method. As can be seen, the proposed method outperforms the baseline clearly. Most confusion is created around three very similar activities, which is typical for this dataset (`EN`=enter/leave room; `ET`=try to enter unsuccessfully; `LO`=unlock and enter). These actions are characterized by people manipulating doors in different ways. In the baseline method, the differences between the action instances are dominated by the differences in viewpoints which makes classification difficult. Extracting the features from a view invariant representation (normalized robust VMT) helps solving this problem. Note that empty rows in the confusion matrix are possible if all samples of this class are detected as *No-Action*, i.e. not considered as detected.

To further analyze the experimental results, we also provide comparison of Recall, Precision and F-Score results in Table 3.2 with fixed quality constraint where all thresholds are set to $\varepsilon = 0.1$. This table also indicates the results for integrated measures presented in Eq. 3.22.

It is evident that our method has a much higher Recall of 0.403, as opposed to 0.129 of baseline method. On the other hand it yields a relatively lower rate of Precision. Derived F-Score results suggest that our method has a better overall performance if quality constraints are fixed at a small value. Other integrated measures, namely I_{tr}, I_{tp}, I_{sr}, I_{sp} and the $CombinedPerformance$ further support this claim

Table 3.1 Detection results are presented in confusion matrices, as percentages. First matrix is for [135], second is for the proposed method; GT: ground truth, D: detection. Please note that empty rows signify the case where all instance of the action are predicted as *No-Action*, which is not shown in the confusion matrix.

GT \ D	DI	GI	BO	EN	ET	LO	UB	HS	KB	TE
DI	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
GI	0.0	0.0	0.0	0.0	0.0	100	0.0	0.0	0.0	0.0
BO	0.0	0.0	0.0	33.3	0.0	16.7	0.0	50	0.0	0.0
EN	0.0	0.0	40.0	40.0	0.0	20.0	0.0	0.0	0.0	0.0
ET	0.0	0.0	0.0	0.0	0.0	100	0.0	0.0	0.0	0.0
LO	0.0	0.0	0.0	0.0	100	0.0	0.0	0.0	0.0	0.0
UB	0.0	0.0	0.0	100	0.0	0.0	0.0	0.0	0.0	0.0
HS	0.0	0.0	25.0	25.0	25.0	25.0	0.0	0.0	0.0	0.0
KB	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100	0.0
TE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

GT \ D	DI	GI	BO	EN	ET	LO	UB	HS	KB	TE
DI	50.0	0.0	0.0	50.0	0.0	0.0	0.0	0.0	0.0	0.0
GI	50.0	0.0	0.0	0.0	0.0	25.0	0.0	0.0	0.0	25.0
BO	14.3	0.0	0.0	0.0	0.0	57.1	0.0	0.0	28.6	0.0
EN	15.0	0.0	0.0	70.0	10.0	5.0	0.0	0.0	0.0	0.0
ET	0.0	0.0	0.0	33.3	33.3	33.3	0.0	0.0	0.0	0.0
LO	0.0	0.0	0.0	33.3	0.0	66.7	0.0	0.0	0.0	0.0
UB	0.0	0.0	0.0	50.0	0.0	50.0	0.0	0.0	0.0	0.0
HS	0.0	0.0	0.0	40.0	0.0	60.0	0.0	0.0	0.0	0.0
KB	40.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	60.0	0.0
TE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100

Table 3.2 *Left:* Recall, Precision and F-Score results with fixed quality constraint where all thresholds are set to 0.1. *Right:* Integrated measures as defined in Eq. 3.22 and the combined performance.

Method	Recall	Precision	F-Score	I_{tr}	I_{tp}	I_{sr}	I_{sp}	Combined
Baseline[135]	0.129	0.290	0.179	0.098	0.132	0.143	0.127	0.125
Ours	0.403	0.125	0.191	0.129	0.116	0.152	0.118	0.128

that our method performs better in terms of Recall while the baseline has advantage on Precision. This may be interpreted as our proposition is more *sensitive* since it produces better amount of true positives but also causes false negatives as a side effect, whereas the baseline method is more inclined to exhibit good amount of true negatives, or more *specific*, but it also misses a notable portion of the actions.

Moreover, comparison of Recall, Precision and F-Score curves are given in Figure 3.9, where each threshold is varied between 0 and 1, while other thresholds are fixed to $\varepsilon = 0.1$. These curves support the remark we made earlier even further, that our method performs better in terms of Recall, while baseline method is stronger on precision. It is also observable that our method is more robust and resilient against variations in spatial thresholds t_{sr} and t_{sp} as illustrated in the figure, both compared to baseline method and to temporal threshold variations. This is arguably an indicator of VMT representation captures the spatial features of a motion successfully, whereas the temporal detection of an activity is more fragile. This is plausibly due to discreet nature of our VMT computation approach, and may be potentially improved in a future work with smaller T values and various overlap amounts with a trade-off to higher computational burden.

3.5.6 Conclusion

For the first part of this manuscript, our objective was to propose a framework for activity recognition task which is particularly invariant to viewpoint changes. We started with Chapter 2, which defines the activity recognition properly and analyze the existing body of work and focus on the difference of global / local representations as well as pose related methods, popular deep learning techniques and strategies that concentrate on view independence. We argue that using deep learning was not a favorable choice in our setting, due to very limited number of videos in the dataset, which mostly like would cause overfitting. In agreement with our initial motivation and considering the state-of-the art on view independent activity recognition, we proposed a robust representation for activities which was complemented with a feature extractor suggestion and a classification scheme. Details concerning our proposition was explained in sections 3.1 through 3.4.

In this section, introduced method was evaluated on a subset of a very challenging dataset which is publicly available. We compared the experimental results to a baseline method that differs from our method in representation of the actions, but identical in terms of classification procedure. This allowed us to demonstrate that our proposition is indeed an improvement with regard to representing actions in a view invariant manner.

To summarize, this part of the thesis introduced a view invariant activity recognition framework and assessed its recognition capabilities with an empirical study. Weaknesses of our method and potential improvement ideas are discussed in Chapter 8.

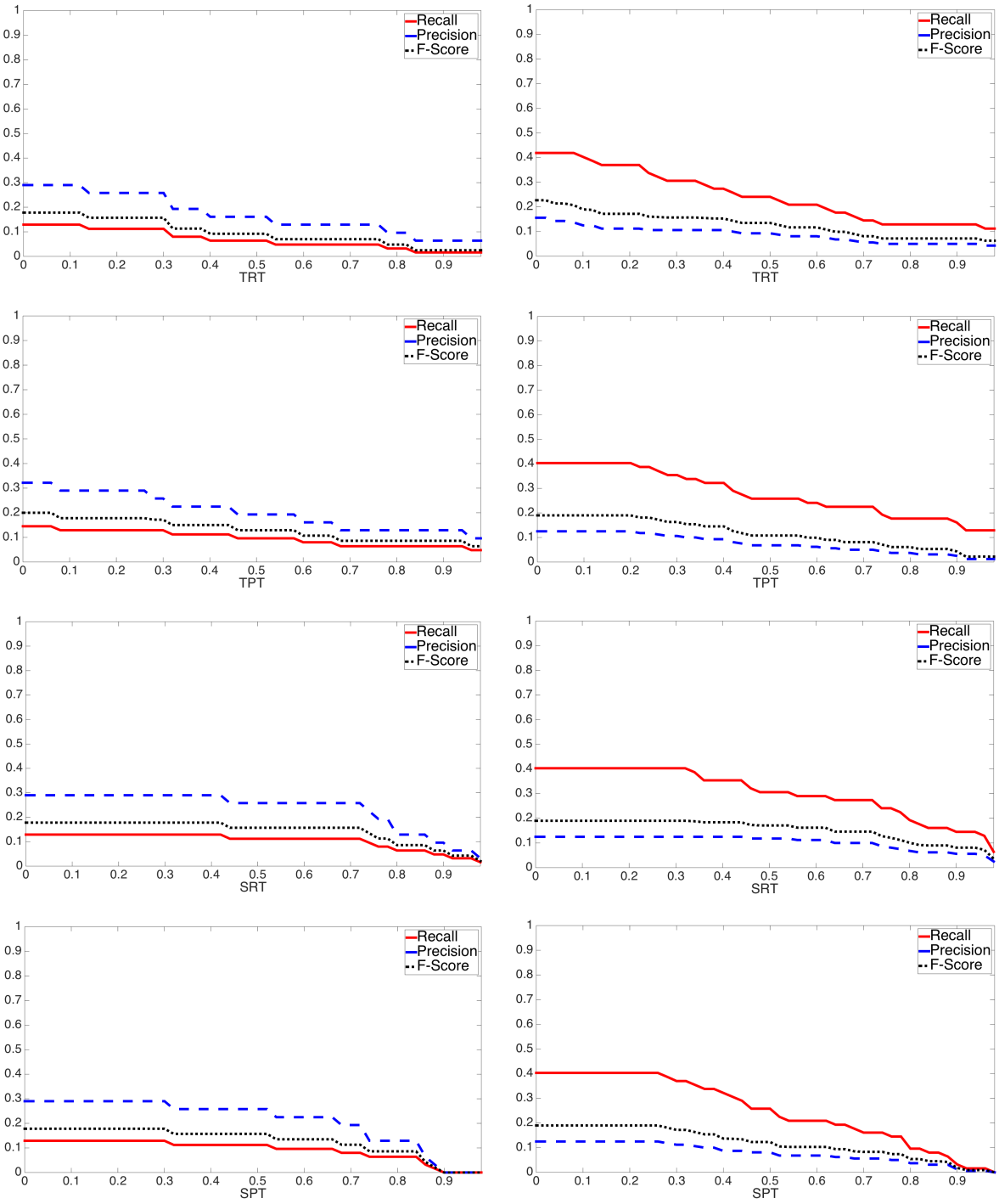


Fig. 3.9 Recall, Precision and F-Score curves are plotted over indicated threshold, while all other thresholds are fixed to $\varepsilon = 0.1$. *Left:* baseline method[135], *Right:* ours.

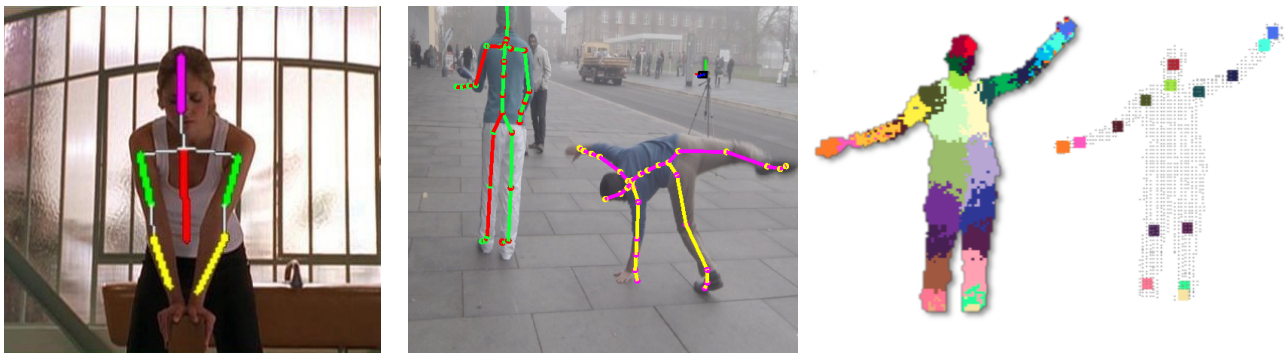
Multi-view Pose Estimation

Chapter 4

Background on Articulated Pose Estimation

4.1 Introduction

Articulated pose estimation is a classic computer vision task where sensor data is analyzed in order to identify the positional configuration of the deformable object in the scene. In other words, it is the recovery of the skeleton which conforms to the image evidence. In human pose estimation, deformable parts become the limbs and estimating the pose become determining the locations of all body parts relative to coordinate system of the image. The outcome of the task is more detailed information compared to human / pedestrian detection, where the goal is to determine the existence and the localization of a human in a relatively large scene, regardless of his / her articulated pose. There are approaches that utilize human detection and tracking as a starting point for pose estimation; while it increases the estimation time, this also limits the variety of the possible poses that can be recovered [62]. In terms of scope of the problem, pose estimation is also different than specialized tasks such as hand gesture recognition or facial expression analysis; hands and the head are merely body parts to be localized within the scope of human pose estimation.



(a) An upper body pose estimation (frame extracted from [69])

(b) A full body pose estimation (frame extracted from [72])

(c) A depth image based pose estimation (frame extracted from [259])

Fig. 4.1 Pose estimation examples.

Popular computer vision tasks such as tracking, activity recognition and video indexing often build on pose estimation. For instance, some tracking methods that track a person with articulation require to be initialized in first frame where they rely on pose estimation, and then use other tracking techniques. Some 3D pose estimation methods and some single-frame action recognition methods utilize 2D pose as an input to their algorithm. Pose estimation is also at the very center of many industrial applications. Human-computer interaction can be established with posture for entertainment and other end-user systems, where users move their arms to interact with a video game, or simply control the volume of the hi-fi system by moving their arms. Mobile robotic systems tracking based security systems are becoming more and more popular along with advanced features such as anomaly detection among others. Film industry benefits already from motion capture systems to modulate computer-generated imagery with actor movements and gestures, where low-budget, markerless pose estimation systems increasingly participate in the competition.

2D or 3D estimation of the articulated pose can be performed based on RGB or depth input data. The problem has been almost solved for easy instances [258]. In cooperative settings for example, depth data proves to be useful and fast. But depth-based pose estimation methods are bound to limitations of the depth sensors they exploit: Indoor environment, close distance subject, little or no occlusion. To handle inner-class variations such as different outfits and body types, depth-based methods usually require a very large training dataset, either synthetic or real, which can be problematic for some scenarios. Other realistic configurations still present a significant challenge, particularly pose estimation from RGB input in non-cooperative settings remains a difficult problem. To achieve good performance in such scenarios, one has to handle all the aforementioned difficulties as well as large variation in scale, color and lighting, different viewpoints, cluttered background and sometimes foreshortening. In Chapter 5, we present an empirical study of some estimation frameworks in different settings.

Methods range from unstructured and pure discriminative approaches, up to complex methods imposing strong priors on pose; some instances are given in Fig. 4.1. The former are frequently used in simple tasks on depth data, which allow real-time performance on low cost hardware. The latter are dominant on the more difficult RGB data, but pose estimation with strong priors is also increasingly popular on depth imagery. These priors are often modeled as kinematic trees (as in the proposed method) or, using inverse rendering, as geometric parametric models, or even detailed triangular meshes. Let us give some brief descriptions for those method types, before an exhaustive and elaborate analysis is given in the following sections.

Purely discriminative methods These methods tend to learn a direct mapping from a feature space (e.g. silhouettes or edges) to human pose. Compared to *generative* methods, which commonly rely on a body model and try to match the image evidence to these models, these methods utilizes more *discriminative* estimation techniques such as decision trees. Details on employed features and example works are provided in Section 4.6.

Parts based models & pictorial structures These methods model the human body as a collection of parts, and are one of the most popular method family for human pose estimation. Basically, the objective of these methods are to learn a body model from data to understand how body parts

look in images and how they can be deformed. With the appearance and deformation models are obtained, body part candidates are determined from image observation and best (if any) configuration of parts that conforms the deformation constraints are marked as detections. Considering the gravity of these methods and their influence with respect to work presented on this thesis, broad description about the approach and underlying formulations will be offered in Section 4.2.

Inverse rendering and optimization Generative models are often used for human pose estimation, which are frequently proceeded by inverse rendering. Basically, the pipeline consists of rendering a 3D mesh in case of depth sensor inputs based on the model parameters, and then comparing the rendering result to the image evidence. Finding a good match between the rendered candidate object and image input usually requires an optimization over the continuous pose space, which are mostly solved through particle filtering, particle swarm or similar iterative optimization techniques. Inverse rendering methods are predominantly used with depth sensors due to convenience of depth data for this task, but RGB methods also exist where binary silhouettes replace the 3D mesh for rendering output.

Deep learning Deep neural networks, convolutional neural networks and recurrent neural networks are often quoted as deep learning. In essence, instead of engineering features for a given task, one employs a neural network with large amount of hidden layers to *learn* the nature of features that are more useful represent the scene in the image. In many challenges deep learning related methods proved to be efficient and accurate. Researchers, engineers and almost all of the computer vision community have quickly adopted the deep learning approach to almost every image processing task, so that it would not be a misjudgment to state that deep learning is the widely accepted standard. Due to its popularity and success, we present a fair amount of study on deep learning in Section 4.3.

Hybrid methods Approaches that combine discriminative and generative models are usually referred to as hybrid methods. They aim to compensate for the lack of discriminative power of generative models and the poor representation capability of discriminative models, by utilizing both approaches. Oftentimes, this approach consists of a discriminative model to produce pose candidates as hypotheses, followed by a generative model to reconstruct image silhouettes (or other artificial data) given those hypotheses [231]. Consequently, this kind of scheme helps to reduce the search space from a continuous pose space to finite set of hypotheses.

There are several surveys that propose their own taxonomy and a way to classify the approaches that tackles pose estimation and recovery problem [196, 197, 215]. Here the categorization is based essentially on the relevance to the method described in Chapter 6, thus for a more complete list and different categorizations the reader is kindly invited to read above-stated surveys. In the rest of this chapter, some relevant categories and corresponding methods will be studied. One of the most common approach for the task at hand –apart from the recent deep learning trend– is *part based models* and it will be reviewed in Section 4.2. *3D methods* are particularly significant for this thesis and will be examined in Section 4.5, since the means to exploit 3D geometry are often similar and comparable. Deep learning

and related methods have lately gained an immense momentum and are being used in almost every aspect of computer vision, therefore a dedicated section (Section 4.3) on deep learning methods for pose estimation is well-deserved. Other methods that worth mentioning but do not fall in the aforementioned categories will be evaluated in Section 4.6. The chapter will conclude by putting those approaches into perspective with respect to the contribution of this thesis, in Section 4.7.

4.2 Part based models for pose estimation

Part based models are a dominant family of models, and they are widely used for RGB images. These models are commonly designed as a collection of body parts or joints, and are often referred to as *constellation based* models or *kinematic trees*. Although the majority of the methods in the literature contain full body parts, some works address pose estimation and motion analysis in TV shows as well as videos and restricts themselves to upper body estimation [69, 240]. The first model in the literature is Pictorial Structures (PS) [83] which dates back to 1973. This model represents the object or the person as a combination of related parts with deformation costs (Fig. 4.2). The "springs" are in fact enforcing the parts to be in certain positions with respect to other parts. Parts that form the object in question are usually represented with an appearance model, i.e. filters learned from features, and their spatial relation are represented with a deformation model. The matching or recognition problem is then reduced to finding parts with similar appearance to the model, and within a "reasonable" configuration with respect to deformation model. Different descriptors for appearance are available such as silhouettes and contours [173], intensity [55], color, texture [9], depth cues [203] and in existence of successive frames optical flow [16].

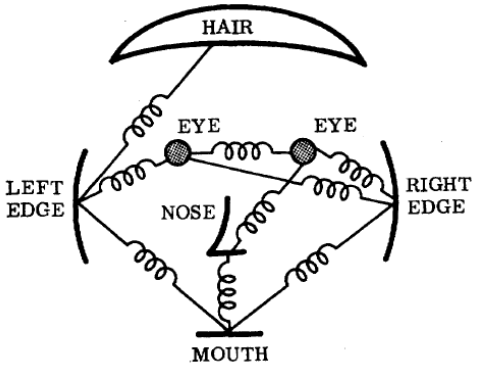


Fig. 4.2 The human face as a pictorial structure: Significant points are represented with *parts*, deformations in between the parts are modeled as *springs*. Figure reproduced from [83]

The rest of this section is organized in three subsections; we first go into details on the statistical framework of Pictorial Structures in Section 4.2.1, a non-exhaustive summary of recent works that rely on part based models are given in Section 4.2.2. Then, we focus on a particular case of pictorial structures, namely flexible mixture of parts in Section 4.2.3.

4.2.1 Pictorial Structures

For many years the computational complexity to solve the pictorial structures problem was considered rather unfeasible, therefore the approach was not thoroughly explored and was not applied to recognition tasks. Instead, model based approaches such as [110, 188] were investigated. In [77, 78, 229] the model was not only formulated as a minimization problem, but also an efficient minimization method was proposed utilizing dynamic programming and generalized distance transforms [79]. If the model structure is acyclic, which is to say it does not contain any 'loops', than the optimization procedure can be carried out exactly and efficiently. More specifically, the object is represented as a graph, $G = (V,E)$ where vertices $V = \{v_1,\ldots,v_n\}$ are the parts, and pairs of connected parts $(v_i,v_j) \in E$ constitutes the edges. The appearance model is defined as a cost function $m_i(I,l_i)$, which measures the matching cost for part v_i in image I at location l_i. It will yield a lower cost if the image evidence is similar in terms of appearance, and higher otherwise. The deformation model is written as $d_{ij}(l_i,l_j)$ which computes the deformation cost between locations l_i and l_j for parts v_i and v_j. Originally d_{ij} is introduced as Mahalanobis distance [165] as:

$$d_{ij}(l_i,l_j) = (T_{ij}(l_i) - T_{ji}(l_j))^T M_{ij}^{-1} (T_{ij}(l_i) - T_{ji}(l_j)) \tag{4.1}$$

where T_{ij} and T_{ji} are transformations that normalize the variables, and M_{ij} is the covariance matrix between the variables. For cases where variables, or dimensions, are linearly independent, i.e. their covariance matrix is the identity matrix, then the Mahalanobis distance is equivalent to the Euclidean distance. In practice, methods that are related to pictorial structures often use Euclidean distance or a simple variant of it [340]. For the deformation term, if the relative positions of parts v_i and v_j are in agreement with the prior information it will produce a lower cost, and higher otherwise. Simply put, this model seeks a configuration in the image where parts are looking "good enough" and are located "well enough". Formally, the optimization function to minimize is formulated as:

$$L^* = arg\min_L \left(\sum_{(v_i,v_j)\in E} d_i j(l_i,l_j) + \sum_{v_i \in V} m_i(I,l_i) \right) \tag{4.2}$$

where L^* is the set of locations that minimizes the function. The statistical framework behind this model is given as a maximum a posteriori (MAP) estimation in [78]:

$$p(L|I,\theta) \propto p(I|L,\theta)p(L|\theta) \tag{4.3}$$

where I is image, L is set of locations l_i and $\theta = (u,E,c)$ are model parameters. This equation actually states that the posterior probability of parts of an object are located at positions L given a model θ and an image I (i.e. $p(L|I,\theta)$), is proportional to the product of the likelihood of seeing an image given that object is configured at L (i.e. $p(I|L,\theta)$) and the prior probability that the object would conform to a particular configuration (i.e. $p(L|\theta)$). The likelihood and the prior probability can be expanded for each

part and can be inserted in Eq. (4.3):

$$P(L|I,\theta) \propto \left(\prod_{i=1}^{n} p(I|l_i,u_i) \prod_{(v_i,v_j)\in E} p(l_i,l_j|c_{ij}) \right) \tag{4.4}$$

where $u = u_1,\ldots,u_n$ are the appearance parameters and $c = \{c_{ij}|(v_i,v_j) \in E\}$ are deformation parameters of the model θ. This is analogous to the energy function in Eq. (4.2) that we want to minimize, if we take negative logarithms of the likelihood as $m_i(l_i) = -\log p(I|l_i,u_i)$ and $d_{ij}(l_i,l_j) = -\log p(l_i,l_j|c_{ij})$ to represent the appearance cost and deformation cost, respectively.

This optimization task is actually an NP-Hard problem for unrestricted graph structures and arbitrary m_i and d_{ij} functions [34]. But as proposed in [78], it can be solved efficiently with two particular constraints: (i) the graph $G = (V,E)$ should be acyclic (e.g. a *tree* shaped structure); (ii) the deformation function d_{ij} should be restricted to a particular form, which is indicated in Eq. (4.1). Given the first constraint, classical dynamic programming can be applied to decrease and the minimization can be performed in polynomial time $O(h^2n)$, where h is the possible locations for each part and n is the number of parts.

To briefly summarize the minimization procedure, let us consider a tree shaped graph $G = (V,E)$ and let $v_r \in V$ be the *root* node of the tree. The restricted shape of the graph dictates that each node other than the root has one and only one parent node and may or may not have child nodes. The nodes that have no children are called *leaf* nodes. For any leaf node v_j, given its parent v_i and location l_j, the quality of the best location can be computed as:

$$B_j(l_i) = \min_{l_j} \left(m_j(l_j) + d_{ij}(l_i,l_j) \right) \tag{4.5}$$

and the best location for v_j is obtained simply by replacing the min with the arg min. Moving up the tree, for any intermediate node v_j which is located at l_j, has children $v_c \in C_j$ and their respective best location qualities $B_c(l_j)$ can be calculated according to Eq. (4.5). Then, this intermediate node v_j with parent v_i will have its own quality of best location as:

$$B_j(l_i) = \min_{l_j} \left(m_j(l_j) + d_i j(l_i,l_j) + \sum_{v_c \in C_j} B_c(l_j) \right) \tag{4.6}$$

Particularly, the equation above indicates that the quality of best location of an intermediate node is calculated in terms of the location of its parent, and consists of three terms: its appearance cost, its deformation cost with its parent, and the sum of qualities of all its children. Finally for the root node v_r, the best location is calculated similarly to Eq. (4.6) except for the deformation term, which is trivial since the root has no parent:

$$l_r^* = \arg\min_{l_r} \left(m_r(l_r) + \sum_{v_c \in C_r} B_c(l_r) \right) \tag{4.7}$$

Following these three Equations (4.5, 4.6 and 4.7) from leaves up to the root to obtain l_r^* in a recursive manner, and then backtracking from atop through to three down to the leaves, the optimal configuration

L^* can be computed. This computation requires $O(h^2n)$ time, because for each possible location of a parent, it should consider every possible location of the child. While this seems like a remarkable gain compared to exponential minimization time in the case of unrestricted graphs, it is still unpractical considering the large amount of pixels in images.

In order to boost the minimization speed even further, it is proposed in [78] to utilize a particular form of the deformation cost d_{ij} that enable us to exploit generalized distance transforms. Distance transforms are calculated on a grid G with respect to a subset of points $B \subseteq G$. Distance transforms are often formalized as $D_f(x) = \min_{y \in G}(\rho(x,y) + f(y))$, where $\rho(x,y)$ is some distance measure between two points x and y, and $f(y)$ is an arbitrary function. This formulation enables a computation time of $O(h)$. Originally, $f(y)$ is used as an indicator function for membership of y in B which yields 0 for cases where $y \in B$, and ∞ otherwise. In other words, the transform searches for a y that is both close to x and has a small value for $f(y)$. With the restriction enforced on d_{ij}, quality of best location function $B_j(l_i)$ of Eq. (4.6) can be expressed as:

$$B_j(l_i) = D_f(x)(T_{ij}(l_i)) \tag{4.8}$$

This can simply be deduced by writing $\rho(x,y)$ as the Mahalanobis distance in Eq. (4.1), in which case $x = T_{ij}(l_i)$ and $y = T_{ji}(l_j)$, and defining $f(y)$ as follows:

$$f(y) = \begin{cases} m_j\left(T_{ji}^{-1}(y)\right) + \sum_{v_c \in C_j} B_c\left(T_{ji}^{-1}(y)\right) & \text{if } y \in \text{range}(T_{ji}) \\ \infty & \text{otherwise} \end{cases} \tag{4.9}$$

Efficient optimization of pictorial structures enabled this model to be used as a baseline, upon which many methods are built. Pictorial structures and part based models in general are extensively used in object recognition, facial recognition and particularly pose estimation tasks, for which we will study several examples in the following sections.

4.2.2 Extensions and Related Work

Minimizing the energy function efficiently as described in the previous section allowed a vast number of methods to emerge, which makes preparing an exhaustive list impractical. In this section, we try to regroup some of the human pose estimation methods which follow the work of [78, 229] based on their approach to the problem.

2D Models The pose estimation community first focused on improving the 2D pictorial structures with carefully crafted enhancements and rectifications. The duality of root and part filters is introduced in [76] which builds directly on pictorial structures. Following the same principle to address deformable nature of the objects and human body, first a coarse and a large filter are applied to the image at low resolution to detect the main body of the searched object. Then parts are detected separately in higher resolution in compliance with the spring-like deformability constraints. [238] extends PS by simply introducing adaptive pose priors to estimate upper body pose. Another PS-inspired model is proposed in [134], where parts are replaced by conditional random fields (CRF). With binary random variables for

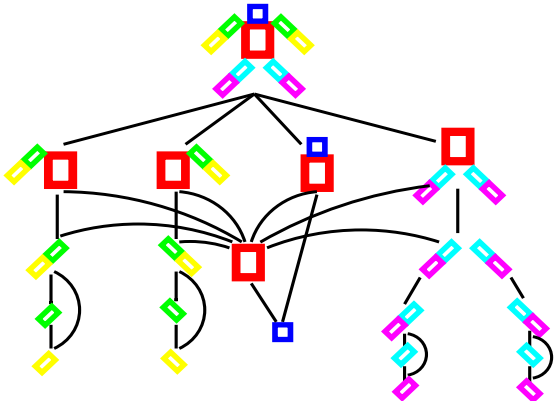
each part, they model the presence and absence for every position, scale and orientation. This results in a very high number of variables, which forces them to opt for approximate inference. To perform pose estimation and motion tracking, [279] introduces the *Sums of spatial Gaussians* model, with underlying color model to represent shape and appearance of the body parts. A large number of parts forms the body model and each part is expressed as a Gaussian sphere, whose diameter is the radius of the corresponding variance of the Gaussian. Unlike most of other body tracking methods this method is fast enough to be considered as a real-time tracking method, but the model is person-specific and should be trained for each actor. [189] benefits from more recent techniques by first obtaining 2D pose candidates with PS, then utilizing a deep model to determine the final pose. [307] takes a relatively uncommon approach to the pose estimation problem by questioning the very structure of the tree model itself. Instead of asserting a tree structure, they propose to learn it from the data which results in a different collection of edges between the vertices. Similar to poselets, they learn by clustering *combined parts*, i.e. sets of two adjacent body parts that further enriches the tree model. [8] builds on both pictorial structures and *strong part detectors* [7], particularly shape context descriptors [170] are used to detect and classify body parts. [86] tackles the human motion capture problem in videos with a multi-layered model based approach. The First layer consists of a stochastic global optimization technique that process the images, extract silhouettes and estimates a preliminary estimate. Then second layer performs filtering and local optimization to further refine the estimation over time and attains a global minimum for the energy function, getting more accurate results as successive frames are processed.

Improving the appearance model It is common practice to attempt to enhance a particular aspect of the pictorial structures framework, such as appearance models. [67] presents a generic body part appearance model that can be used in conjunction with any Pictorial Structure approach. Their method is based on two priors: First, a positional prior that assumes the likelihood of body part positions with respect to other body parts, such as the fact that the torso is usually located below the face. The second prior relies on the likelihood of color consistence between the body parts, e.g. color of the arms are either same as the torso (due to clothing) or same as the face color (due to skin). These premises imply the deduction of appearance of some body part from the appearance of another body part. [10] builds further more on [7] by employing a discriminative appearance model for pictorial structures. To that end, they use densely samples shape context descriptors [170], SIFT descriptors [161] and AdaBoost classifiers. Subsequently, normalized margins of the classifiers are interpreted as likelihoods so that the marginal posteriors can be computed using belief propagation for each part. Furthermore, relationships between the parts are modeled as Gaussian relations; in other words, sum-product algorithm [193] for inference is computed with Gaussian convolutions in joint space, to be eventually re-transformed into the image space. [57] argues that the HOG features [55] that are trained with linear SVMs are sensitive to noise. To address this, they employ random forests as part detectors, and considers both appearance and co-occurrence of the parts.

Poselets and variants Another widely-known term, *poselet* is introduced by [33] to represent small groups of body parts. The novelty of this mid-level representation lies in the type of association that they establish between the tight clusters of appearance and 3D part configuration of the subject in the image, as seen in Fig.4.3a. [318] extends the notion of poselets first by allowing non-rigid parts, or

combination of parts in other words. A second extension is made by presenting multi-scale hierarchy for parts, which can be considered as a nested tree of part combinations that explicitly decompose the human body into part combinations and then parts, as depicted in Fig. 4.3b. A specialized version of the poselets, namely 'armlet's [91], are introduced to tackle the detection task of the arms particularly for cluttered scenes. The main contribution of the method comes from the use of various information such as strong contours, skin color and contextual cues in addition to the standard HOG features, which leads to a richer representation of the arms. [201] argues the importance of relationships between the non-adjacent body parts and proposes a fully-connected model. Due to poselets that become tractable after the observation, their model does not result in a loopy graph structure and can be directly solved.

(a) In appearance and configuration space, clusters of parts or combination of parts are defined as poselets [33]. From top to bottom examples are frontal face, right arm crossing torso, pedestrian, right profile and shoulder, legs frontal view.



(b) Hierarchical representation of pose in [318] where full body is decomposed in poselets and single parts in a tree structure.

Fig. 4.3 *Poselets* are mid-level representations for cluster of parts or combination of parts. They allow multi-scale hierarchical decomposition of body into combination of parts, and further into single parts.

Iterative parsing Iterative parsing of images and its descendants are also an important line of research. [214] tackles the background clutter and body segmentation side of the pose estimation problem by focusing on low-level image features. They treat visual inference as an iterative process, where an edge-based deformable model is first matched to get an initial estimate of the body parts. Using color cues, rough region models are built for each body part, then region based deformable models are re-matched to these regions; the process is repeated to reach satisfactory part estimations. [69, 80] builds on image parsing [214] methods to estimate the human pose. Their enhancement consists of imposing strong priors about the orientations of torso and head, which leads to higher chances of correct estimations for these parts. But the drawback of this method is that it focuses on upper body estimations, moreover, the subject should be upright and only front or back viewpoints are supported.

Unifying techniques A multi-task method [343], takes a step further and addresses two objectives at the same time: human pose estimation and object detection. Interestingly, they turn this seemingly-harder effort to their advantage by benefiting from context awareness. That is, employing object and human parts to serve as context for each other, or 'mutual context' as they name it. In order to achieve this, they

propose a unified model that represents activity, object, body parts and their interrelations. A test image is then scored for each object and body part templates, object detection is achieved by maximizing the likelihood of the image given the models specific activities and finally inference is carried out with a compositional inference method [47]. The required amount of training data is another topic that is open to discussion. For instance, [124] argues that the use of very large and rich datasets is a reasonable way to address the problem of variation in clothing and posture in human pose estimation task. To remain feasible, they rely on the crowd sourcing for ground truth annotation. A recent method [145] proposes a new method to unify several datasets with different annotations, and makes use of powerful deep architectures to first pixelwise part segmentation [103], then human pose estimation with 91 landmark [118] and finally 3D human pose estimation from a single image and the obtained landmarks [30].

4.2.3 Flexible Mixtures of Parts

Flexible Mixtures of Parts (FMP) have been introduced by [341]. A similar work [105] has previously been proposed for object recognition, but its focus was mainly the variation of number of parts instead of variety of the appearance of each part. On the other hand, [341] proposed mixtures instead of orientations of parts as depicted in Fig. 4.4, whose components are obtained by clustering appearance information. Basically, it leverages the idea that appearances of body parts can be grouped for each body part by some similarity measure, and any of these appearance should be sought in the image evidence instead of one generic part template. For instance, searching for a possible appearance of a 'left foot', with different articulations for instance, would yield more precise matches compared to locating a match with a universal template in the image. Please note that estimating the correct appearance type, i.e. mixture, is not rewarded by the estimation performance metrics but experiments suggest that mixture models yield better pose estimation performance. Without a doubt, this advantage comes with a trade-off on larger search space, which inevitably turns into longer inference time. Although, dynamic programming helps to alleviate this problem by reducing the time spent on searching by passing messages between the parts, as described in Section 4.2.1.

Such an approach can be considered as more data-driven, because it will capture the common properties of body parts seen in the training samples. If data contains different orientations of a part for instance, mixtures most likely will implicitly handle those orientations. In any case, the execution time of the FMP method is theoretically shorter due to the gain from the reduced search space, unlike other methods where search for the correct orientation is a burden. A detailed review of the FMP method can be found in Section 6.2. Several extensions have been proposed to improve the accuracy of FMP. For instance, [68] first runs the original FMP algorithm on the dataset to obtain initial estimates, to group them into clusters based on color cues given the foregrounds. This allows to obtain a sensible estimation for the appearance of people with the similar clothing, used to build a color appearance model for each cluster. This pixel-based likelihood information is then translated to unary potentials to re-estimate the human pose with the modified FMP algorithm. In a similar way, [358] proposes a two-staged mixture of parts model that first detects the upper body and classifies its category, then proceeds to the full body estimation from that knowledge. Another method [292] supplements the FMP framework with a hierarchical

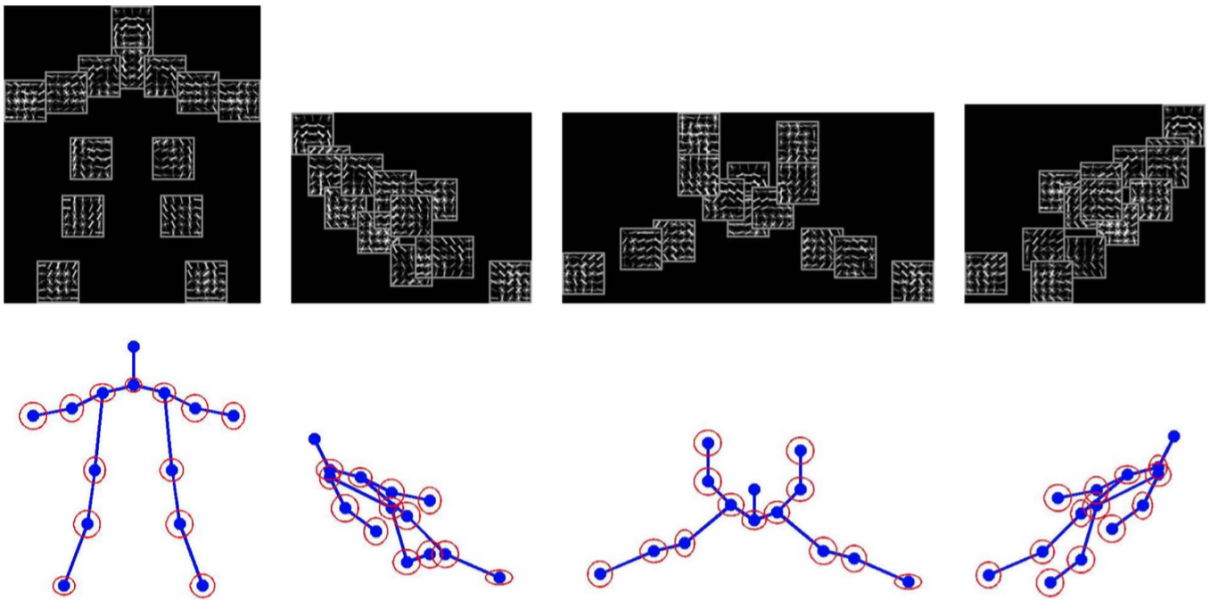


Fig. 4.4 Flexible mixture of parts model for articulated pose estimation with $K = 14$ parts and $T = 4$ mixtures. *Top:* Local mixtures, *bottom:* tree structure. Different mixtures have different best scoring locations with respect to their parents. Here, only four trees are shown, but there are T^K possible combinations of mixtures, each admitting a different estimation score. (Reprinted from [341].)

approach, and introduces latent nodes. These nodes are in fact the combination of spatially proximate parts and they can capture the compatibility of types (or mixtures) of this neighboring parts. Intrinsically, FMP yields multiple pose hypothesis before the estimation is finalized, from which many works benefit to increase accuracy. For example [49] proposed to aggregate multiple hypotheses from the same view using kernel density approximation, essentially introducing a new compatibility term that weighs the part hypotheses considering their parent parts. Similarly, [267] stochastically explores multiple hypotheses and relies on geometric and anthropomorphic constraints for disambiguation. Furthermore, [239] proposes to use strong priors on the human posture. Instead of employing different appearance types (as in [341]), they put forward a structure of graph that has modalities which are learned with clustering from training data. These modes contain different body configurations such as *arm-folded, arm-raised, arm-down* and so on, and the authors report that a setup with 32 modes perform well for human pose estimation.

[123] leans on the idea of 'mixture of trees' [316] model to propose a richer appearance model. They opt for pose clustering to find out similar postures of the human body, which also reduces the appearance variation within a single pose cluster. Moreover, they utilize different set of detectors along with non-linar SVM classifiers for each cluster which eventually favors the implicit capturing of the correlation in appearance between the parts that belong to the same pose cluster. [202] proposes a two-fold extension to the Pictorial Structures by both appearance representations and more flexible spatial models. The former explores the rotation (both absolute and relative) dependent mixtures and pose dependent - rotation invariant mixtures that are implemented through deformable part models [76]. The latter considers the additional prior on location, rotation and scale of the adjacent body parts as well as

mid-level representations such as poselet-conditioned pairwise deformation terms. The work in [46] can also be considered as a mixture model, but instead of vertices, the edges has different types. Another fusion of mixture-of-parts and pictorial structures approaches is [105], though main target of this work is object detection and more specifically recognizing the formations of football players.

Although all these methods and given examples improve and add on the FMP to some extent and report improved performance on pose estimation accuracy, yet no method incorporates geometrical constraints with appearance constraints in a multi-view setting.

4.3 Deep learning methods

In this section, we will focus on human pose estimation methods based on deep learning approaches. For a review of deep neural networks in general, please refer to Section 2.4.

Although a relatively short period of time has passed since the beginning of use of deep learning in computer vision, there is a considerable amount of work on object detection and human pose estimation. Arguably, Krizhevsky et al.'s deep convolutional neural network [138] is among the most pioneering works in the field, especially in image classification domain. They have a successful record in ImageNet Large Scale Visual Recognition Challenge (ILSVRC) of 2010 and 2012; with a network that has 60 million parameters and 650,000 neurons structured in an architecture of five convolutional layers followed by pooling layers, three fully-connected layers and a 1000-way softmax as a final layer. Two years later, same challenge was won by Szegedy et al's *Inception* network [284], which is essentially a larger and a deeper convolutional network with sparse properties and carefully applied dimension reductions to keep the computational cost at a reasonable level.

Learning the features to extract, in contrast to carefully crafting them by hand has become quite popular especially when pre-built models and frameworks were introduced with community support. Namely, *OverFeat* [248], *Regions with CNN features* [90] and *DeCAF* [64] are widely used due to their publicly availability. For instance, [220] demonstrates that off the shelves features computed by *OverFeat* perform better than the handcrafted methods not only for object recognition as it was originally intended, but also other challenging tasks such as scene recognition, image captioning and others. Similarly, [186] shows that mid-level image representations that are learned from a large dataset (e.g. ImageNet [60]), can be transferred to other recognition tasks where available datasets are considerably smaller (e.g. Pascal VOC [73]). Another example of "task invariance" of the features would be the case of *Regions with CNN features* [90], since it is used both for people detection [93], pose estimation and action detection [92].

Pedestrian detection is also carried out with convolutional neural networks, as proven in [249], where the approach is fairly similar to the aforementioned workflow with exceptions such as layer-skipping multi-stage features to integrate global shape information with local motif information. In order to determine various human attributes, a collection of convolutional neural networks are trained in [351], where each one learns a poselet [32] from a set of image patches. An image is then represented by a collection of part-based deep representations which are then concatenated to obtain a full representation. The architecture consists of four stages of convolution - normalization - pooling layers, one fully connected layer and finally a logistic regression layer, which is utilized as a classifier of linear nature. Poselets are

commonly used in conjunction with deep convolutional neural networks for people detection [93] and pose estimation as well [104].

Deep convolutional neural networks are frequently used to tackle the human pose estimation problem, which is the most important task for this thesis. [189] address the problem by first obtaining 2D pose candidates with PS, then employing a deep model to determine the final pose. [74] feeds both local patches and their holistic views into the convolutional neural networks, while [293] propose a new architecture where a deep convolutional neural network is used in conjunction with Markov Random Fields. [294] on the other hand, follow a more direct approach and employ a cascade of deep neural network regressors to handle the pose estimation task. [72] first uses a joint detector [293] which is based on convolutional neural networks. The resulting unary potentials for each joint are then treated as constraints for tracking, where [279] is used as a tracker. [42] introduces a new top-down procedure called *Iterative Error Feedback*, which allows error predictions to be fed back in the convolutional neural network to progressively change the initial solution. That eventually causes the model to be self-correcting and more expressive in terms of features. Belagiannis and Zisserman contributes by introducing a convolutional neural network for pose estimation that combines feed forward and recurrent modules that is able to suppress false detections progressively [21]. A recent study [180] proposes to apply the convolutions and pooling steps in a way that allows the image to be processed repeatedly in a bottom-up and top-down manner with intermediate supervision. [156] proposes to integrate a consensus voting scheme within a convolutional neural network, where votes gathered from every location per keypoint are aggregated to obtain a probability distribution for each keypoint location. A 3D pose estimation method [155] has also been proposed to compute a score given an image and a 3D pose using two separate deep networks that embed the image and the given pose into a common space. Another convolutional neural network is trained to infer 3D human pose from uncertainty maps of 2D joint estimates [357]. To estimate human pose in videos, [199] exploits the ability of convolutional neural networks to benefit from temporal context which is established by combining information between successive time frames using optical flow.

The method proposed in [46] is particularly is interesting, since it uses a kinematic tree and an energy function similar to one in FMP [340]. It consists of both unary appearance terms and binary deformation terms between the parts. Yet, these binary terms are observation dependent, thus do not rely solely on anthropomorphic priors. Furthermore, relation between the parts can have different types, based on their relative location with respect to each other. Types of part relations are learned with K-Means clustering in the experiments and govern spatial connections between the parts. Since both appearance terms and relation terms are image dependent, they can share parameters and can be learned with a single deep convolutional neural network, for which the architecture is illustrated in Fig. 4.5. Output of this network is a conditional probability distribution for both image dependent terms, which is used in the energy function and optimized with dynamic programming (again, similar to FMP) to find the best pose configuration that explains the image evidence.

All the research reviewed in this section and the advances in GPU hardware are pointing to the fact that deep learning will be the industrial and academic standard for almost all the computer vision tasks, if it is not already is. But one problem is standing as a drawback: If there is not a *very* large dataset

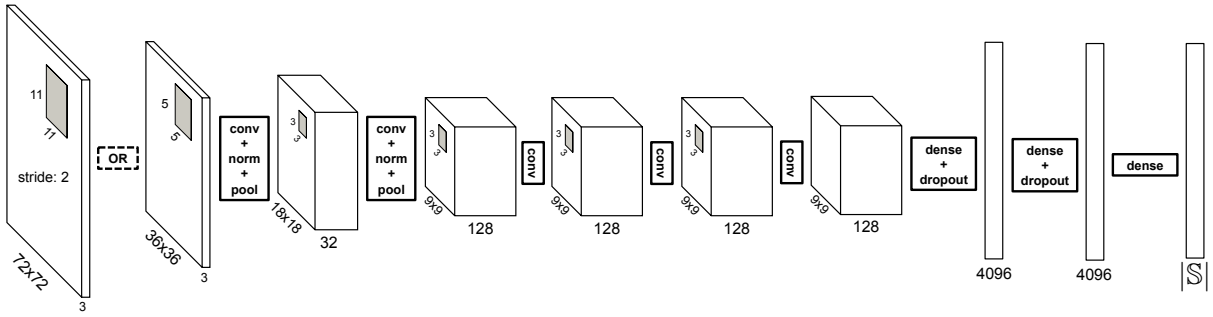


Fig. 4.5 Network architecture used in [46]. Both possibilities for input layers are shown, to handle 36×36 and 72×72 pixels of inputs. After the classical convolution - normalization - pooling pipeline, three fully connected layers with dropout are designated give a softmax output to get conditional probability distribution of an image over parts and connection types.

available for training, these deep networks are cursed with the problem of overfitting, considering the *very* large number of parameters.

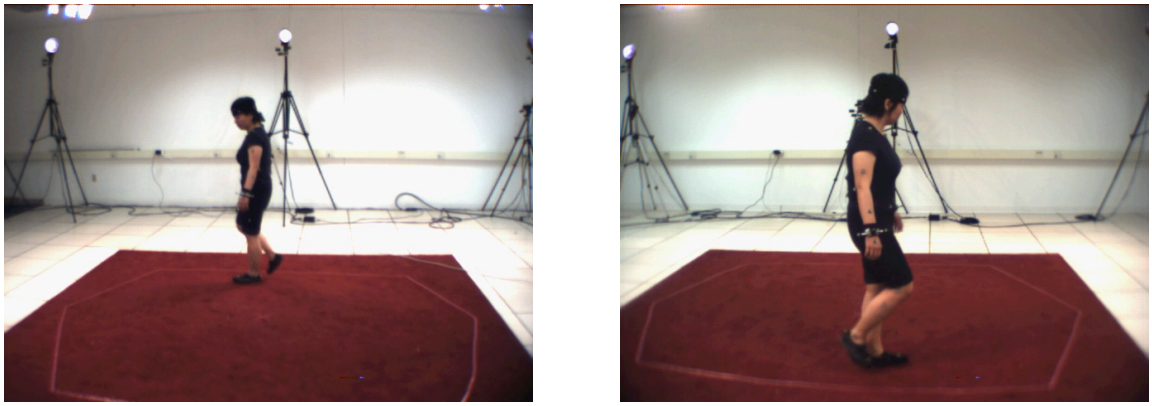
4.4 Multi-view settings

Humans have two eyes, as most mammals and many other animals do. Without going into details of the evolution theory, we could speculate that this is the minimum number of visual sensors to fully comprehend the environment in terms of *depth*. In other words, a living organism seems to need at least two eyes to reliably understand *how far* an object is located. This is due to small differences in what is seen with right eye and left eye, called *disparity*. The amount of disparity is high if the object is at a close distance, and decreases proportionally when it gets farther away. Furthermore, in order to understand the 3D form of an object one must look at it from different point of views, obviously. These primordial statements have inspired researchers to investigate stereoscopic vision systems and multi-view settings in computer vision.

One of the most common use-cases of multiple sensors in computer vision was the depth estimation and the main area of application was navigation for mobile robotics [85]. Modern depth sensors [353] use infra-red patterns and observe their deformations to estimate depth; or time of flight cameras [136] emit a light signal, wait for it to be reflected back and measure the time spent to compute the distance to the object. But before those technologies were available and affordable, binocular (or stereo) camera rigs were utilized [195, 207] to estimate depth information from two images by the means of disparity maps. The reader is kindly invited to read further to understand the task of calculation the disparities based on dense and accurate correspondences between the images [101, 285].

As well as 3D reconstruction and scene reconstruction that are traditionally carried out in multi-view settings, other computer vision tasks such as object tracking, object recognition and action recognition often benefit from multiple sensors. The motivation of using multiple sensors for a task that can be performed with a single one, is basically to have more information about the scene. A simple example would be the case where one camera sees a standing still person from right side, in a way the left arm

remains invisible. If an additional camera is available seeing the same person from another angle, from the left side for instance, in a way that the right arm is invisible this time as observed in Fig. 4.6. If correspondence can be established between the images, information gathered from both views can be integrated to determine the locations of both right and left arms. For images acquired from different sources to be profitable, some form of correspondence ought to be established; otherwise there will be no added value in terms of information. To this end, principles of *epipolar geometry* is commonly exploited to relate some pixels from one image to sets of pixels, more specifically *lines* in the other image.



(a) Person seen from left.

(b) Person seen from right.

Fig. 4.6 An example of setting from HumanEva [261], where each camera is able to see an arm but not the other.

Since a large part of this thesis deals with multiple view pose estimation, the rest of this section will discuss the basics of multiple view geometry. Before starting to review epipolar geometry, let us recall some basic definitions. In the case of a pinhole camera model, which is the most simplified model, any point in 3D space is projected to the image plane using the *camera matrix* or *projection matrix*, P, see Figure 4.7. Here, C is the *camera center* (also known as *optical center*) that is placed on the origin of the 3D coordinate system, so that the Z axis becomes the *principal axis*. All rays of light pass from C and hits the *image plane* in the camera, which is located at a distance of f (*focal distance*) to the camera center. Depicting the image plane in front of the camera center is a common practice to simplify the figures and calculations. Any point X in space is projected to the image plane at the intersection point x of the image plane with the line from X to C. Calculating the coordinates of x is fairly easy using the similar triangles theorem, as can be seen on the right of Fig. 4.7. Thus, a 3D point located at $(X,Y,Z)^T$ is projected to image plane with following coordinates $(fX/Z, fY/Z, f)^T$ and last coordinate can be ignored since it is constant. This projection can be expressed in a matrix multiplication form, if we assume that the image points are represented by homogeneous coordinates:

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{4.10}$$

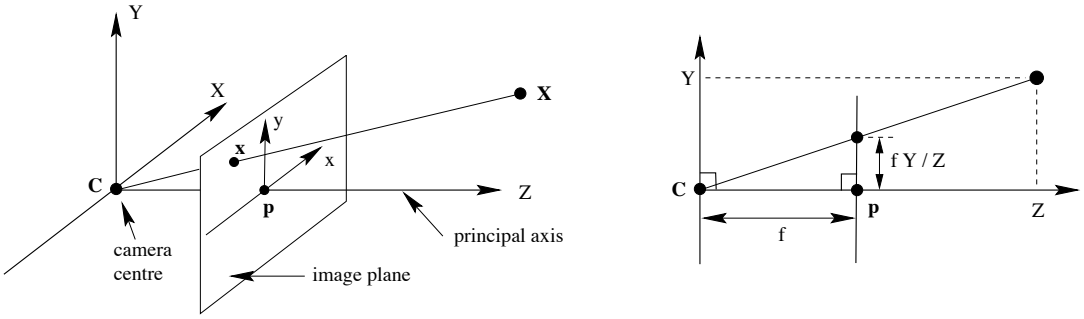


Fig. 4.7 Pinhole camera model geometry, as depicted in [101]. C is the camera center that is placed on the origin.

However, the origin of the coordinate system in the image plane is not necessarily placed at the principal center p. Therefore Eq. 4.10 should be adjusted as follows:

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{4.11}$$

where $(p_x, p_y)^T$ are the coordinates of the principal point. Let us write K for the 3×4 matrix in Eq. 4.11, which is called *camera calibration matrix* and it is defined with respect to the *internal parameters* of the camera. Moreover, another adjustment is required to handle the cases where the camera is not located at the origin 3D coordinate system, also known as *world coordinate frame*. Additionally, the camera is likely to be rotated so that its principal axis would not be parallel to the y axis. The rotation is defined by a 3×3 matrix, R and translation along three axes is defined by a 1×3 vector t. Rotation and translation of the camera is called the *external parameters* of the camera, since it is subject to change every time a camera is placed somewhere and oriented in some direction. By convention, the projection matrix that maps a 3D world point to the corresponding point in image plane considers both internal and external parameters and expressed as follows $x = PX$ where $P = K[R|t]$. In this thesis, we will not get into the details additional parameters that handles various distortions such as *skew* and *radial distortion*, but they are usually plugged in into camera matrix, since they are mostly intrinsic properties of the camera lenses.

All of the internal and external parameters of a camera are generally determined with an estimation process called *camera calibration* [280, 296, 324]. Most of the popular computer vision tools [31] and frameworks uses a *calibration plate*, a special plate with carefully measured patterns on it, that looks like a checkerboard. Multiple images of the calibration plate are captured and expected points are detected with basic matching algorithms. Then these points and their detected projections on the image plane are used to solve a system of equations of type Eq. 4.11. This calibration scheme is introduced in [352] and studied in detail in [101], [85] and [285].

In the case of two cameras that are pointed to the same scene, epipolar geometry and fundamental matrices are used for determining the correspondences between the views. Epipolar geometry is the

projective geometry between two views, that depends on the internal and external parameters of the cameras [101, 330]. It is often illustrated with a point X in the scene and its projections x and x' to two views as in Fig. 4.8a. Optical centers of views, C and C' along with the point X form a plane, namely *epipolar plane* π. The intersection points of the image planes with the line between C and $C\prime$ are called the *epipoles* and denoted as e and e'.

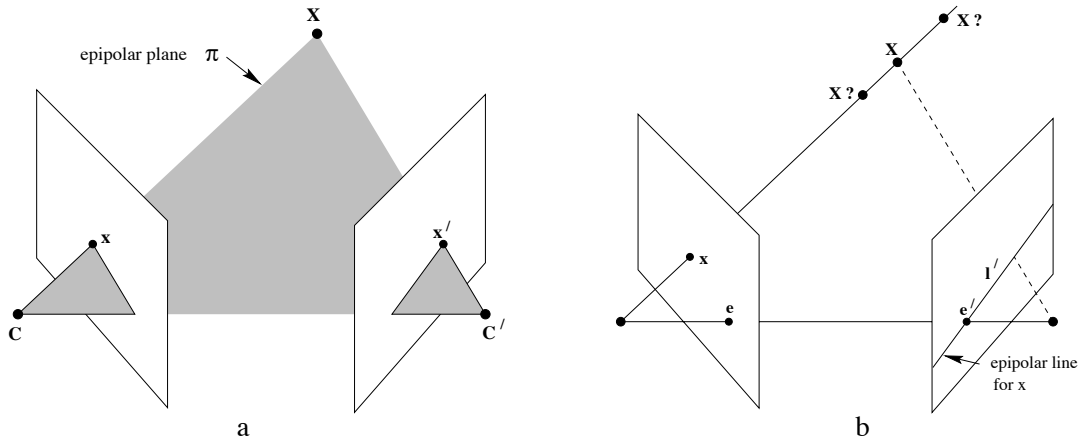


Fig. 4.8 Epipolar geometry and point correspondences as illustrated in [101]. Two views are portrayed with their image planes and optical centers, C and C'.

Assuming that the relative positions of cameras and their internal parameters are known, a strict constraint can be computed from one view. As in Fig. 4.8b, knowing plane π and x is sufficient to limit the location of the x' to a single line, called *epipolar line* that is denoted as l'. In order to calculate the epipolar line, we need an algebraic representation of the epipolar geometry, which is called *fundamental matrix*, F. Using this matrix, mappings from points in one view to corresponding epipolar lines in the other views, e.g. $x \mapsto l'$ and $x' \mapsto l$ can be computed in both directions. The fundamental matrix can be derived algebraically from projection matrices P and P' as proven in [335]. Once the fundamental matrix is derived, it can be used for correspondence of points from two views with the equation $x'^T F x = 0$. The following should also hold for epipoles: $Fe = 0$ and $F^T e = 0$, since all epipolar lines must pass from the epipole in that image plane. But the most handy correspondence of all, which we exploit intensively throughout our multi-view pose estimation scheme, is the one that maps points from one view to corresponding epipolar lines in the other view:

$$\begin{aligned} l' &= Fx \\ l &= F^T x' \end{aligned} \tag{4.12}$$

This correspondence between the views proves to be useful in many applications such as tracking [300], pose estimation [6, 108] and action recognition tasks [44, 313]. The relation between a specific point and its respective epipolar line in the second view, for instance a detected location of a particular object, can be used to modify pixel-wise probabilities or energy function of some task in the second view, and can eventually lead to a more precise result. Additionally, knowing both projection matrices, two 2D points from separate views that is known to be correspondent can be *triangulated* [101] to estimate the

objects 3D coordinates in world coordinate frame. More applications and their details are investigated in Section 2.6 for activity recognition and Section 4.5 for pose estimation.

4.5 3D Pose Estimation

3D pose estimation is a special case of human pose estimation, where locations of joints and body parts are estimated in 3D world coordinates. In comparison to 2D pose estimation, this is considerably more challenging since one have to estimate also the distance from the camera using real world metrics. As seen in Section 4.4 3D information about the observed scene is often achieved from multiple images, either from same or different viewpoints. Other ways to obtain 3D data are also possible, such as 3D sensors that provides depth images or other dedicated 3D sensor hardware that exploits laser technologies and so on.

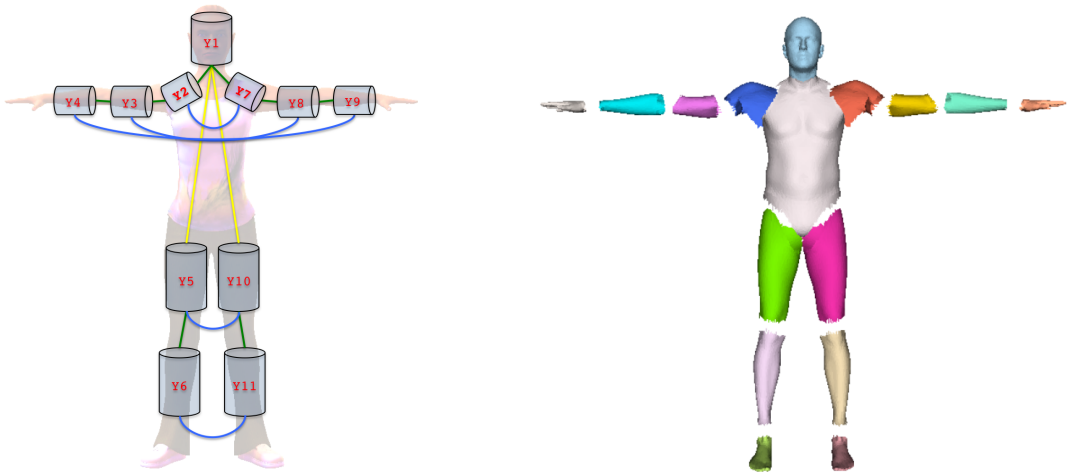
Articulated pose estimation in 3D has popular applications such as markerless motion capturing. A group of techniques focus on the extension of pictorial structures and two examples are shown in Fig. 4.9. Several works [18, 262, 265, 266] proposed a generalized version of pictorial structures, which also exploits temporal constraints. They employ graphs spanning multiple frames and which are inevitably loopy. Inference is performed with non-parametric belief propagation. 3D extensions of pictorial structures were explored, and the burden of the unfeasible 3D search space is handled by reducing it with discretization [37], supervoxels [241], triangulation of corresponding pairs of body parts from different viewpoints [19] or by using voxel based annealing particle filtering [40].

Additionally, using a 3D model allows to use priors such as "body parts must not share the same space" unlike 2D models, where the subject can be self-occluded and more than one part can occupy the same image space. With this kind of prior one can avoid self-intersections, or in other words so-called double counting errors. [241] propose another 3D pictorial structures model, and address the high complexity and intractable dimensions problem by reducing the search space. To this end, they take the discretization-by-segmentation approach by applying supervoxels to the 3D model.

Recently, [359] proposed a strategy similar to 3D pictorial structures, but with a more realistic body model, and inference is carried out with particle-based max-product belief propagation. [6] introduced a scheme where pictorial structures are employed to estimate 2D poses, then incorporates these poses to obtain a 3D pose with geometrical constraints as well as color and shape cues. Inferring 3D pose from multiple 2D poses is rather common, with various underlying strategies such as hierarchical shape matching [107], random forests [130] and optical flow [209].

Quantization of possible human postures to build a codebook of 3D human poses is used by [257]. Then the 3D representation of the test voxels is compared and matching to the nearest pose in the codebook is performed. Since this codebook of available human postures is very large, they also propose an extension of *parameter-sensitive hashing* [252] to complete the matching procedure in an efficient way.

The 3D pose estimation problem can be tackled by transforming the data acquired by multiple RGB cameras into a 3D representation [127] in a way that depth-based techniques, such as [259], can be exploited. To do so, first they extract the 3D visual hall by non-parametric background subtraction [70]. First the orientation of the human body is estimated similarly to [257], then an extended version of



(a) 3D PS model from [18], where additional constraints that act as a deformation prior for 3D parts are shown. Rotation, translation and collision constraints are depicted in green, yellow and blue, respectively.

(b) *Stitched Puppet* [359], is a realistic 3D body model with a graphical structure of separate body parts, which can handle body deformations with *stitching* cost, i.e. multiple springs between the interface points of parts.

Fig. 4.9 3D extensions examples of pictorial structures.

shape context [22] features are extracted in 3D in the human centered reference frame. Segmentation of the 3D visual hull is then performed using the pixel-wise classification of as in [259] and finally joint locations are detected with mean-shift mode finding. Three dimensional stereoscopic videos are exploited to address the human pose estimation task in some cases [160], where two images for each scene is captured from very close cameras, implicitly simulating the human eyes. This setup allows for calculation of the disparity maps, thus the depth information becomes available and the subject body is segmented from background. Then an extended version of pictorial structures is applied to compute the body part configuration probability given image evidence from two cameras and the previously calculated disparity map.

The 3D human pose problem is sometimes dealt with discriminative methods, such as conditional mixture of Bayesian experts [27] or silhouette based techniques [2, 3]. Although these methods are usually fast for feed-forward 3D prediction, they tend to be inefficient to train with large amount of training data. Discriminative methods are reviewed in detail in Section 4.6.

Multiple cameras are not mandatory for 3D pose estimation, instead multiple frames from a monocular camera can be used to estimate the 3D pose. For example [9] infers 3D pose from monocular camera, where first 2D pose is estimated to form tracklets for body parts and and tracked over a time period. On the other hand, [302] argues that bone lengths and absolute depth values can not be estimated through rigid constraints to torso and hip with a finite number of frames.

Alternatively the missing depth information can be supported with strong priors to yield multiple hypotheses of 3D pose from a single 2D pose estimation. [306] presented a method to estimate 3D pose from single image where they use FMP and camera parameter estimation, in conjunction with anthropomorphic constraints. Or, several works [86, 223, 267, 319] start from 2D estimation of human

pose and densely back-project it to 3D to form a high number of hypotheses. By iteratively imposing geometric and kinematic constraints, or by other means of filtering the ambiguity is resolved and a final 3D pose is obtained. Alternative ways of inferring 3D pose from a collection of 2D poses are investigated frequently, using *kinematic jump processes* [273], convolutional neural networks [357], implicit mixture of Conditional Restricted Boltzmann Machines [288] or supervised spectral embedding [347]. A recent method [30] exploits deep convolutional neural networks to obtain 3D pose estimation from a single image, by first obtaining a 2D pose, then estimating a 3D mesh of the body and projecting back to 2D, and finally minimizing a cost function which penalizes the projection error between the 3D joints and detected 2D joints.

Methods also focus on dealing with multiple subjects for 3D pose estimation [18, 19]. To achieve this, they model the 3D pictorial structure as a conditional random field (as in [134]), and they explicitly model the kinematic constraints with rotation, translation and collision components. In a follow-up work, temporal constraints are also incorporated [20]. Inference on the graph that is no longer a tree, and is achieved with a loopy belief propagation algorithm [25], which is actually an approximation.

4.6 Other methods

Methods that are worth mentioning but do not fall into the previous categories will be reviewed in this section. One group would be tracking-related methods, or more specifically methods that use temporal relationships between successive frames. Temporal strategies are commonly used both for pose estimation and articulated tracking in videos. Tracking is often employed in order to establish coherence between the poses over time; whether all parts are tracked separately or an arbitrary body center is tracked. Using spatio-temporal links between the individual parts of consecutive frames seems promising, but intractability issues arise for graph-based methods since the number of connections increase very rapidly. To this end, [48] opt for approximation with distance transforms [79]. [350] reduce the graph by combining symmetrical parts of human body and generating part-based tracklets for temporal consistency. [334] uses a spatio-temporal And/Or Graph to represent poses where only temporal links exist between parts. [336] perform articulated tracking with particle filtering while [152] avoid an explicit body model but estimate the pose using a visual hull instead. Recently, [191] proposed synthesizing hypotheses by simply applying geometrical transformations to initially annotated pose and match next frame with nearest neighbor search. As seen in Section 4.5, consecutive frames can even be used for pose estimation in 3D, although some disagreements have arisen [302].

A second group would be discriminative approaches, where a direct mapping from feature space to pose is learned, often by avoiding any explicit body models (although models cannot be integrated). This body of work has been explored for many years, some of popular researches are dating back to early 2000's. Silhouettes [2, 3] and edges [27] are frequently used as image features in conjunction with learning strategies for probabilistic mapping. There are several examples that use regression [3] and a non-linear supervised learning model called *specialized mappings architecture* [230], as well as Gaussian Processes [299], nearest-neighbor [206, 253] and mixtures of predictors such as Bayesian Experts [260], density propagations [271] and its conditional counterpart [264]. These approaches are usually compu-

tationally efficient and perform well in controlled environments according to various researches, while they are highly dependent on the training data and therefore may generalize poorly in unconstrained settings.

More recent studies usually do not take a fully discriminative approach, but favor generative models in conjunction with discriminative classifiers and similar elements. For instance [72] combines a discriminative deep-learning detector [293] with a generative tracking method [279]. The former is based on convolutional neural networks and operates as a monocular joint detector while the latter tracks the resulting joints with sums of spatial Gaussians. Combination of generative and discriminative models are commonly used for tracking with depth data [12] and hand pose tracking techniques [276]. Pictorial structures are also used in combination with discriminative approaches such as randomized decision forests [57] or a non-linear support vector machine [123] where the pose space is clustered and a pictorial structure is assigned for each cluster center. The combination of discriminative approaches and generative ones are usually motivated by the need of better separation between the object classes, but arguably they fall short in terms of generalizing and tend to perform worse if training data is diverse and extensive.

4.7 Conclusion

In this chapter, we have reviewed several classes of approaches for the human pose estimation problem. First, we started with the definition of the problem and initial motive to achieve the solution. Given RGB input, pictorial structures are the most classical and famous model that we reviewed. Then other pose-related methods that follow the light of pictorial structures were explored considering their novelties and estimation performances. Following that, it was inevitable to mention deep learning methods since they are becoming the industry standard for almost every pattern recognition task. Some deep convolutional and recurring neural networks were reviewed according to their relevance to our work. Seeing that our work is essentially a multi-view method for pose estimation, it seems imperative to give the basic background about camera geometry models and multi-view geometry in particular. The section was then substantiated with some examples of applications where multiple sensors are involved, in order to base a comparison to our contribution. Following that, we investigated the cases where 3D information is obtained with various techniques. Examining the works that infer 3D information from a monocular camera was nonetheless required, since they constitute alternatives to what we are trying to achieve, and it is important to understand in what conditions they are insufficient and inadequate. The final set of models was mentioned in the last section, that is to say temporal strategies and some discriminative approaches. All of the aforementioned techniques were studied both in terms of theory and in terms of applicability to scenarios which we aim to propose a solution.

It should be quite apparent by now that human pose estimation is a very popular and competitive field of study and a great body of work was published in the last 20 years. This chapter was intended to i) Give highlights and point out to the inspirational works that become a real building block that is accepted by the research community ii) Portray the state-of-the-art of the human pose estimation problem with respect to our contribution.

In the following chapters of the manuscript, the preliminary tests of our contribution to human pose estimation will be described in detail (Chapter 5). Following that, our proposition to tackle the estimation problem will be explained in detail (Chapter 6) and our approach will be evaluated with two popular benchmarks later on (Chapter 7).

Chapter 5

Preliminary Experiments

In Chapter 4 it has been substantiated that human pose estimation is an exceptionally active field of research and new methods -or extensions of existing methods- emerge very frequently. Each one of those methods has an advantage over the others; either it addresses a scenario where others fail to handle, or it boosts the estimation accuracy by proposing a novel representation of the human body, or a clever technique for energy optimization etc. In this chapter, we conducted a series of preliminary experiments to assess advantages and disadvantages of certain types of methods. We start by stating the motivation of these preliminary work and continue with the details of the experiments, where the data that we recorded, the methods that we tested and the results that we obtained will be discussed. Finally, we will conclude the chapter with an analysis of the experimental results. In fact, this conclusion reveals the rationale behind our decision regarding the path to follow on the pose estimation task.

5.1 Motivation

Human pose estimation can be achieved under various conditions, and numerous application areas emerge accordingly. Different family of methods have different operational requirements, and have advantages in certain scenarios. Here, we target situations with mobile robots; specifically indoor, large public spaces such as airports, hospitals, museums, shopping malls, hotels, office spaces etc. Figure 5.1 depicts a few examples of mobile robots in such environments.

A common property of these environments is the large, open, publicly accessible space. In this case, a freely roaming robot will most likely to encounter with people and is able to see humans from any distance. Additionally, we are motivated to propose a human pose estimation method that would form an intermediate step for activity recognition in a multi-robot environment. To this end, every method that we consider ought to be tested with various proximity settings.

A related criterion to consider, would be input modality of the data. Most common data for computer vision tasks are either RGB or depth streams; obviously, former is easier to obtain while the latter requires special hardware. The two modalities produce very different types of data, which require particular families of methods to process. Depth imagery has been noticeably popular in the last years and have both advantages and disadvantages in terms of pose estimation. Moreover, depth sensors are known

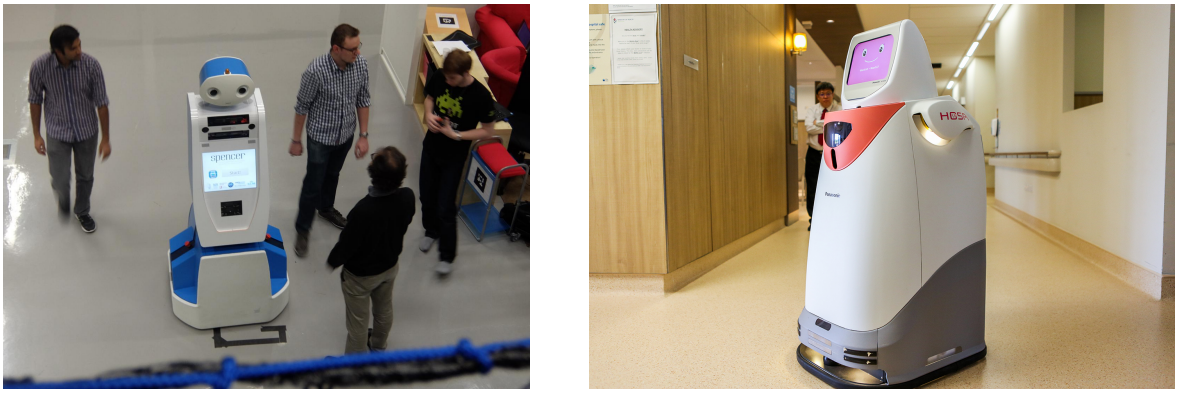


Fig. 5.1 Examples of assistive mobile robots in large, indoor public spaces. *Left:* SPENCER[1] at Schiphol Airport, Amsterdam; *right:* HOSPI[2] at Changi General Hospital, Singapore.

to have hardware limitations with regard to operational distances and environments, e.g. they do not work well with sunlight nor in cases where subject is located farther than a few meters. On the other hand, RGB images are not subject to such limitations, but they also do not provide easily exploitable 3D information about the scene. As a result, RGB methods require some degree of prior information to estimate an articulated pose in most cases. Considering the benefits and inconvenience of these two modalities, it is reasonable to evaluate pose estimation techniques of both alternatives.

In Chapter 4, a non-exhaustive list of various approaches for pose estimation task was reviewed. Discriminative methods and inverse rendering based techniques are well suited for depth input, and usually exhibit decent results in short execution times [258], given that reliable depth data can be provided. Also, these methods are often robust against occlusions due to their non-holistic natures and pixel-wise classification schemes. However, these methods are heavily dependent to the quality of the input data and will inevitably fail when the designated operational conditions are not met. On the other hand, part based models, and methods based on kinematic trees in particular (as seen in Section 4.2.1), are more appropriate to perform with RGB input. This is essentially due to local appearance models, which require RGB data to measure the visual similarity. But this kind of visual input is not sufficient alone, thus is accompanied with a strong prior, e.g. a body structure model. The strong prior on articulated pose, or the expectation to observe a certain type of body part configuration, compensates for the lack of 3D information and enables RGB methods to compete with depth-based techniques in pose estimation task. Furthermore, RGB methods do not require specific hardware and therefore are not limited to aforementioned operational requirements, which practically makes these methods more convenient for uncontrolled environments. Thus, it is a good practice to experiment with both discriminative and part based models to assess their estimation capabilities in various settings.

Short execution times are always desirable, especially on real-time systems such as mobile robots in public spaces, since they require to infer information about their surroundings as they go. On account of execution time, depth based methods have superiority for two reasons: First, modern depth sensors execute a considerable portion of their code directly on the hardware, which yields real-time stream of depth images. Second, employed discriminative methods for pose estimation are intrinsically faster compared to generative models. As for part based models that use RGB images, computational complexity

of the employed algorithm directly translates into execution time. The concerns about large search space for optimization algorithms and faster inference techniques such as dynamic programming and distance transforms, are discussed in Section 4.2. Furthermore, some algorithms are more appropriate than others in terms of parallel implementation, which can then be executed on multiple cores or GPU. Therefore, runtime requirements and ability of quick inference should also be a criterion in our experiments.

Finally, and most importantly, we should evaluate whether a method is compatible for extension to multi-view schemes. This decision is tightly coupled with the nature of our contribution on the collaboration of multiple images. As a principle, we seized upon the idea of *early fusion of information between the views*. In other words, we envisaged a system where some information is acquired from a view should be *transferable* to the other view in a useful format, so that this second view can benefit from that initial finding. Additionally, we intended to propose a scheme, an iterative one possibly, to further profit from the enhanced results in the second view by transferring them back to the first view, and so on. To this end, generative models that employ pixel-wise energy functions or probabilistic maps appear to be more suitable, because information from one view can be translated into another as *additional score* or *increased probability* on certain locations, while other locations can be implicitly labeled as "less likely".

In order to evaluate methods and techniques in the light of these criteria in a reasonable manner, conducting preliminary experiments are imperative. We selected three methods, three distances and three cases for occlusion, then we carried out the experiments with two subjects.

5.2 Experiments

In this section, the preliminary experiments are described in details. First, the content of the recorded data is covered where modality of the data and what challenges we introduced are discussed. Then, we will present our exploratory review of a selection of existing methods and what final three methods are chosen to utilize in the experiments. The section will be finalized with the reports on performance of these three methods with the small amount of data we recorded.

5.2.1 Recorded Data

A classical first generation Microsoft Kinect [353] was selected as an hardware, since it is capable of recording both RGB and depth images of size 640×480 pixel simultaneously. It should be noted that acquired depth data is actually at 320×240 pixels and upsampled via underlying hardware. The restriction towards operating systems and the lack of open source compelled us to not to use the original Kinect SDK, but to use OpenNI[3] instead for data acquisition. In order to make evaluation and assessment of the methods, we wanted to cover as many scenarios as possible with limited resources and within a limited time. To that end, we diversified the recordings with following variables and properties:

[1]SPENCER - Social situation-aware perception and action for cognitive robots, a European research project - `http://www.spencer.eu/`, last accessed on 18/04/2017

[2]HOSPI - A multi-purpose, commercial autonomous delivery robot. Related article: `http://news.panasonic.com/global/topics/2015/44009.html`, last accessed on 18/04/2017

[3]OpenNI SDK and PrimeSense™Sensors SDK which includes the NITE algorithms are discontinued, but latest builds and documentation are available to download on http://structure.io/openni, last viewed on 22.06.2016

Distance: First variable is the distance of the subject to the camera. We decided that three separate distances, specifically close-range (~2 meters), mid-range (~7 meters) and long-range (~10-12 meters), are reasonable to assess both method and imaging modality performances. This variable enables us to evaluate a pose estimation approach in terms of proximity to the recording device.

Orientation: Second property to consider is the orientation of the subject with respect to the camera. Again, most of the existing methods excels their performance if the subject is facing the camera, but we wanted to assess a case of observation from side and back of the subject. Since our contribution will cover a multi-view scenario, it is very likely to that some of the viewpoints will be seeing the person another angle than *en face*.

Occlusion: An additional challenge is introduced by placing objects in the scene to cause occlusions to the human body. Each recording has three versions, one with occluded legs, another one with right or left half of the body is occluded and a clean shot without any occlusion. This variable enables us to assess a method in terms of robustness against partial visibility. The results are especially significant for scenarios where camera attached mobile robots are roaming in a real life environment.

Intra-class variation: Ideally, in order to properly evaluate a pose estimation technique in terms of generalization there should be numerous test subjects with different gender, physical appearance, clothing, skin color etc. In our experiments however, the shootings were performed with two different people for convenience. Nevertheless, it further increased the variety of the image collection to some extent.

Naturally, we did not simply recorded every possible combination of the stated variables and skipped recording the combinations that are not meaningful, such as "recording from side while left of the body is occluded" and similar, and instead ended up with 48 sets of sequences in total. We believe that this amount is sufficient to conduct preliminary experiments and to have a general idea about the method families.

On Figure 5.2 samples from the recorded data are depicted. The first column is recorded from close-range, the second is from mid-range and the third is from long-range. The first two rows are frontal views, where first row has no occlusion while a vertical half of the body is occluded on the second row. In the following two rows the subject is seen from side, where the third row has no occlusion while the fourth row has a noticeable occlusion on both legs. Images on the last two rows are recorded from the back. Again, the fifth row has no occlusion while both legs are occluded on the sixth row.

5.2.2 Tested Methods

Considering our motivation stated in Section 5.1, we decided that it was acceptable to choose at least one part based method that operates on RGB data, at least one discriminative method that estimates the pose from depth data and one supplementary method as a control basis. Before starting the tests we were certain that we would use the methods with pre-trained models and avoid perform any training, in

Fig. 5.2 Example of recorded data, only RGB images are shown. Please see Section 5.2.1 for details.

Table 5.1 Potential part-based methods for preliminary tests.

Method Title	Language	# of joints	Test duration per image
Flexible Mixture of Parts [341]	MATLAB	26	~14 secs
Tree models in HPE [307]	MATLAB	14	~20 secs
MODEC [239]	MATLAB	12	~28 secs

order not to unnecessarily lose time. At that point, we supposed the premise that the good generalization of a method is also an indicator of its good qualities; therefore if method performs well with the pre-trained model on our preliminary dataset, it should be acceptable and promising after a proper training procedure.

For the discriminative model, the work of Jiu et al., who is a former Ph.D. student of our laboratory LIRIS, [121] was a convenient choice. Basically, it is a randomized decision forest method that relies on pixel-wise classification, akin to Shotton's seminal work [259] which also exploits depth data; but Jiu's model is also augmented with RGB cues for edge detection and it features a spatial learning technique which exploits neighborhood relationships between parts. Thus, it is not a skeleton based model, nor involves kinematic trees; but proceeds with segmentation of body parts and then infers the body pose. The method is already implemented and included in the computer vision toolset called *Starling*, which is publicly available[4].

Choosing a model with kinematic trees was particularly harder due to abundance of related methods in the literature, as we review to a certain degree in Sections 4.2 through 4.6. Considering the multi-view compatibility, we focused on part based methods and their descendants. After discarding the methods that were introduced prior to 2010 (such as [67]) we reduced the candidate methods to the list seen in Table 5.1. All methods in the table are part based approaches as described in Section 4.2, specifically descendants of Pictorial Structures.

All implementations of the mentioned methods were downloaded and compiled if necessary. Test durations for a single image are measured on an Ubuntu PC with 2.4Ghz QuadCore processor and 8 GB of RAM. Observing the comparison in table 5.1, Yang and Ramanan's model [341] seems to be faster than others. Furthermore, the model in [307] is an extension of flexible mixture of parts (FMP), but it does not provide a prominent improvement considering the additional execution time. On the other hand, the multi-modal approach proposed in [239] indeed provide some improvements on certain body parts at the expense of twice the execution time. But this approach is less suitable for multi-view extensions due to its complexity which is increased with mode selection steps. As a consequence, we decided on FMP as the part based model to use in our preliminary experiments.

As for the control method, the tracking algorithm that was integrated to the OpenNI SDK seemed practical and sufficient. User has to initially stand in front of the camera to perform initial estimation, then he/she has to open his/her arms wide to perform calibration so that the skeleton is fully detected and ready for tracking. Please refer to next section for preliminary test results.

[4]GitHub page for the Starling project: `https://github.com/liris-vision/starling`, last accessed on 01.05.2017.

Table 5.2 Comparison of three pose estimation methods under different conditions. Qualitative results are shown for different working distance and occlusions in columns, while rows indicate different methods and subject orientations.

		Close range		Mid range		Long range	
Method	View	Clear	Occluded	Clear	Occluded	Clear	Occluded
NITE	Face	Good	Good	-	-	-	-
	Side	Good	Good	-	-	-	-
	Back	OK	OK	-	-	-	-
RDF-C [121]	Face	OK	Poor	Poor	Poor	-	-
	Side	Poor	Poor	Poor	Poor	-	-
	Back	OK	Poor	Poor	Poor	-	-
FMP [341]	Face	Good	OK	Good	Poor	Good	Poor
	Side	Good	Poor	Good	OK	OK	Poor
	Back	Good	OK	Good	OK	Good	OK

5.2.3 Test Results

In this section our findings about the preliminary tests are presented. The selected methods of Section 5.2.2, namely NITE SDK[5] (abbreviated as *NITE* for convenience), randomized decision forests with color cues of Jiu et al. [121] (abbreviated as *RDF-C* for convenience) and flexible mixture of parts of Yang and Ramanan [341] (abbreviated as *FMP* for convenience) were tested against all recordings described in Section 5.2.1. All evaluation was carried out manually and qualitatively, because we argue that normalizing the output of various algorithms, proposing a quality metric and comparing them would be extravagant for a preliminary test. Figure 5.3 gives the results of selected algorithms in close-range. First column are the results of NITE tracking algorithm, second column is RDF-C and the FMP results are on the last column; rows are organized similarly to Fig. 5.2.

For the mid-range tests, where algorithms were tested on images that are recorded from ~7 meters, we noticed that NITE stops working as it was suggested in the implementation notes. Estimation results of RDF-C and FMP are given in Fig. 5.4, where the first column is RDF-C and the second column is FMP. Row organization is the same as in Fig. 5.2. Please note that images of size 640×480 were divided into four regions before the tests for improving test performance. The model provided with FMP had been trained on the PARSE dataset [214], where people occupy most of the image space, contrary to our recordings of mid-range and long-range, where the subject occupies very little image space.

Finally, for the long-range tests RDF-C also stops given any meaningful output; as the operational specifications of Kinect [353] states that depth measurement becomes more and more uncertain after 8 meters and eventually is indefinite. As a result, only results of FMP is presented on Fig. 5.5. While the rows are organized as in the previous figures, this time each column is allocated for a subject.

The qualitative comparison of the results and our final verdict on the selected algorithms is shown on Table 5.2 and further discussed on the next section.

[5]Prime Sensor™NITE 1.3 Algorithms notes, Version 1.0, PrimeSense Inc. 2010, `http://pr.cs.cornell.edu/humanactivities/data/NITE.pdf`, last viewed on 30.03.2017

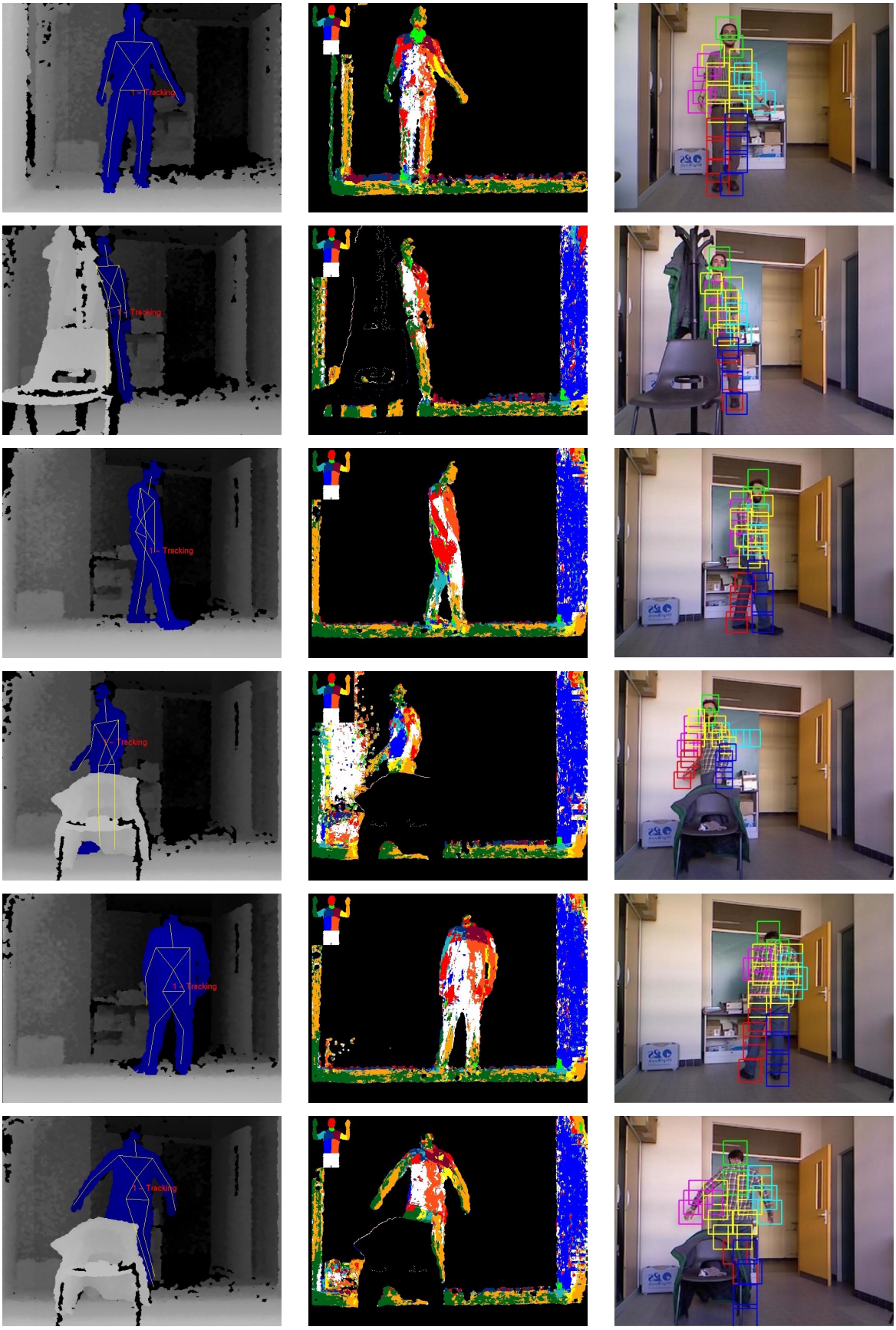


Fig. 5.3 Qualitative results for close-range preliminary tests. Please refer to text for details.

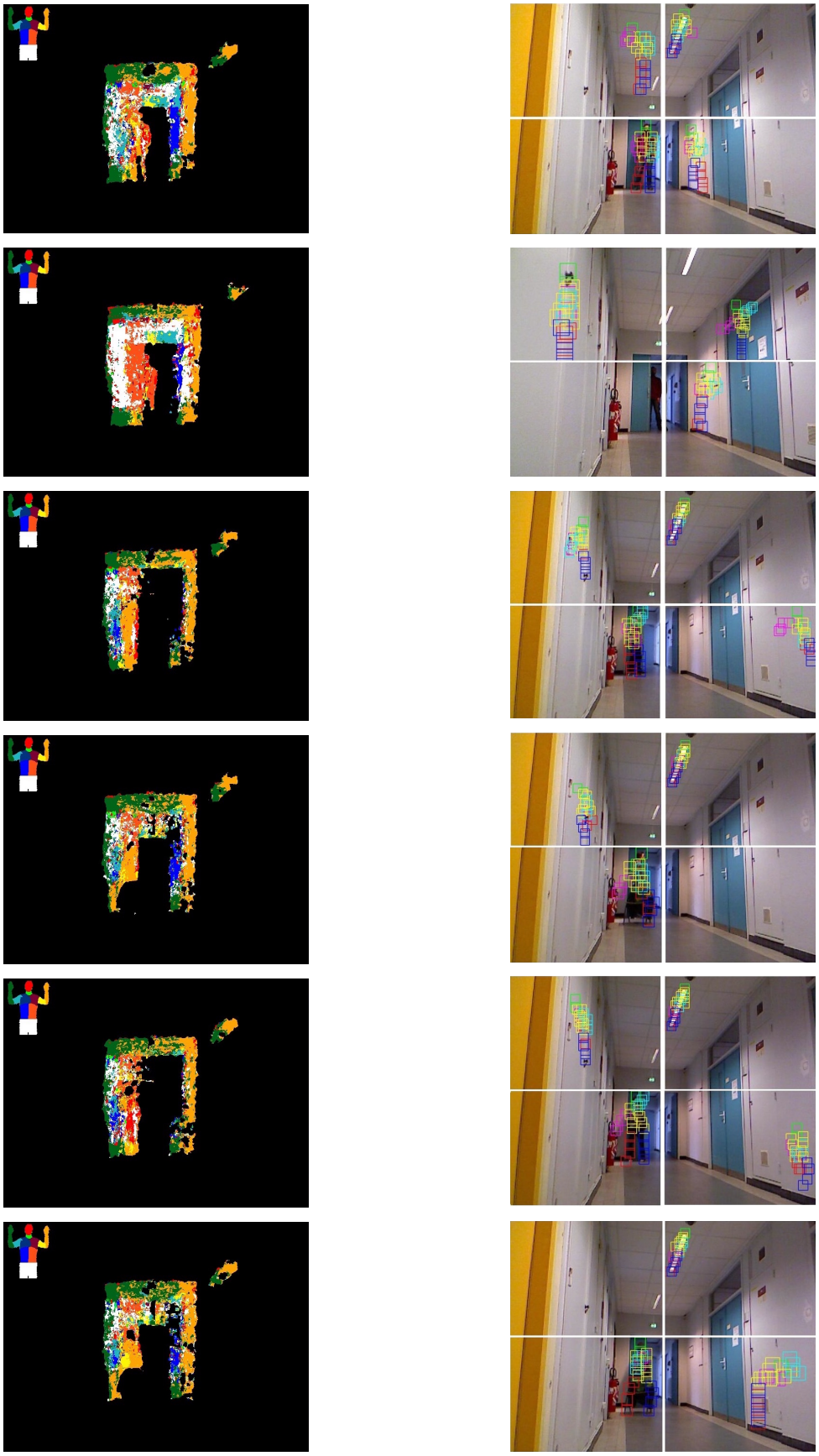


Fig. 5.4 Qualitative results for mid-range preliminary tests. Please refer to text for details.

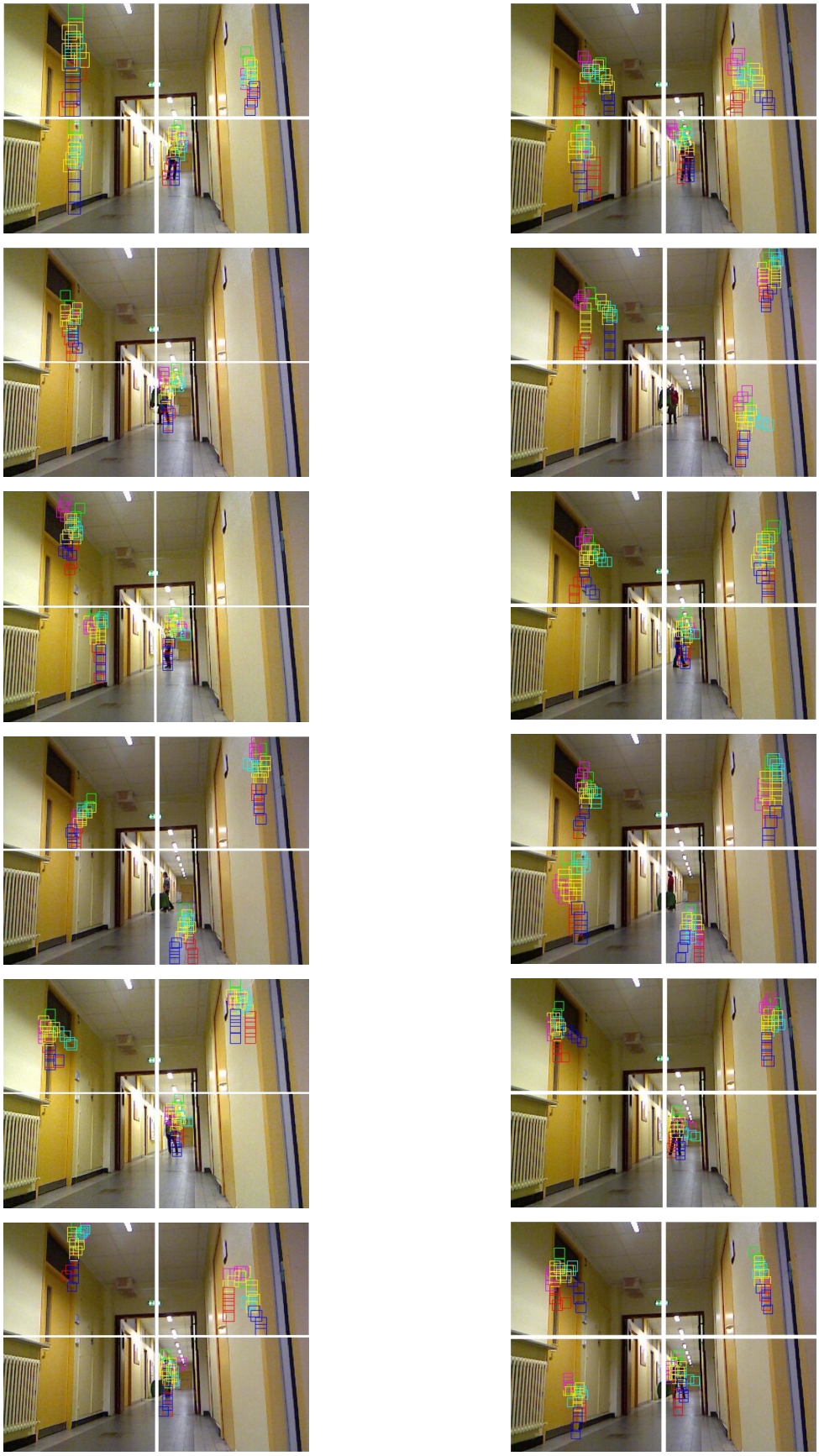


Fig. 5.5 Qualitative results for long-range preliminary tests. Please refer to text for details.

5.3 Conclusion

As explained in Section 5.2, we recorded a relatively small dataset, selected three implementations of different families of methods and executed them on our dataset. In this section the output of these three algorithms will be evaluated qualitatively and our justification for choosing the method as a starting point will be presented.

Let us start with the *close-range* partition of our experiments which is seen in Fig. 5.3 and in the first column of Table 5.2 . It appears to be the ideal working distance for the available depth camera. NITE algorithm performs body tracking very well, even for cases with occlusions and different body orientations; except for the last row where it double-counts the right leg. It should be emphasized that NITE relies on body tracking but not body pose estimation, and initial pose estimation is always necessary *before* the person was occluded by objects. The output of RDF-C is as expected, where some of the body parts are classified correctly even before post-processing. Leg and body occlusions degrade the classifications only for the occluded part, since the underlying method is pixel-based and only considers a small amount of neighboring pixels. Estimations of FMP are at least as good as the others; it is almost flawless in cases without occlusion regardless of the body orientation, a few double-count error in some occluded cases and one very faulty detection on fourth row. For this distance, it is safe to suggest that all three methods have very similar estimation performance. It also should be added that depth-based methods are remarkably faster compared to RGB methods, in this case FMP.

Moving away from the camera, results for the *mid-range* experiments are shown in Fig. 5.4 and in the second column of Table 5.2. As mentioned earlier, NITE did not yield any meaningful detection for this distance and is consequently excluded from the figure. We believe that it would be fair to claim that classification output of RDF-C does not seem very promising in cases where the body is seen without obstruction. It gets worse with occlusion, whether it be for legs or half of the body. The poor performance of the RDF-C is mostly because of the long distance, which is merely in the allowed operational distances for the depth sensor. On the other hand, FMP does not seem to have any trouble for cases where no occlusion is present. Even with the occluded scenes, a considerable part of the body is estimated correctly. As stated before, to avoid re-training and also speeding up the experiments we divided the VGA images into four parts and executed FMP separately. As a result, we observe many false positives in the partitions where no human is present. As seen in Chapter 6, underlying algorithm of FMP yields the global maximum for the pose, hence it is reasonable to anticipate that these false positives will have less score than the correct one and will be discarded when FMP runs on a full size image. Arguably, FMP performs better on mid-range experiments compared to RDF-C.

Finally the farthest ones, results of the long-range experiments are presented in Fig. 5.5 and in the third column of Table 5.2. A distance of ~10-12 meters exceeds the limits of our depth sensor, therefore only the results of the RGB method are shown. It is clearly seen that estimation results are worse than the results of mid-range experiments. The number of false positives is considerably higher, and in many cases of occlusion not a single part is detected. That being said, a remarkable number of body parts are estimated correctly in cases without occlusion.

According to our motivation that is stated in Section 5.1, our goal is to establish a system that is

able to operate on a setting that is multi-agent (i.e. multi-view) that may or may not be located on a large environment. To this end, methods that use depth data are demonstrated to be inefficient after a certain distance, while FMP is not subject to such constraints. On the other hand, like almost any other RGB based method, FMP imposes strong prior on pose. This is because the RGB input is less suited for pose estimation task, compared to depth input, which makes the problem in RGB settings much more difficult. Consequently, methods that use solely RGB input are inherently slower compared to their depth counterparts. Yet, methods from pictorial structures and particularly FMP is intrinsically more adaptable for multi-view extensions, as mentioned earlier.

All experiments considered, the qualitative evaluation of the candidate methods points to following conclusions:

1. Discriminative methods on depth images are faster than kinematic trees on RGB images.
2. Estimation performance of FMP is comparable to, if not better than, NITE and RDF-C in short distances.
3. Depth based methods can not operate in long distances, whereas RGB based methods seem promising in such settings.

One issue is remaining open to discourage us to use FMP as a starting point in our research and that is the execution time of the algorithm. Nonetheless, implementation of FMP that are used in this chapter is written in MATLAB, which is known to be rather slow. We address this lack of execution speed by porting the implementation to C++ and CUDA[6] for partial parallel execution on GPU, and the implementation details can be found in Section 7.4. Our verdict was that the FMP was indeed the best available option present at that time.

In the following chapter, our single-view and multi-view pose estimation methods that follow the lights of Yang and Ramanan's FMP are described in detail.

[6]CUDA is the proprietary parallel computing platform of NVIDIA: `https://developer.nvidia.com/cuda-zone`, last accessed on 01.05.2017.

Chapter 6

Multi-view Pose Estimation

6.1 Introduction

Throughout Chapter 4, background and state-of-the-art on human pose estimation are presented and prominent works were cited accordingly. We discussed about existing methods with various approaches, body models and inference techniques. In Chapter 5, we put forth our motivation in this thesis and presented our preliminary experiments to validate our methodological choice for our contributions.

In this chapter, we propose our multi-view proposition to tackle the human pose estimation task. In the remainder of this section, our stance towards the problem is explained and the overview of our method is presented. In the following sections, we first describe the single-view model for pose estimation in details and develop our multi-view solution with reference to single-view one. To that end, we introduce two novel constraints that are only applicable in a multi-view setting, which are elaborated in sections 6.3.1 and 6.3.2. Moreover, in a multi-sensor environment not always all sensors can provide equally useful data to solve the problem, therefore their influence to the solution can be balanced to maximize the quality of the estimation. The concept of adaptive viewpoint selection is discussed in Section 6.4. Thereafter, the training procedure of the proposed model is discussed in Section 6.5. Finally, the inference technique and our iterative scheme for pose estimation is described in Section 6.6.

6.1.1 Overview

We consider a multi-view setting with two or more calibrated cameras visualize a scene where a person is present. Cameras can be installed on simple tripods or they can be mounted on mobile robots; but the robots are assumed stationary within the scope of our research. Additional topics such as automatic camera calibration and compensation for ego-motion are planned as future work and discussed in Section 8.3.

In a multi-view setting, it is anticipated that one of the cameras will have a relatively better positioning compared to the other cameras, with respect to the orientation of the human body that is targeted for pose estimation. The fact that one of the cameras has a better potential to observe the person was our impetus to establish a medium to share information between the viewpoints. In other words, we were motivated to form a way of transferring the information between the cameras to eliminate the disadvan-

tages, such as self-occlusions due to certain point of view. For instance, let us imagine a scenario where a camera sees a person from some angle in a way that only some parts of the body is visible. If there is another camera that sees the same person from different angle so that it can observe the missing parts, it can help the other view to collaboratively produce a better pose estimation. Fig. 4.6 depicts an example for this case.

In order to fulfill this information transfer, we take advantage of two constraints that we impose and we introduce an iterative scheme to perform incremental estimations that gradually improve. The overview of this scheme, also illustrated in Fig. 6.1, is as follows for a two-view setting:

1. Two initial estimations are produced in a single-view manner, with corresponding confidence scores.

2. The pose with the higher score is marked as the *support pose*, which acts as the information provider for the first iteration.

3. Based on the support pose, two different constraints are enforced to the next estimation step:

   (a) Body part based geometric constraint encourages some locations in the target image for each part, while penalizing other locations that are unlikely to found the parts.

   (b) Appearance constraint promotes certain appearances for each part given an appearance of parts in the support view.

4. The estimation is carried out on the target image with these two additional information acquired from the support pose.

5. At the end of each iteration, the newly estimated pose is assigned as the new support pose and the estimation is repeated on the other view with switched support / target roles.

To put it another way, given a part position and appearance in one view, we are able to infer probabilities of locations and appearances in the other view. We argue that this additional information allows us to obtain better pose estimations from both images. This iterative scheme tends to stop at the global maximum that is supposed the be the point where two poses are at their best, but we also utilize some stopping conditions for the rare cases that it oscillates between two local maxima.

6.2 Single-view pose estimation

Along the lines of [341], an articulated pose in a single 2D image is modeled as a flexible mixture of parts (FMP). Related to deformable part models introduced by [76] which is actually a star shaped model for object recognition, part based models for articulated pose estimation are classically tree structured. Respecting the same principles, our model is a kinematic tree on which a global energy function is defined including unary terms where image evidence is attached, and pairwise terms acting as a prior on body pose. The underlying graph is written as $G = (V, E)$, where vertices are body parts and edges are defined on adjacency between parts. In traditional methods, the body model consists of part templates

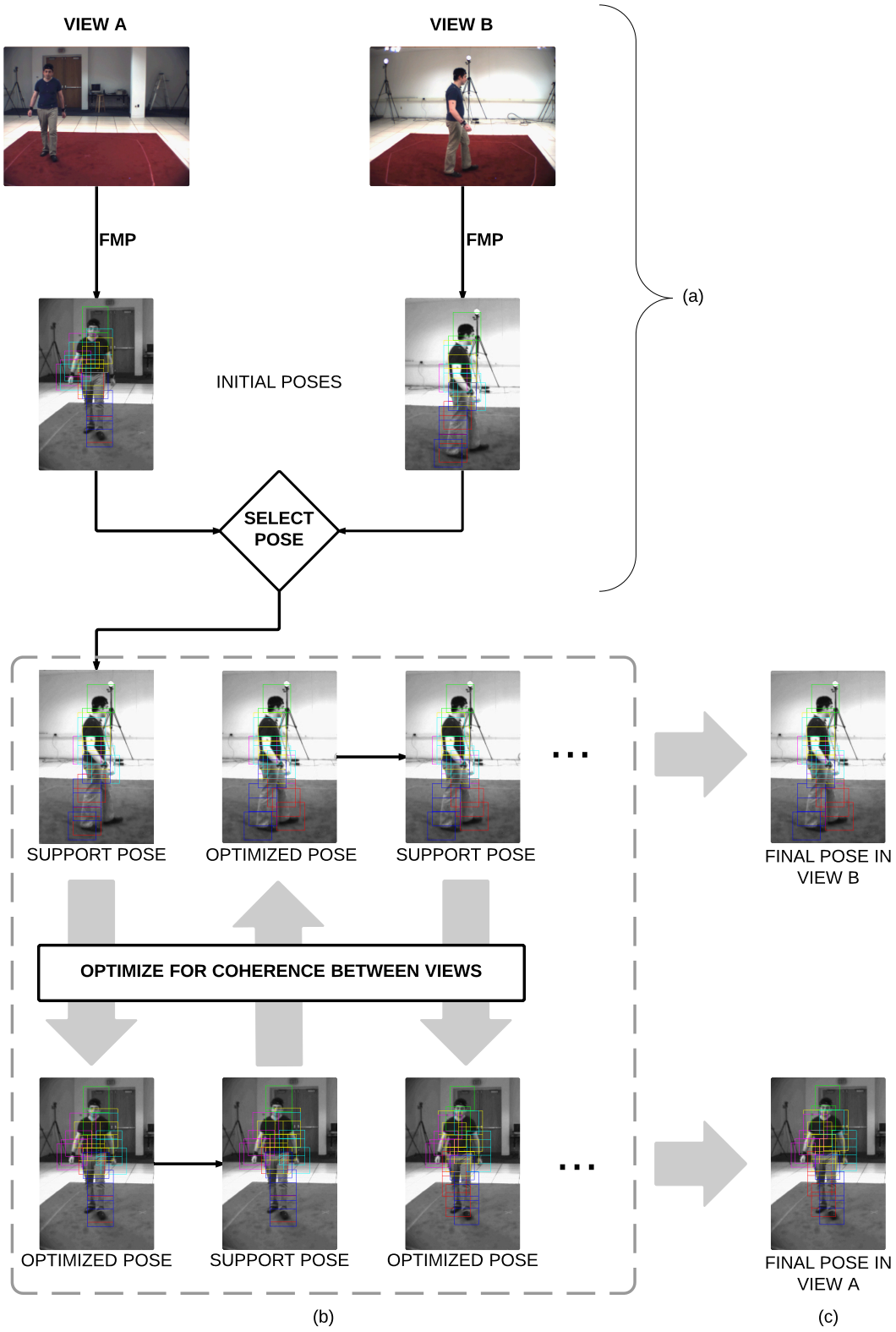


Fig. 6.1 Method overview: (a) initial pose estimation running the single-view model on each view separately. The pose with the highest score is selected as the support pose; (b) joint estimation loop with geometrical and appearance constraints. The newly obtained pose becomes the support pose at the end of each iteration; (c) After convergence, the last two poses are returned as the final results.

which are searched in test stage with different orientations. In this model however, a local mixture of small, non-oriented parts are used to model the articulation instead of family of warped templates. By using local mixtures of parts, the change in appearance of body parts are captured and resulting model is adept to represent different articulations of the body. The mixture of small parts tend to facilitate capturing contextual co-occurrence between parts and permit better encoding of the spatial relationships in between body parts, in a way that local rigidity of the human body can be implicitly represented by the model.

Let $p_i = (x, y)$ be the pixel coordinates for part $i \in \{1, \ldots, K\}$ in image I. We would like to determine the position of each part in the image, which is equal to optimization over the values of p_i for each i. Additional latent variables t_i with $i \in \{1, \ldots, T\}$ model a *type* of this part, which allows to model terms in the energy function for given types. This introduces supplementary content to the pose estimation and effectively creates a powerful mixture model, where both location and type for each body part should be considered for the optimization. Three possible appearances of part *right foot* are illustrated on Fig. 6.4, where different part templates would yield different appearance scores for each case. Ideally, a configuration of conforming part types for each body part should produce the maximum score, for instance types where all parts are seen from left side, therefore finding the best match given the image evidence. In practice, the part types are learned during training, and they are set to cluster centers of appearance features and the related position of each part with respect to its parent part.

In the single-view version, the energy function corresponds to the one given in [341]. This energy function is defined over a full pose $p = \{p_i\}$, latent variables (or *part types*) $t = \{t_i\}$ and for the input image I with the following formulation:

$$S(I, p, t) = \sum_{i \in V} w_i^{t_i} \phi(I, p_i) + \sum_{ij \in E} w_{ij}^{t_i, t_j} \psi(p_i - p_j) + \sum_{i \in V} b_i^{t_i} + \sum_{ij \in E} b_{ij}^{t_i, t_j} \tag{6.1}$$

The expression in the first sum corresponds to data attached terms, where $\phi(I, p_i)$ are appearance features extracted at p_i from the image I (HOG in our experiments, see Section 7.2). Note that the corresponding trained parameters $w_i^{t_i}$ depend on the latent part type t_i. In other words, each position of the image is evaluated for every part and part type in terms of appearance similarity. For instance, some position might yield a high score for a clenched right fist (a part type), whereas the score calculated at a completely different position might be similarly high for a stretched right hand (another part type).

The pairwise terms in the next expression model the prior over body pose using a classical second degree deformation between adjacent parts $\psi(p_i - p_j) = [dx\ dx^2\ dy\ dy^2]^T$ where $dx = x_i - x_j$ and $dy = y_i - y_j$. They control the positions of the parts with respect to their parents and this terms act as a "switching" spring model where the switching controlled by the latent part types t_i.

The last two sums define a weak prior over part types. First one is a unary part type bias $b_i^{t_i}$ which signifies that some part types are more likely then the others. Second one is a pairwise part type term $b_{ij}^{t_i, t_j}$ that put emphasis on some combinations of types of different parts, for instance, an upwards hand is most likely to see in combination with a vertical lower arm.

Although scale information is not specified in the equations, a pyramid is built from the image in the implementation and features are extracted from every level of this pyramid. This is to address the

various sizes of the subjects in the image. Inference in this model (and in our generalization to multi-view problems) will be addressed in Section 6.6.

6.3 Multi-view Pose Estimation

In this section, we generalize the single-view model that is presented in Section 6.2 to multiple views and show that both geometrical consistency constraints and appearance based constraints can be leveraged to improve estimation quality. These constraints are explained in details in Sections 6.3.1 and 6.3.2, respectively. Without loss of generality, let us fix the number of views to two for the remainder of this chapter, and please note that formulations with three or more viewpoints are straightforwardly derivable.

We consider a setup with calibrated cameras where internal and external parameters of the cameras are known. A global energy function models the pose quality over a pair views A and B, where input images I^A and I^B are acquired respectively. This energy function seeks to estimate pose variables p^A and p^B, position of body parts in local coordinates of view A and view B, while additionally optimizing over latent part types t^A and t^B which are the types of the parts that are observed in corresponding views. This global energy function is formalized as follows:

$$\begin{aligned} S(I^A, I^B, p^A, p^B, t^A, t^B) &= S(I^A, p^A, t^A) + S(I^B, p^B, t^B) \\ &\quad + \alpha \textstyle\sum_{i \in V} \xi(p_i^A, p_i^B) + \beta \sum_{i \in V} \lambda(t_i^A, t_i^B) \end{aligned} \tag{6.2}$$

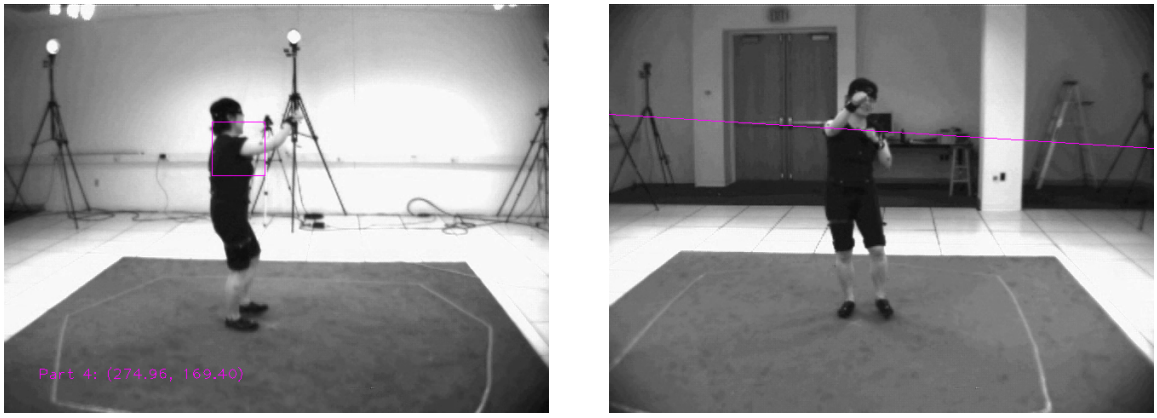
Here, $S(I^A, p^A, t^A)$ and $S(I^B, p^B, t^B)$ are the single pose energy functions from equation 6.1. The two additional terms ξ and λ ensure consistency of the poses over the two views and their descriptions are given in the following sections.

6.3.1 Geometric Constraints

The principles of epipolar geometry, as we described in Section 4.4, are exploited to encourage spatial consistency between the views for the pose estimation task. Assuming temporal synchronization, images I^A and I^B show the same articulated pose from two different viewpoints. Given calibrated cameras, points in the first view correspond to epipolar lines in the second view as depicted for right shoulder part in Fig. 6.2.

In practice, for each estimated part in the support view, an energy map can be computed from the corresponding epipolar line. Locations that are near to the epipolar line should have higher energy, since it is more likely to find the part there; other locations should have decreasing energy as the distance from the epipolar line increases. This energy map can then be plugged into the target view to encourage some positions while penalizing others for the part estimation process. The procedure of adding the energy map to the appearance feature responses for a single part is portrayed in Fig. 6.3.

This additional geometric constraint can be integrated to the global energy function to strengthen consistency between the views and it works in both ways. The geometric term ξ from equation 6.2



(a) (b)

Fig. 6.2 Detected part in view A shown on (a) in magenta. Epipolar line in view B is calculated based on the center point of the bounding box, shown on (b).

leverages this constraint as follows:

$$\xi(p_i^A, p_i^B) = -d(p_i^A, e(A, p_i^B)) - d(p_i^B, e(B, p_i^A)) \tag{6.3}$$

where $e(A, p_i^B)$ is the epipolar line in view A of point p_i in view B and $d(\cdot,\cdot)$ is the Euclidean squared distance between a point and a line. In other words, for a given part position p_i^A in view A, this term ξ prioritize the locations that are close to the corresponding epipolar line $e(B, p_i^A)$ in view B, inclining the energy function towards this locations, since in view B that same part is more likely to appear in this neighborhood.

6.3.2 Appearance Constraints

The geometric constraints described in the previous chapter are imposed on the solution targeting the positions p_i. The term $\lambda(t_i^A, t_i^B)$ of equation 6.2 adds additional constraints on the latent part type variables t_i, which further pushes the result to consistent solutions. Recall that the latent variables are clusters in feature space, which implies that they are related to types of appearance. Appearances might of course be different over views as a result of the deformation caused by viewpoint changes; for instance appearance of a facing forwards head is different from the appearance of a side view of a head. However, some changes in appearances will likely be due to the viewpoint change, whereas others will not. Intuitively, we can give the example of an open hand in view A, which will certainly have a different appearance in view B; however, the image will not likely be the one of a closed hand. Let us give another example that is depicted in Figure 6.4: A foot seen from front is quite different then its appearance from left and from right. Assuming that the part types are conforming with these appearances, which may not be true all the time, one can expect to see a left pointing foot from the left-side viewpoint and right pointing foot from the right-side viewpoint.

We suggest that this changes in appearance are coherent between the viewpoints. We propose that they can be exploited to improve body pose consistency between the viewpoints, by prioritizing some

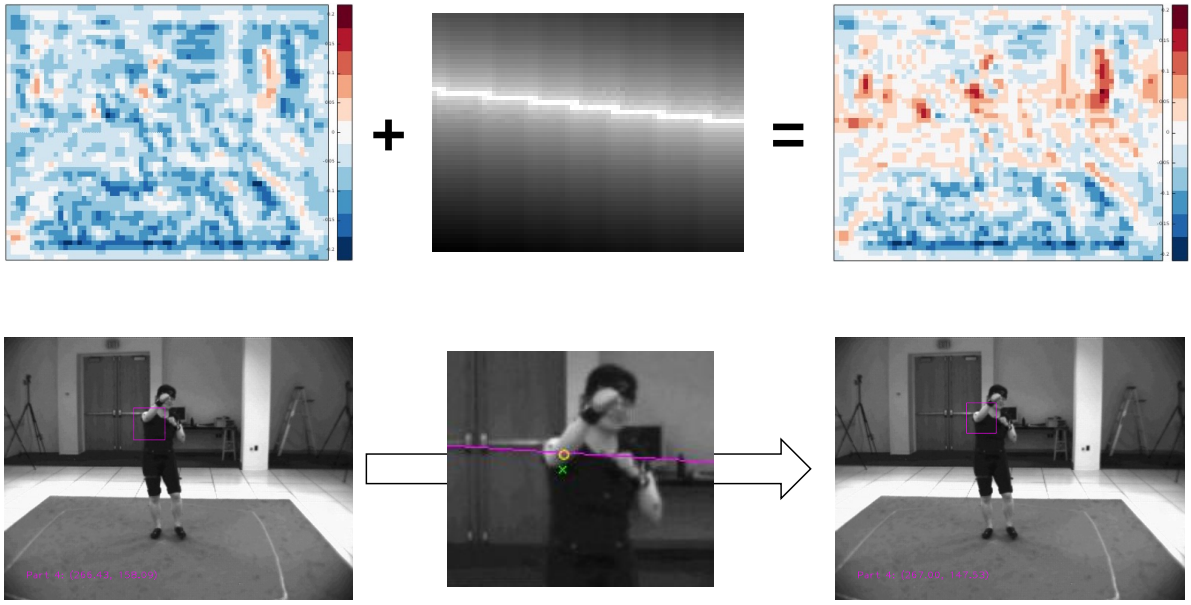
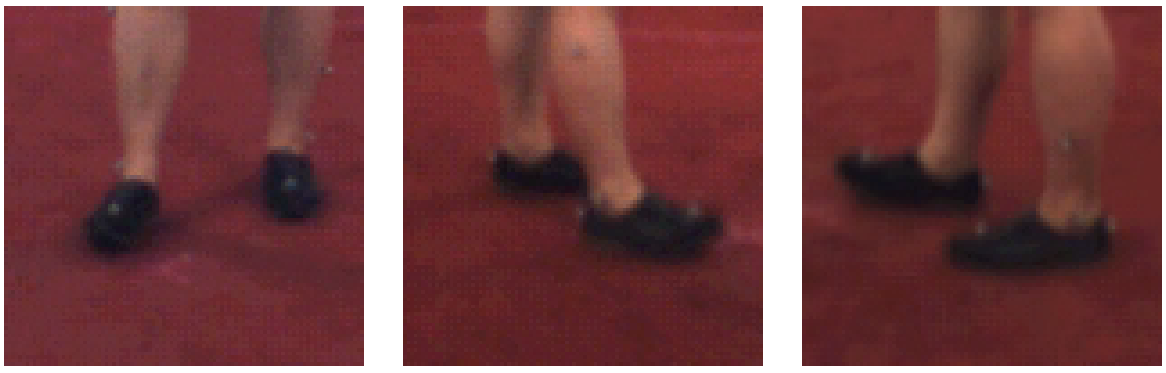
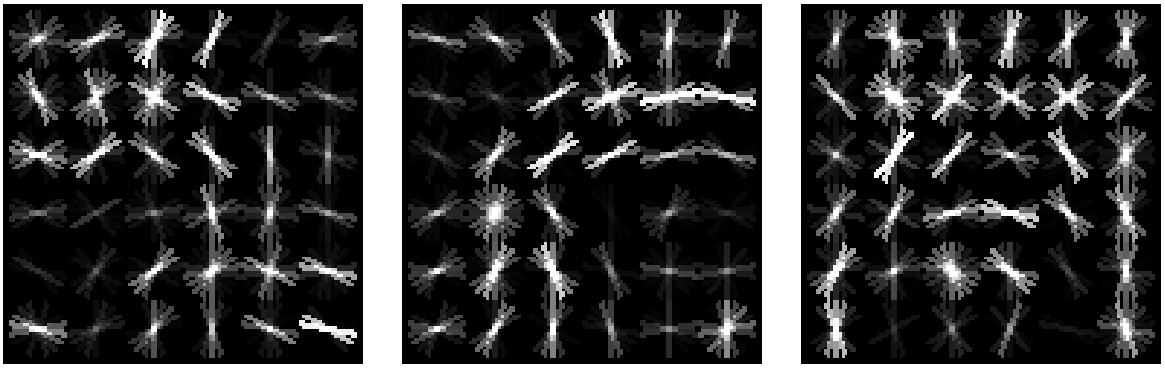


Fig. 6.3 *Top:* Epipolar energy map (*center*) is added to the initial feature response map (*left*), where final energy map for part estimation is seen on the right. *Bottom left:* Estimated position for the part, without the geometric constraint. *Bottom center:* Close up display of displacement of the part position towards the magenta epipolar line, where old position is marked with a green cross and new position is marked with a yellow circle. *Bottom right:* Part position is estimated jointly, this time considering the geometric consistency.



(a) (b) (c)

Fig. 6.4 Right foot seen from front-side viewpoint (6.4a), from right-side viewpoint (6.4b) and from left-side viewpoint (6.4c).



(a) (b) (c)

Fig. 6.5 Illustration of the multi-view consistency term over latent appearance (part types). The HOG filter pair (6.5a)—(6.5b) is highly compatible, whereas compatibility of pair (6.5a)—(6.5c) is low.

part types over the other with respect the viewpoints. Particularly, given a part type and a viewpoint, some part types are more likely to occur than others in another specific viewpoint. Our aim is to benefit from these co-occurrences of the part types and learn their compatibility to reach more accurate part estimations in a multi-view manner.

We model these constraints in a non-parametric way as a discrete distribution learned from training data, i.e. $\lambda(t_i^A, t_i^B) = p(t_i^A, t_i^B)$ (see Section 6.5). Figure 6.5 illustrates this term using three filter examples shown for the learned model of part *right shoulder*. The value of λ term, that is their *compatibility*, is high between (6.5a) and (6.5b), but low between (6.5a) and (6.5c). Intuitively, (6.5a) and (6.5b) look like the same 3D object seen from different view angles, whereas (6.5a) and (6.5c) do not.

6.4 Adaptive viewpoint selection

Geometric and appearance constraints rely on the accuracy of the initial single-view pose estimates as seen in Fig. 6.1. In certain cases, the multi-view scheme can propagate poorly estimated part positions over views, eventually deteriorating the multi-view result. To solve this problem, we would like to estimate beforehand, whether an additional view can contribute, i.e. increase performance, or whether it will deteriorate good estimations from a better view.

We propose an adaptive viewpoint selection mechanism and introduce a binary indicator vector (over parts) that switches on and off geometric and appearance constraints for each part during inference. If an indicator is switched off for a part, then the support pose does not have an effect on the optimized pose for this part. The binary indicator vector a is given as follows:

$$a_i = \begin{cases} 0 & \text{if } \sigma_i(p^A, \theta) > \tau_i \text{ or } \sigma_i(p^B, \theta) > \tau_i \\ 1 & \text{else} \end{cases} \tag{6.4}$$

where τ_i is a threshold obtained from median part errors on the training set and $\sigma_i(p^A, \theta)$ is a function with parameters θ that estimates the expected error committed by the single-view method for part i, given an initial estimate of the full pose p^A. The binary indicator vector, a integrates into Eq. 6.2 as

follows:

$$\begin{aligned} S(I^A, I^B, p^A, p^B, t^A, t^B) \quad &= S(I^A, p^A, t^A) \qquad\qquad + S(I^B, p^B, t^B) \\ &+ \alpha \textstyle\sum_{i \in V} a_i \xi(p_i^A, p_i^B) \quad + \beta \textstyle\sum_{i \in V} a_i \lambda(t_i^A, t_i^B) \end{aligned} \tag{6.5}$$

We implemented σ as a mapping that is learned as a deep convolutional network taking image tiles cropped around the initial (single-view) detection p^A as input. Training the network requires to minimize a loss over part estimation errors, i.e. an error over errors, as follows:

$$\min_{\theta} || \ \sigma(p^A, \theta) - \mathbf{e} \ ||_2 \tag{6.6}$$

where $\mathbf{e}$ is the vector of ground truth errors obtained for the different parts by the single-view method, and $||\cdot||_2$ is the L_2 norm which is here taken over a vector holding estimations for individual parts.

We argue that such a network is suitable to anticipate whether an individual part is useful for multi-view scheme, by implicitly learning multi-level features from an image tile. For example, self-occluded parts or other poor conditions would most likely to be associated with high error rates, whereas unobstructed clean views would yield low errors. Thresholding the output of the network, namely the error estimations σ_i, can provide the decision whether the support view has an influence for part i or not.

6.5 Training

In this section, the learning procedures of the single-view and multi-view models are described. First, the training process for the single-view is explained including the learning of the latent part type variables. Then, determination of the multi-view parameters such as the part type consistency parameters and such is specified.

6.5.1 Single-view parameters

The parameters related to the single-view (appearance coefficients $w_i^{t_i}$, deformation coefficients $w_{ij}^{t_i,t_j}$ and part type prior coefficients $b_i^{t_i}$ and $b_{ij}^{t_i,t_j}$) are learned as in [341]: We proceed by supervised training with positive and negative samples, I_n, p_n, t_n and I_n, respectively. The optimization of the objective function is formulated as a structural SVM, similar to proposition in the seminal work of Felzenszwalb et al. [76]. To benefit from their formulation, let us merge part type and position variables into a new one, $z_n = (p_n, t_n)$. Also, it should be noted that single-view scoring function from equation 6.1 is linear in terms of model parameters $\beta = (w, b)$, therefore we can formulate the scoring function as $S(I, z) = \beta \cdot \Phi(I, z)$. Then the model that we aim to learn can be written as follows:

$$\begin{aligned} arg \min_{w, \zeta_i \geq 0} \quad & \frac{1}{2}\beta \cdot \beta + C \sum_n \zeta_n \\ \text{subject to} \quad \forall n \in \text{pos} \quad & \beta \cdot \Phi(I_n, z_n) \geq 1 - \zeta_n \\ \forall n \in \text{neg}, \forall z \quad & \beta \cdot \Phi(I_n, z_n) \leq -1 + \zeta_n \end{aligned} \tag{6.7}$$

where ζ_n are the slack variables that are used for penalizing the violations of the above constraints, which simply impose that positive samples should score higher than 1 and score of the negative samples should be less then -1 for all part positions and part types. For structured prediction task a negative training set is not necessary, but a trained model with negative samples as well includes an implicit "detection" component into the model, where model produces high score on ground-truth poses and low score for images with no people in it. The learning problem described above, which is a structured SVM, can be solved with appropriate solvers such as [76, 82].

In practice, part type coefficients are learned with respect to their relative positions to their parents by clustering. This mixture of parts approach ensures the diversity of appearances of part types where their appearance is associated with their placement with reference to their parents; for example a left-oriented hand is usually seen on the left side of an elbow, while a upward facing hand is likely to occur above an elbow.

6.5.2 Consistency parameters

Other than the single-view parameters that must be learned for the single-view model, three others parameters are ought to be learned for the multi-view pose estimation method. First and the most interesting one is the consistency parameters between the part types which capture the compatibility amount between the part types with respect to viewpoints.

The discrete distribution $\lambda(t_i^A, t_i^B) = p(t_i^A, t_i^B)$ related to the appearance constraints between views is learned from training data as co-occurrences of part types between the viewpoint combinations. We propose a weakly-supervised training algorithm which supposes annotations of the pose (positions p_i) only, and which does not require ground truth of part types t_i. In particular, the single-view problem is solved on the images of two different viewpoints and the resulting poses are checked against the ground truth poses. If the error is small enough, the inferred latent variables t_i are used for learning. The distribution $p(t_i^A, t_i^B)$ is thus estimated by calculating histograms of eligible values for t_i^A and t_i^B. In practice, the single-view method is executed for each view separately, and part types for the resulting poses are recorded for every training sample. After that, all the obtained pose estimations are verified against the ground truth poses. If the estimation of a part is not within the acceptable error margin, it is discarded. Remaining poses are considered good estimations given their positions, so their part types are accounted for parameter learning. An histogram is formed each part type and viewpoint, where bins are all the part types for that part. This histogram is then normalized for further use in the inference. Figure 6.6 shows an example of all learned filters for lower right arm and their compatibility is given in table 6.1. Please note the symmetry around the diagonal for part types 6.6b and 6.6c: The second row of the table should be interpreted as follows: the most likely part type for the lower right arm is the 6.6c in view B, given part type 6.6b in the view A. And vice versa for third row. It should be also noted that in this example view A and view B are the viewpoints that the person is seen from its left and right sides, and filters 6.6b and 6.6c seem like the view of the lower right arm from corresponding angles. This dual interpretation is also valid for part types 6.6d and 6.6e.

Other two parameters, namely α and β from equation 6.5 are the amount of the contribution for their

Table 6.1 Part type compatibility matrix for lower right arm. Every row is probability distribution of a part type in view B, given the part in view A.

Part type in Fig. 6.6	6.6a	6.6b	6.6c	6.6d	6.6e	6.6f
6.6a	0.000	0.019	0.000	**0.600**	0.025	0.356
6.6b	0.037	0.090	**0.875**	0.000	0.000	0.000
6.6c	0.000	**0.651**	0.256	0.000	0.000	0.093
6.6d	0.238	0.000	0.000	0.091	**0.671**	0.000
6.6e	0.002	0.000	0.000	**0.721**	0.223	0.054
6.6f	0.454	0.000	**0.515**	0.000	0.031	0.000

respective constraints. These parameters are learned and fine-tuned with simple grid search approach using the validation set.

6.5.3 Neural network weights

As seen in Section 6.4, σ is a mapping that estimates error of a single-view pose estimation, given an image tile cropped around the bounding box. To determine σ, we use regression of the expected error and train a deep convolutional neural network. In practice, it is rather uncommon to train a deep convolutional neural network from scratch, that is, initializing the weights randomly and learning these on a very large dataset with a backpropagation scheme of choice. Instead, common practice is to make use of a scientifically acclaimed convolutional neural net, such as *AlexNet* [138], *ResNet* [103], *VGG-16* [269] or similar, that is pre-trained on a very large dataset, such as *ImageNet*, as a starting point for the task at hand. This procedure is often called *transfer learning*, and involves either using the pre-trained network as-is, i.e. employing it as a fixed feature extractor; or *fine-tuning* the convolutional neural net, i.e. replacing the classifier / regressor part of the network with a suitable top model, fixing the lower parts of the convolutional network (or *freezing* them) and allowing the higher parts of the convolutional network to be updated with backpropagation along with the top model, according the dataset at hand. It is theoretically possible to fine-tune the network completely by not freezing any part of it, but due to overfitting concerns it is strongly discouraged. The underlying motivation for such an approach is the observation that lower layers of a convolutional neural network usually involve low level features such as edges, corners or color cues while the higher parts of the network tend to learn dataset specific high level features. Furthermore, the training of unfrozen layers and the top model is usually conveyed with a small learning rate and usually simple backpropagation algorithms such as *stochastic gradient descent* to prevent substantial changes that are caused by large loss produced by the unseen data. Therefore, it is a reasonable and popular strategy to maintain the low level features as they are and slightly modifying the high level ones for the specific task requirements.

Here, we use a VGG-16 network [269] pre-trained on *ImageNet*. After removing all the top fully connected layers we replace them with a single small hidden layer for regression. We fine-tune the last convolutional block of VGG and learn the weights of the newly added fully connected layers with augmented data. Further details regarding the data augmentation, architecture of VGG-16, modifications thereof such as the appended top model and the procedure of fine-tuning are discussed in Section 7.2.

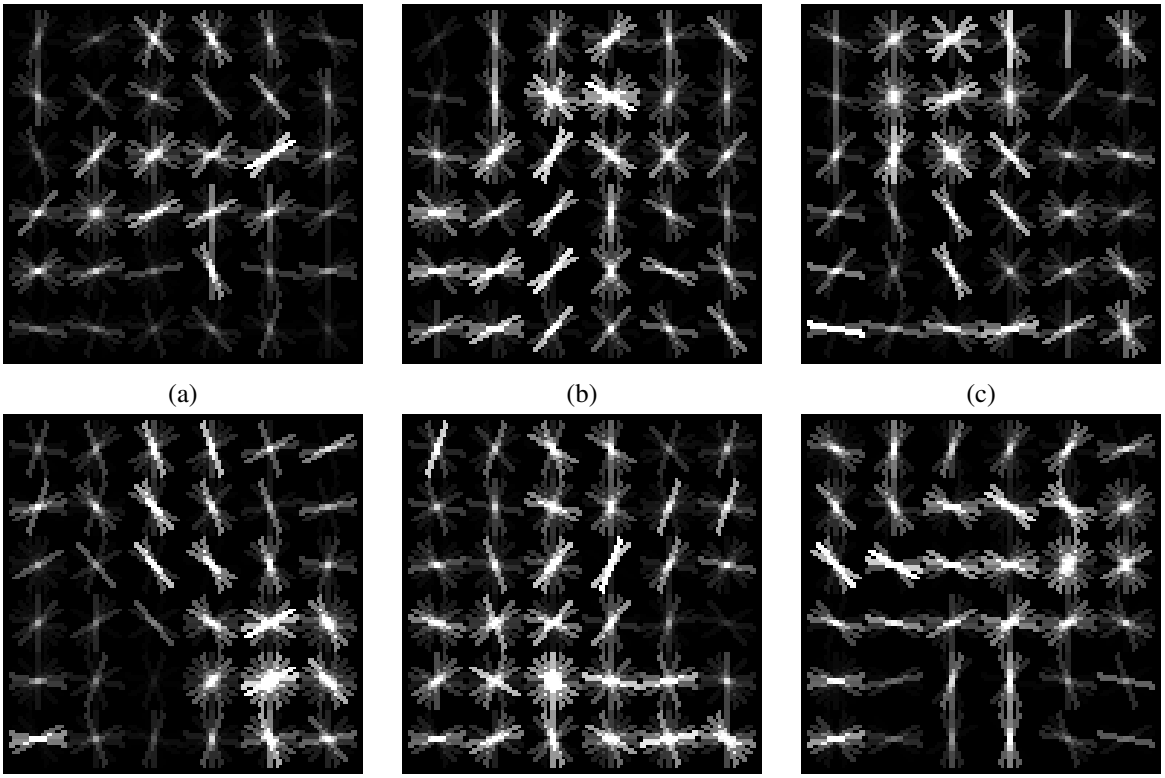


Fig. 6.6 Learned HOG filters for all part types of lower right arm.

6.6 Inference

Inference of the optimal pose pair requires maximizing $S(I^A, I^B, p^A, p^B, t^A, t^B)$ from equation (6.2) over both poses p_i^A and p_i^B and over the full set of latent variables t_i^A and t_i^B for views A and B. Whereas the graph $G = (V, E)$ for the single-view problem, which is the graph underlying equation (6.1) is a tree and can be solved exactly, the graph of the multi-view problem contains cycles. This can be seen easily, as it is constructed as a union of two identical trees with additional edges between corresponding nodes which are due to the consistency terms. This multi-view graph is illustrated in figure 6.7 where blue nodes are the body parts, black edges are the relations between the parts in single-view variable space, and red edges are relations between the viewpoints that are governed by the consistency parameters.

In the single-view model, the maximization task is carried out efficiently, similar to the well known dynamic programming technique described in [78]. Basically, it is an optimization method where score of each node of the tree is expressed in terms of sum of its individual score and scores of its children. The objective here is to compute scores for every possible parent location for every possible child location in the discrete space. This takes exponential time for each part. Instead, the idea is to establish a message passing scheme where score of a child is transmitted to its parent via a computed message. This message contains several terms, such as the score of the best scoring child and relative position of the child to the parent that is being computed. The computation of the spatial term can be performed with distance transform [78], delivering further efficiency to the global optimization task. The complete optimization

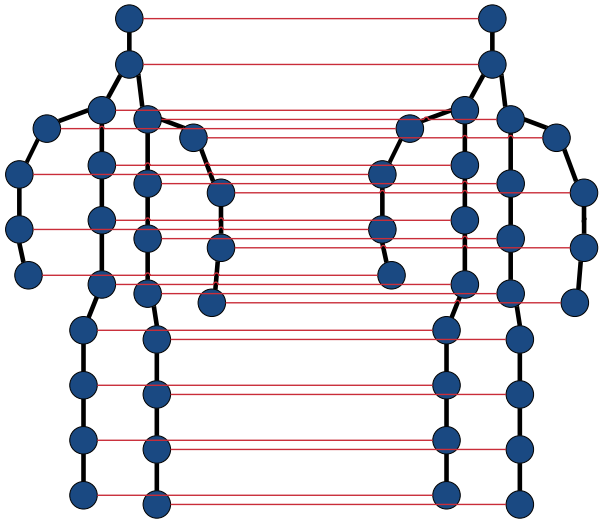


Fig. 6.7 Illustration of the multi-view model for two subjects of 26 parts. Vertices are body parts, black edges are single-view relations, red edges are the multi-view relations that are governed by consistency parameters.

in the single-view model can be summed up as follows: The score of a leaf node is calculated for every possible location and part type, and the location - part type pair that yields the maximum score is used to compute the distance transform map. Then the bias terms are added to obtain the final value of the message. Moving up through the tree, the score of the parent node is simply a sum of its individual score (filter response in our case) and the sum of the messages that it receives from its children. This process is repeated until the root node is reached, which yields root scores for every locations and part types. For every root candidate that is above a threshold, one can obtain a configuration of parts that can be retrieved by tracking back the *arg* max indices of all nodes. Using non-maxima suppression, it can be deduced whether the maximized configurations belong to the same person or multiple people.

Unlike the single-view problem, the maximization cannot be carried out exactly and efficiently with dynamic programming anymore since the graph is not a tree but a loopy graph. Several strategies are possible to maximize the multi-view scoring equation (6.2): approximative message passing (loopy belief propagation) is applicable for instance, which jointly optimizes the full set of variables in an approximative way, starting from an initialization. We instead chose an iterative scheme which calculates the *exact* solution for a subset of variables keeping the other variables fixed, and then alternates. In particular, as shown in figure 6.1, we optimize for a given view while keeping the variables of the other view (the "support view") fixed. Removing an entire view from the optimization space ensures that the graph over the remaining variables is restricted to a tree, which allows to solve the sub-problem efficiently using dynamic programming as described in the previous paragraph.

In order to formalize the multi-view inference, let us write $kids(i)$ for the child nodes of part i. The score of a part location p_i for a given part type t_i and the message that part i passes to its parent j is

computed as follows:

$$\begin{aligned}\text{score}_i(t_i^A, t_i^B, p_i^A, p_i^B) = & \; b_i^{t_i^A} + w_i^{t_i} \cdot \phi(I^A, p_i^A) \\ & + b_i^{t_i^B} + w_i^{t_i} \cdot \phi(I^B, p_i^B) \\ & + \alpha\xi(p_i^A, p_i^B) + \beta\lambda(t_i^A, t_i^B) \\ & + \sum_{k \in kids(i)} m_k(t_i^A, t_i^B, p_i^A, p_i^B)\end{aligned} \tag{6.8}$$

with

$$\begin{aligned}m_i(t_j^A, t_j^B, p_j^A, p_j^B) = \max_{t_i^A, t_i^B} \Big[& b_{ij}^{t_i^A, t_j^A} + b_{ij}^{t_i^B, t_j^B} \\ & + \max_{p_i^A, p_i^B} \text{score}_i(t_i^A, t_i^B, p_i^A, p_i^B) \\ & + w_{ij}^{t_i^A, t_j^A} \cdot \psi(p_i^A, p_j^A) + w_{ij}^{t_i^B, t_j^B} \cdot \psi(p_i^B, p_j^B) \Big]\end{aligned} \tag{6.9}$$

As mentioned, one of the two sets A and B is kept constant at each iteration. Messages from all children of part i are collected and summed with the bias term and filter response, resulting in the score for that pixel position and mixture pair. As classically done in deformable parts based models, the inner maximization in 6.9 can be carried efficiently with min convolutions (a distance transform, see [79]).

The algorithm is initialized through by solving the single-view problem independently for each viewpoint. The viewpoint with the highest scoring pose is chosen as initial support pose, the other pose of the other viewpoint being optimized in the first iteration. Afterwards, the optimization result is set as the new support pose and the initial support pose –which is now the target pose– is optimized again with this new support pose. The iterative process is repeated on until convergence or a maximum number of iterations is reached. Optimizing each sub-problem is a classical approach, where the message passing scheme iterates from the leaf nodes to the root node. After thresholding to eliminate weak candidates and non-maximum suppression to discard similar ones, backtracking obtains the final pose in each viewpoint.

In the next chapter, first two public datasets are presented. Then, training and evaluation of the proposed method are detailed and the results are given with respect to relevant metrics. Additionally, implementation details are disclosed for faster inference.

Chapter 7

Experiments

7.1 Introduction and datasets

In this chapter, we evaluate the method proposed in Chapter 6 and aim to demonstrate its capabilities on different image datasets. For this assessment to be as general and broad as possible, two datasets were selected. These two sets, which will be presented in detail in the following sections, provide different people as subjects, various number of visual sensors and assorted activities which in some cases include interactions with surrounding objects.

After the presentation of datasets, details concerning the training phase are given in Section 7.2. Additionally, methods for assessment and chosen evaluation metrics are explained. Thereafter, the results of experiments on the presented data are disclosed in Section 7.3 and the implementation details are reported in Section 7.4. Finally all findings are analyzed and discussed in Section 7.5.

7.1.1 HumanEva

First, we evaluated our work on the well known *HumanEva I Dataset* introduced in [261] and its technical details are revealed in [263]. This large dataset has been shot using seven calibrated cameras, their positioning as well as sample images acquired at a certain time are depicted in figure 7.1. There are four black and white cameras (abbreviated as BW1 through BW4) and three color cameras (abbreviated as C1 through C3), which provide images of size 640×480 and 684×484 pixels recorded in 60 frames per second, respectively. Calibration data is provided as plain text files and images are available as compressed video (AVI) files. Ground truth data is recorded using an industrial motion capture system, called *ViconPeak*[1] and 20 markers. Although the authors have provided an extensive MATLAB code to demonstrate the dataset and access the ground truth joint location data, the underlying third party libraries to extract frames from compressed videos are quite outdated and not straightforward to run in Linux systems. Therefore, we opted for using the third party image converting tool called *FFmpeg*[2] to extract images with the indicated frame rate. Then the provided MATLAB code was modified to use the images that we extracted instead of the video files, since this code manages the synchronization between

[1] Details can be found on `http://www.vicon.com/`, last accessed on 01/07/2016

[2] Accessible at `http://ffmpeg.org/`, last accessed on 01/07/2016.

the image stream and the motion capture stream. As a side note, we also extracted the ground truth data and recorded the position of each joint in each frame in an arbitrarily formatted comma separated values (CSV) files for further use.

Another detail is that background images are provided with the dataset, but our method does not need these images since we do not rely on background subtraction or any other localization technique. As described thoroughly in Chapter 6, our method takes a global approach to search the entire image space. The only occasion that we used those background images is in training as negative samples, as described in Section 6.5.

In our experiments we only used color cameras. As seen in the Fig. 7.1, C1 records the subject facing forward while C2 and C3 are located at right angles. It should be noted that our method does not necessarily require RGB images since we use gradient features, whose details are explained further in Section 7.2. Therefore our choice of the camera subset is not based on camera type, but camera position: We think that three camera with right angles, with one of them directed from the front is a sufficient setup to demonstrate our capabilities and difficult enough challenge our approach. It should also be noted that in our setup two cameras, namely C2 and C3, are pointed to each other which causes an exceptional situation where the *epipoles* reside within the 2D images (see Section 4.4 for further details.). In this case the epipolar lines intersect within the image, which induces a further challenge for multi-view pose estimation task.

There are four actors, and each one performs different activities such as walking, boxing, jogging, gestures and throw-catch at a time. Ground truth joint locations were recorded with a motion capture system, with 20 joints. There are three takes for each sequence, with the following configuration for the first three subjects: First trial is divided into training and validation subsets, second trial is for test but the joint information is withheld, third trial is also for training but the video is withheld, only motion capture is shared. Fourth subject is purely for testing, meaning that video is available but not the ground truth. It should also be noted that subjects are varied in appearance, composition of the actors consists of one female and three males, with different clothing and various body properties.

7.1.2 UMPM

As a secondary dataset, *Utrecht Multi-Person Motion (UMPM) benchmark* [303] was selected to further evaluate our method and its technical details are available in [304]. UMPM dataset has been shot with four cameras that are located in various heights and in a manner that no camera can be seen in the scene by other cameras. The placement of cameras and their field of views can be observed in Fig. 7.2.

There are four color cameras (abbreviated as C1 through C4), which provide images of size 644×484 pixels recorded at 50 frames per second. Images are provided in uncompressed AVI format and offered without any post-processing, therefore we transformed them to PNG format, using *FFmpeg*. Calibration data is provided in C3D file format and the authors offer sample C++ code snippets and hyperlinks to relevant libraries in the project site[3]. Similarly to HumanEva, ground truth data is recorded using a *Vicon* system, but with 15 markers and at a rate of 100 frames per second.

[3]C3D helpers page on project site `http://www.projects.science.uu.nl/umpm/c3d.html`, last accessed on 10/03/2017.

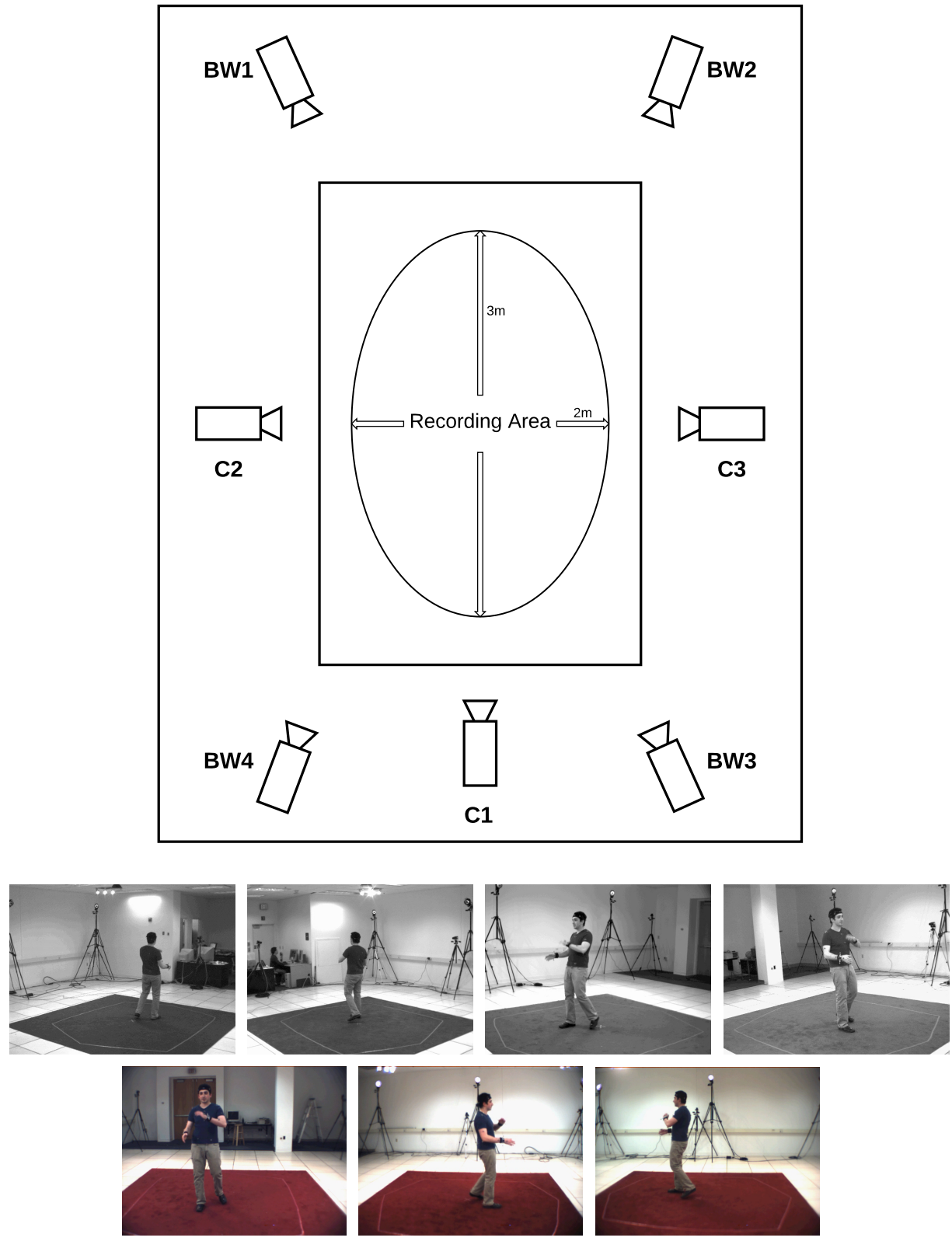


Fig. 7.1 *Top:* Placement of the seven calibrated cameras as described in [263]. C1 through C3 are color cameras, BW1 through BW4 are black and white cameras. Subject is always recorded in the designated area. *Bottom:* Acquired images from these cameras.

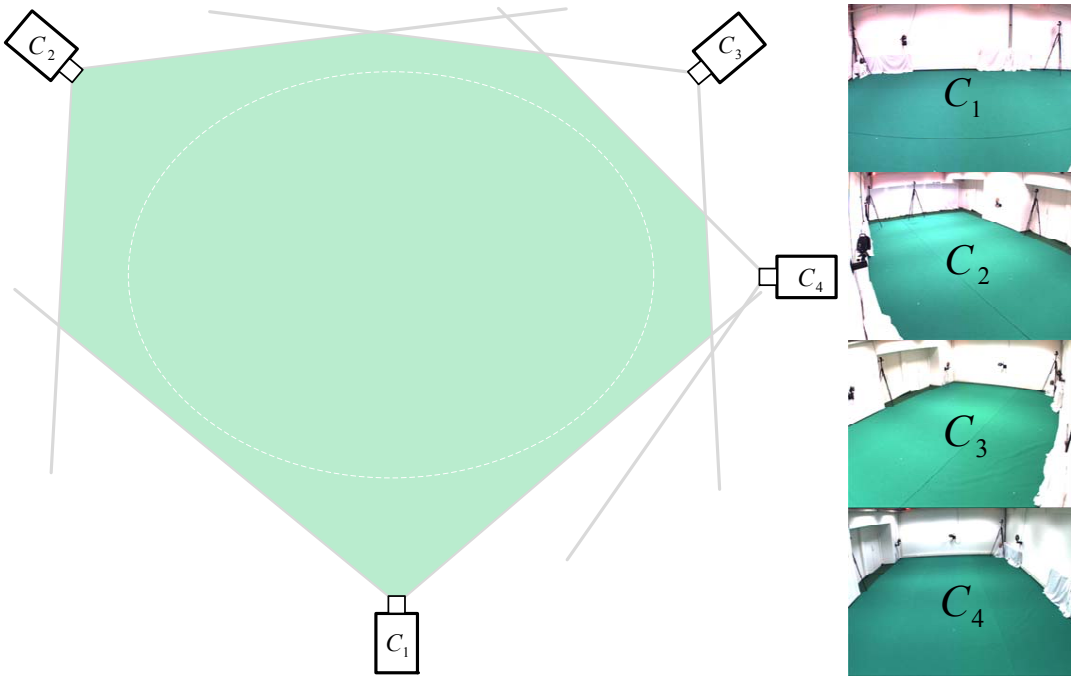


Fig. 7.2 *Left:* Placement of four color cameras, C1 through C4. *Right:* Corresponding field of views. Reprinted from [304]

Various types of actions are offered in the dataset and these are performed by an individual or a group of individuals. Among actions are *triangle* where subjects are walking following a triangle, *chair* where subjects sit down and stand up, *table* where subjects walk and lean against and lie on table, *grab* where subjects grab various objects from a table, *orthosyn* where subjects are performing predefined gestures, etc. Other actions are also available, but only provided for multiple-subject scenarios. Following [241], we conducted our experiments on video sequences with only one subject. Nonetheless, 52856 distinct images that are captured from four cameras were used for training, validation and test purposes.

Interaction with objects, especially unusual activities such as leaning or lying on a table suggest this dataset is arguably more difficult than HumanEva in terms of pose estimation. Therefore, we believe that using UMPM along with HumanEva is an adequate evaluation approach that appropriately challenges our proposed method.

7.2 Training and Evaluation

For HumanEva set there are three takes for each sequence, used for training, validation and test. Since the creators of HumanEva favor online evaluation, the original test set does not contain ground truth joint positions. We divided the original training set into training and validation sets and used the original validation set for testing purposes, as it is done in [6] for walking and box actions. But we consider jogging and gestures actions as well. Specifically, there are a little less than 7000 instances in the original training set for three subjects which sums up to almost 21000 images from three viewpoints. To perform

the training, we took only 100 frames of the first subject (S1) only with equal time intervals for every activity from three cameras, which sums up to 1500 images and the remainder of the data is assigned as our validation set. The ratio of training / validation seems a little low, but considering the single-view model were originally trained on only 100 images of the *PARSE* dataset [214], we find it sufficient. All hyper-parameters have been optimized over our validation set.

For UMPM set, we only considered all available sequences with one subject (P1), which includes interactions with objects such as sitting on a chair, picking up a small object, leaning and lying on a table. The training, validation and test partitions were manually divided using 60%, 20% and 20% of the all available data, respectively. While dividing the partitions, we carefully considered the rightful division of activities; i.e. each partition contains at least a full occurrence of each activity. In terms of numbers, the HumanEva test set consists of 4493 images per camera, while UMPM test set has 6074 images per camera. The number of distinct images used in the tests sums up to 13479 and 24296, respectively.

The data attached terms $\phi(\cdot,\cdot)$ in this work were based on HOG features from [55]. Specifically, the features that we are using are intensity features and in our implementation only the channel with the strongest gradient is considered. Other features are possible, in particular learned deep feature extractors as in [46] or [178]. This does not change the setup, and in the context of a low amount of training data, deep learning does not necessarily provide any advantage.

Parameters of the single-view model (Equation 6.1) are learned on all activities of first subject, S1. As stated before, we used 1500 positive samples and as for the negative samples, background images from *HumanEva-I* and *UMPM* were used in addition to the *INRIAPerson Database* from [55]. The remainder of the data was set as the validation set and used to learn hyper-parameters α and β, equation (6.2). All hyper-parameters have been optimized over our validation set.

There is a shift in data definition between two datasets we used: HumanEva is recorded with 20 joints, whereas UMPM is recorded with 15 joints. Since our model is trained with 26 parts, we used linear functions to convert 20 joint locations to box centers for training purposes for HumanEva set, and transform back the box centers to the 20 joint locations for evaluation. The correspondence is given in Table 7.1, where first two columns are the model examples for HumanEva (HE) and single-view flexible mixture of parts (FMP) model. It is stated in [263] that the following joint locations are considered the same for tree structured models: 1&19, 4&5, 8&9, 12&13 and 16&17. The third column is the conversion from HE to FMP for converting the ground truth data for training. Please note that the 2^{nd} joint is assigned to the middle point of the 10^{th} and the 22^{th} box centers. The fourth column is the inverse transformation which is used in evaluation. The final column (marked *ratio*) is the transformation detail which indicates the position of the FMP box center with respect to HE joint locations. For example, 8^{th} box of the FMP is between the 15^{th} and 7^{th} joints, closer to the 7^{th} joint by 1/3 of the distance between 15^{th} and 7^{th}. It should be noted that we employed a very similar linear transformation for UMPM dataset as well, where provided 15 joint locations are translated to 26 box centers and vice versa.

The parameters of the single-view model (Eq. 6.1) are learned on all activities of first subject S1 for HumanEva. We took 100 frames with equal time intervals for every activity from three cameras for training, which sums up to 1500 images. The remainder of the data was set as the validation set. For UMPM, nearly 400 consecutive frames for each sequence were used as positive samples. As for the

Table 7.1 Conversion between the 20 joint HumanEva (HE) skeleton model and 26 joint FMP model, in two direction. '10&22' signifies the middle of these points in the first conversion. In the second conversion, ratio indicates the proximity to the second box center. See text for detailed explanation.

HE	FMP	HE to FMP		FMP to HE		(ratio)
		1&19	2	1	20	
		2	10&22	2	1	
		3	15	3	7	
		4&5	17	4	8&7	1/10
		6	19	5	8	
		7	3	6	8&10	1/2
		8&9	5	7	10	
		10	7	8	15&7	1/3
		11	22	9	15&7	2/3
		12&13	24	10	15	
		14	26	11	16&15	1/5
		15	10	12	16	
		16&17	12	13	16&18	1/2
		18	14	14	18	
		20	1	15	3	
				16	4&3	1/10
				17	4	
				18	4&6	1/2
				19	6	
				20	11&3	1/3
				21	11&3	2/3
				22	11	
				23	12&11	1/5
				24	12	
				25	12&14	1/2
				26	14	

negative samples, background images from corresponding datasets were used in addition to the *INRIA Person Database* [55]. Hyper-parameters α and β of equation (6.2) were learned on validation sets.

To learn the weights of the error estimating convolutional neural net $\sigma_i(\cdot)$, training data sets were augmented with horizontal flip, Gaussian blur and additive noise. As mentioned earlier, we used a fine-tuned version of VGG-16 [269] model using pre-trained weights on *ImageNet* to estimate the part based error of the single-view pose. We removed all the top fully connected layers and introduced our own top model with a hidden layer of 1024 nodes, an output layer of K nodes and parametric ReLU (PReLU) as non-linearity. Architecture specifics are depicted in Fig. 7.3, along with output dimensions for each convolutional block. To fine-tune the VGG-16 network, we followed a two-stage approach. First, weights of the complete VGG-16 network were frozen so that they are unaffected by the backpropagation and weights of the top model were roughly learned with a high learning rate. Then as a second stage, the top model were initialized with these weights, and the last convolutional blocks (namely the last three *conv3-512* layers) were unfrozen for fine-tuning. We preferred stochastic gradient descent as optimiza-

tion algorithm with small learning rate to ensure that the weights of the last convolutional block are marginally updated. To prevent overfitting to augmented data sets we applied strong regularization and also employed *Dropout* [277] with probability of 0.5. In terms of training time, 20 epochs of top-model training and 100 epochs of fine-tuning yielded satisfactory results.

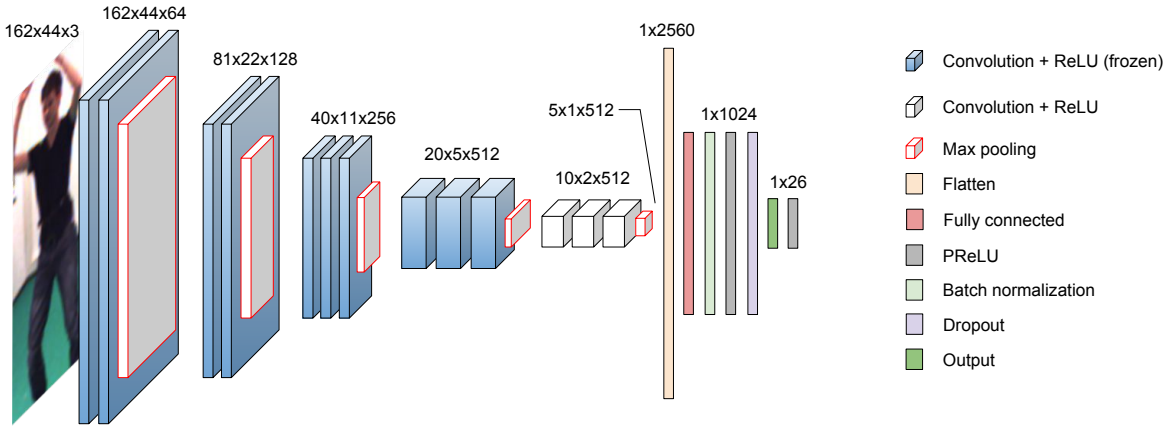


Fig. 7.3 Fine-tuned version of VGG-16 and along with our top model. Dimensions of layer outputs are indicated above each block. Each max pooling layer halves the output dimensions. First convolutional block consists of 3x3x64 convolutions, second one consists of 3x3x128 convolutions, third and fourth ones consist of 3x3x512 convolutions. Note that frozen layers are not updated during backpropagation. (Zero-padding layers are not shown for simplicity.)

For each multi-view arrangement, i.e. pair combinations of available cameras, two pose estimations are produced. Since each view belongs to several multi-view arrangements, we end up with several pose candidates for the same viewpoint, e.g. we obtain two pose candidates for C1, one from the C1-C2 pair and one from the C3-C1 pair. These candidates are simply averaged and the obtained 2D poses are triangulated non-linearly to obtain 3D pose for a single time frame. Following the literature on 3D pose estimation [37, 241] we use the percentage of correctly detected parts (PCP) in 3D, which is calculated as

$$\frac{\|\hat{s}_n - s_n\| + \|\hat{e}_n - e_n\|}{2} \leq \gamma \|\hat{s}_n - \hat{e}_n\| \tag{7.1}$$

where s_n and e_n are the estimated start and end 3D coordinates of the n'th part segment, and $\hat{s}_n$ and $\hat{e}_n$ are the ground truth 3D coordinates for the same part segment. By convention we take $\gamma = 0.5$ in all our computations, unless specified otherwise.

7.3 Results

In this section, results of the experiments are presented to demonstrate the capabilities of the proposed method. First, we evaluate our multi-view approach against the single-view method given in [341]. Table 7.2 shows 3D PCP scores on train subject S1 only and over all subjects; while table 7.3 shows 3D PCP scores on UMPM test set. We provide three versions of our method: geometric constraints only,

Table 7.2 HumanEva – PCP 3D scores of our model trained on subject 1, evaluated on subject 1 and all subjects combined, with PCP threshold 0.5. Performance is compared to Flexible Mixture of Parts (FMP) [341] method.

Test set	Sequence	FMP [341]	Ours		
			Geometric	+Appearance	+Adaptive
S1	Box	77.34	82.70	83.87	85.31
S1, S2, S3	Box	67.14	69.45	70.23	71.57
S1	Gestures	78.91	84.27	84.08	88.14
S1, S2, S3	Gestures	74.68	77.38	78.81	80.34
S1	Jog	84.91	86.75	86.70	86.86
S1, S2, S3	Jog	77.52	80.16	79.84	80.97
S1	Walking	84.65	86.71	86.50	87.68
S1, S2, S3	Walking	78.49	81.69	81.96	83.17
S1	Overall	82.02	85.43	85.49	87.24
S1, S2, S3	Overall	74.86	77.62	78.11	79.40

geometric and appearance constraints combined, and both constraints with adaptive viewpoint selection. It is clear that in all cases and both data sets, the multi-view scheme significantly improves performance. Depending on the performed action, gains can be significant up to **9.2%** in HumanEva and **10.1%** in UMPM. The last columns of tables 7.2 and 7.3 show that the additional coherence terms and their adaptive control further decrease the error. Fig. 7.5 demonstrates that this error is distributed over all different parts of the body: we improve most on wrists and elbows, which are important joints for gesture and activity recognition, as seen in table 7.4. Plots for overall PCP 3D curves with respect to various PCP thresholds are also given in Fig. 7.6.

Fig. 7.7 depicts 4 histograms where each one depicts the error difference between single-view and multi-view methods. Negative difference signifies that multi-view error produced a pose with a higher error, thus a 'deterioration' of the single-view pose. Correspondingly, a positive difference signifies an improvement over the single-view method. The top row histograms are calculated on HumanEva, while bottom row ones are calculated on UMPM. On the left column, methods were executed *without* the adaptive viewpoint selection, while on the right column they were executed with the adaptive viewpoint selection. It is clearly evident that the number of deteriorations (red parts of the histograms) are substantially lower with the adaptive viewpoint selection scheme. This can be interpreted as 'preventions' of badly detected parts over views. However, these preventions are not always directly translated into 'improvements', but instead remain exactly the same as the single-view part estimations. Nevertheless, it is apparent that the adaptive viewpoint selection helps with the overall performance of the proposed method.

Fig. 7.4 depicts intermediate poses and epipolar lines throughout the course of algorithm while Fig. 7.8 shows several examples from the test set where faulty poses are corrected with the multi-view approach. Note that limbs are in particular subject to correction by geometrical and appearance based constraints, since they are considerably susceptible to be mistaken for their respective counterpart. Some extreme cases are indiscernible in the tables. We achieve an improvement rate of 55% on 390 frames of the *Gestures* action performed by S1 and evaluated on C3 with support of C2. Similarly, improvement

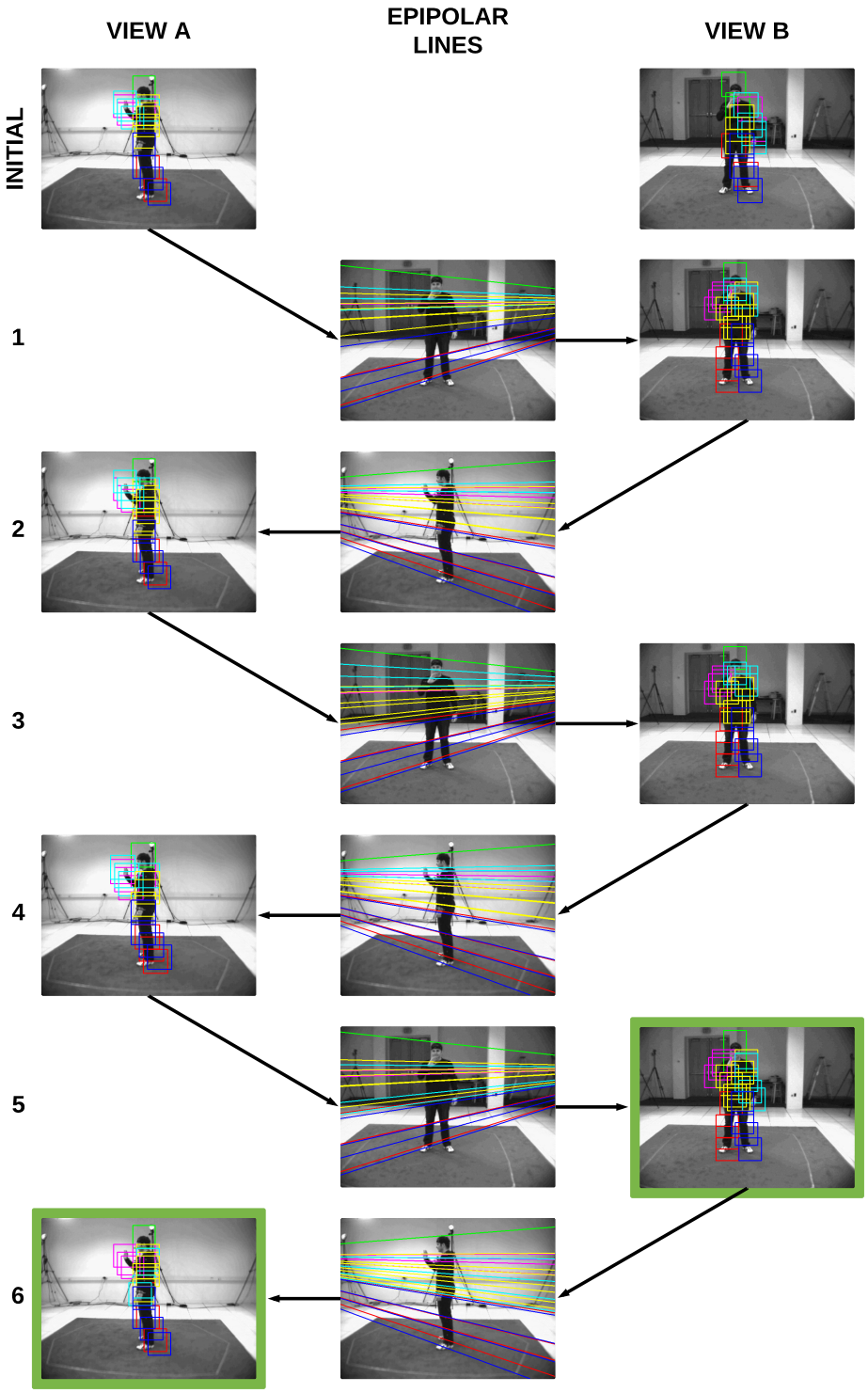


Fig. 7.4 Illustration of the iterative optimization process. The first and last columns are two respective viewpoints, the middle column shows epipolar lines overlaid over the respective viewpoint. Diagonal arrows show the pose that the epipolar lines are based on. Each row is an iteration and horizontal arrows shows the resulting pose and epipolar lines used in joint estimation. Final poses are marked with green borders.

Table 7.3 UMPM – PCP 3D scores on all sequences with PCP threshold 0.5, compared to Flexible Mixture of Parts (FMP) [341] method.

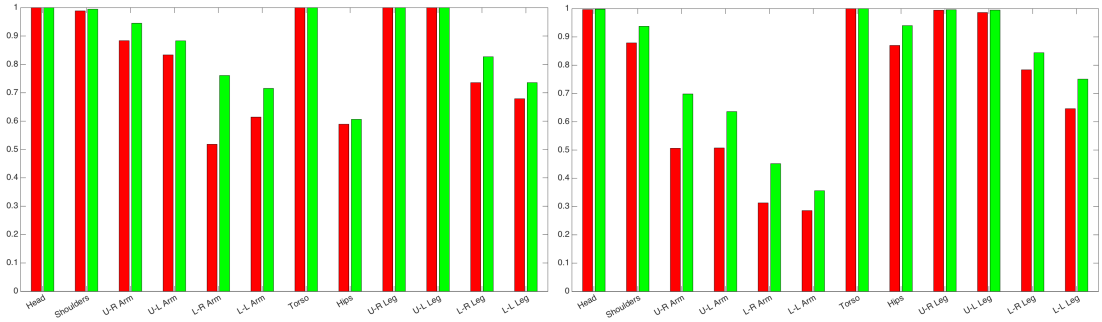
Sequence	FMP [341]	Ours		
		Geometric	+Appearance	+Adaptive
Chair	74.72	78.09	77.54	79.94
Grab	74.23	76.25	77.18	81.92
Orthosyn	72.47	74.65	75.22	76.48
Table	70.30	73.49	74.18	77.86
Triangle	73.69	77.26	77.81	83.81
Overall	73.07	75.91	76.37	80.04

Table 7.4 PCP 3D scores (%) for all limb parts with PCP threshold 0.5, compared to FMP[341] on UMPM and HumanEva datasets.(*U-L: upper left, U-R: upper right, L-L: lower left, L-R: lower right)*

Configuration	U-R Arm	U-L Arm	L-R Arm	L-L Arm	U-R Leg	U-L Leg	L-R Leg	L-L Leg
FMP[341] on HumanEva	88.4	83.4	51.8	61.4	100	100	73.6	67.9
Ours on HumanEva	94.5	88.3	76.1	71.5	100	100	82.7	73.6
FMP[341] on UMPM	50.6	50.7	31.3	28.6	99.4	98.6	78.4	64.6
Ours on UMPM	69.8	63.6	45.2	35.6	99.6	99.5	84.4	75.1

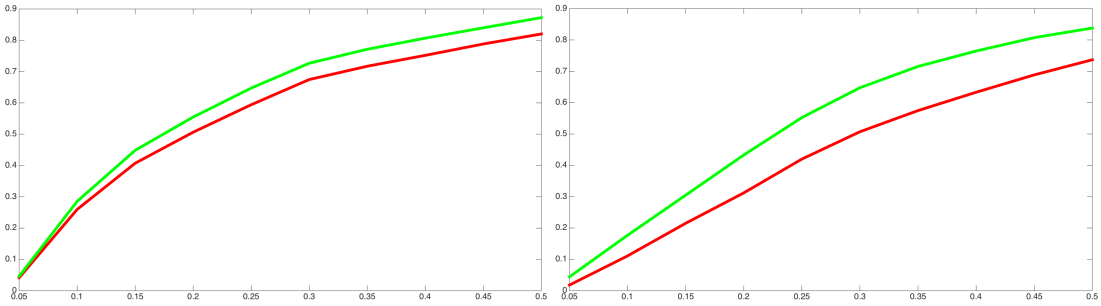
rate is above 27% on 416 frames of *Gestures* performed by S2 that and evaluated on C1 with support of C3. On the other hand, a few cases may lead to deterioration of one the two views.

We also compare our work to Schick et al.'s voxel carving based 3D pictorial structure method [241], apart from the original FMP [341]. For HumanEva dataset and for all sequences of S1 and S2, they report 78% PCP score, and for UMPM dataset for all sequences of P1, they report 75% of PCP score. For the same settings, our PCP scores are **83.42%** and **80.04%**, respectively.



(a) HumanEva, subject 1, all sequences (b) UMPM, all sequences

Fig. 7.5 PCP 3D scores for individual parts obtained by FMP[341] (red) and ours (green) on both datasets. (*U-L: upper left, U-R: upper right, L-L: lower left, L-R: lower right)*



(a) HumanEva, subject 1, all sequences (b) UMPM, all sequences

Fig. 7.6 PCP 3D curves as a function of threshold γ from Eq. 7.1, obtained by FMP[341] (red) and ours (green) on both datasets.

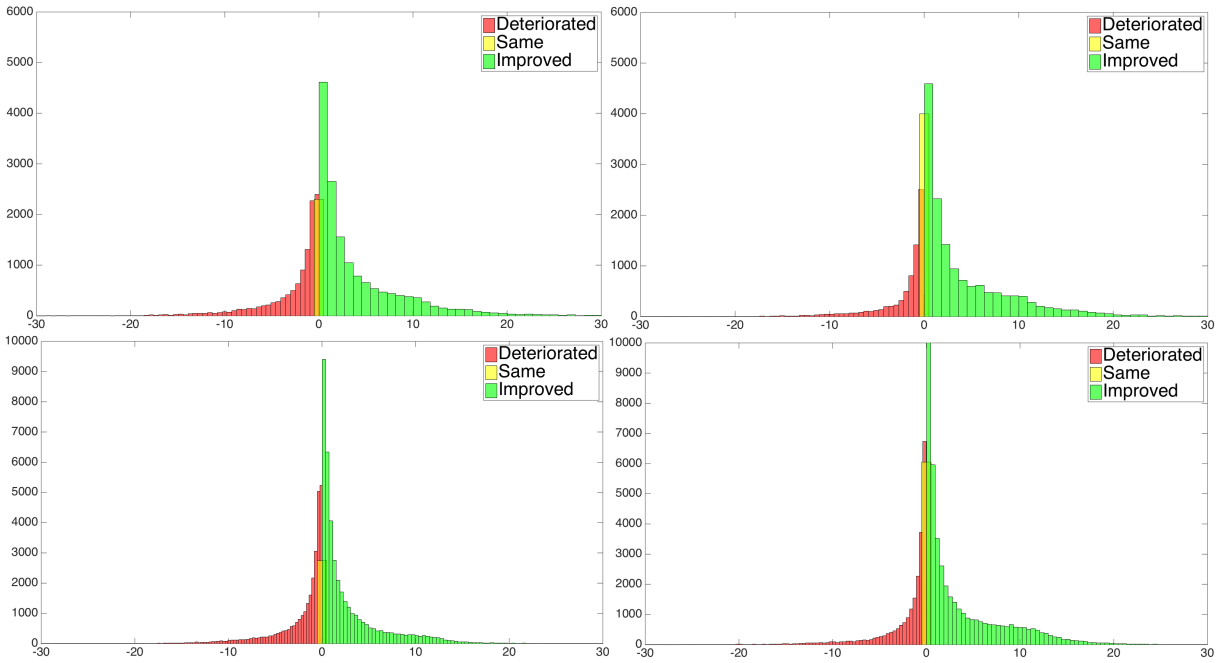


Fig. 7.7 Breakdown of differences of errors for each part, compared to FMP[341]. Negative differences (red) indicates cases where our method performs worse than FMP, zero difference (yellow) indicates same poses were estimated and positive difference (green) indicates our method yielded a better pose. *Top row:* HumanEva, *bottom row:* UMPM, *left column:* without adaptive viewpoint selection, *right column:* with adaptive viewpoint selection.

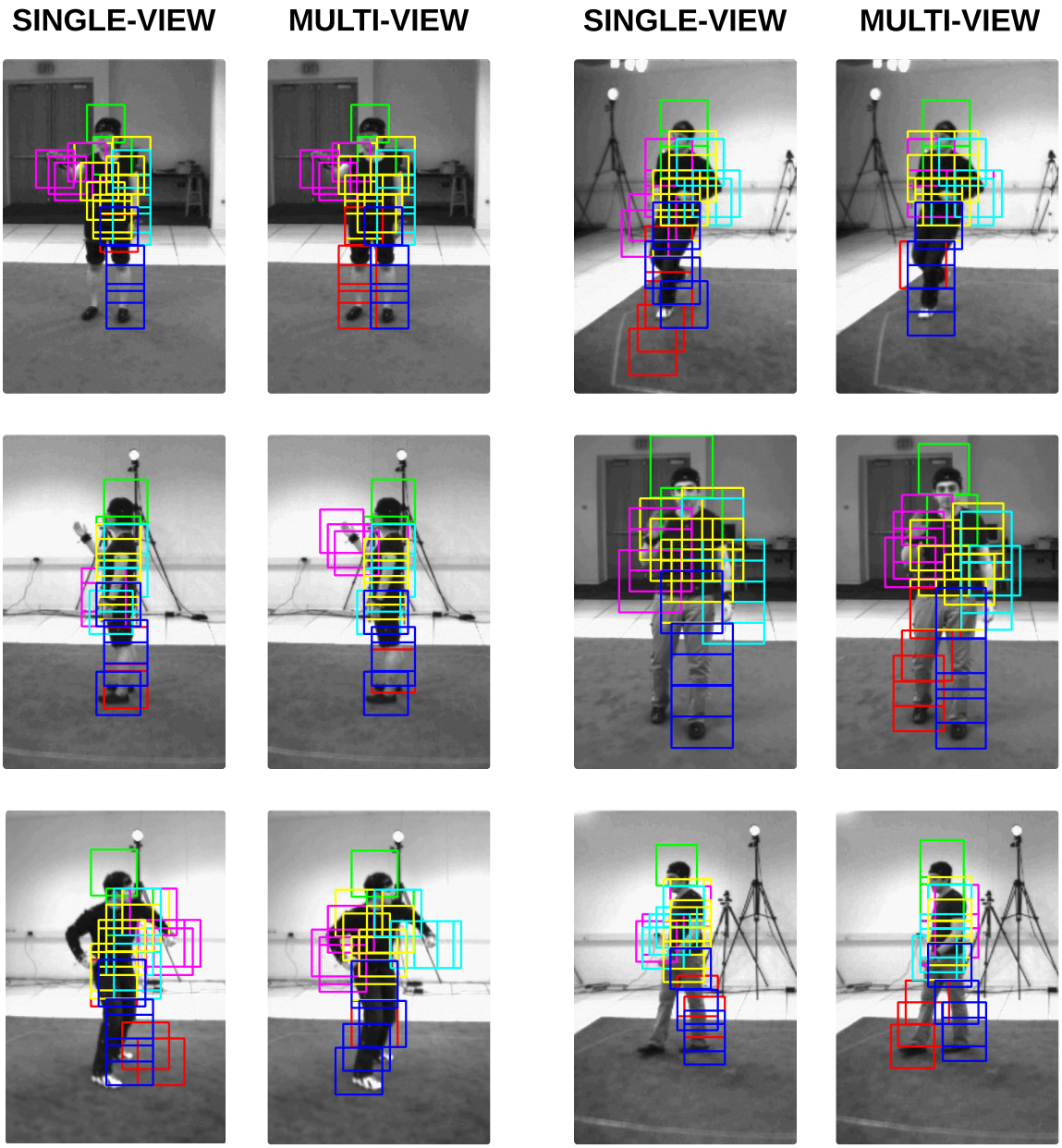


Fig. 7.8 Qualitative comparison of all three subjects performing various activities from different viewpoints. First and third columns: poses obtained with the single-view model. Second and fourth columns: poses obtained with multi-view pose estimation.

7.4 Fast implementation

Before implementing the multi-view code, we first investigated the single-view code in details, which is publicly available[4]. The code is written in Matlab, and takes 3 seconds for small images (150×200 pixels) and takes up to 30 seconds for full VGA images on an ordinary personal computer. Our final goal is to achieve 500ms for each image with a good accuracy level.

Reducing the search space. As explained in Section 6.2, execution time increases exponentially with respect to the image space, or pixel locations in other words. To that end, we experimented with some pre-processing techniques to overcome the burden of large search space in order to end up with a smaller region of interest for executing our single-view pose estimation algorithm. Most practical and appropriate technique to determine a region of interest that would surround a person in an image would be the popular HOG for human detection[55], considering that its implementation is available in OpenCV[5] for a long time. It works fast on CPU (and would be considerably faster on GPU), specifically 135ms for 640×480 images and 30ms for half the size, which could leverage the execution time for single-view pose estimation. But reducing the search space with a people detector means that we heavily rely on the accuracy of region interest, which might be problematic in some cases. To assess whether this approach is reliable or not, we conveyed a small experiment simultaneous to the preliminary experiments described in Chapter 5. We used 258 samples from all cases, using samples three different distances which means different area occupancy of a person in terms of pixels. Furthermore, since our method is an eventually multi-view method which aims to overcome occlusions, samples with different occlusion settings were tested. We executed the built-in people detector in OpenCV with a pre-trained model on different scales. Experiment results were not promising. For 101 images, people detector did not found the subject in the image. Among 157 detections, 33 were false detections and 30 were partially correct. 94 detections were correct, which concludes the accuracy of this test to **36%**. It should be noted that false detections and partial detections were mostly for samples that are in close setting (2-3 meters from the camera) and in the presence of occlusion. Needless to say, we abandoned the idea of determining the region of interest with a people detector method, and carried on to other ways of decreasing the execution time of single-view pose estimation algorithm.

Another way to speed up the single-view algorithm is using less levels in the feature pyramid, which in fact means searching the image spaces in less different scales for a person. This is obviously a trade-off between the speed and generalizability of the algorithm, since lesser pyramid levels mean lesser flexibility for the distance of the person to the camera. Although we hacked the model and decreased the interval levels in development stages for fast prototyping, we always used full pyramid intervals in the reported experiments.

Code optimization. Finally, we decided to port the Matlab/C++ code to pure C++ for both single-view and multi-view pose estimation algorithms. Our multi-view implementation is based on our port of the Matlab/C++code from the single-view method by [341] to 100% pure C++, where crucial parts have also been ported to GPU processing using NVIDIA's CUDA library. This sped up runtime from 3000ms

[4] Available at `http://www.ics.uci.edu/~dramanan/software/pose/`, last accessed on 17/07/2016.
[5] Available at `http://opencv.org/`, last accessed on 15/07/2016.

per frame to 880ms per frame on a computer equipped with a 2.4Ghz Xeon E5-2609 processor and an NVIDIA 780 Ti GPU for the single-view algorithm (runtime given for a 172×224 image with 32 levels of down-sampling). The multi-view algorithm is slower, because 5.73 iterations are performed in average before the results are stable. We are currently working on additional optimizations of computational complexity using approximative parallel implementations of the distance transform on GPUs.

7.5 Conclusion

In this part of the thesis, our goal was to propose a pose estimation method that would benefit from the information obtained from multiple viewpoints. In Chapter 4 state-of-the-art was given to portray the current research activities and trending approaches in pose estimation task. After that, the procedure that helped us choose the appropriate were described in Chapter 5. These preliminary tests suggested that using global RGB method was the best way to build a multi-view pose estimation method. Therefore, we developed a such method that optimize coherence between the single-view pose estimations in an iterative way by exploiting geometric and appearance based constraints. Formal description of our method was given in details in Chapter 6.

In this chapter, we evaluated our multi-view method and compared to the single-view counterpart to demonstrate that pose estimation task can benefit from information obtained from multiple viewpoints. The results state that in some cases, such an approach improves the accuracy by up to 27.7%.

On the other hand, we argue that performance of our multi-view method depends on the errors of single-view FMP estimations. If both single-view estimations are erroneous, shared information between the views are unlikely to be practical to correct the initial poses. Or in an even worse case, those errors are likely to propagate between the views resulting in a further deteriorated pose at the end. However if one or both of the poses are accurate, the shared information between the views is more likely to be correct which culminates both pose estimations, which is supported by the results that we report in this chapter.

All things considered, in this part of the manuscript we proposed a multi-view articulated human pose estimation method and demonstrated its improvement compared to the single-view counterpart by tangible and concrete results.

Chapter 8

Conclusion

This last chapter constitutes a summary of efforts carried out within the scope of this thesis. First, the main contributions of this thesis will be recapitulated and the essential results will be outlined. Then, the limitations of the proposed methods will be discussed along with the potential improvements that can be made. Afterwards, possible future work will be introduced as well as prospective research goals. Finally, a list of scientific publications that are associated to this manuscript will be provided.

8.1 Summary of Contributions

In this thesis, we have dealt with the problems of viewpoint independence and multi-view observations of scenes, especially for human activity recognition and articulated pose estimation tasks, respectively. These two separate but related problems may in fact be consolidated under the broader notion of *geometrical flexibility for computer vision tasks*. We considered the questions of view invariance and improvements for multi-view settings individually, providing proposals for both of them.

For the first part, our goal was to establish a method where the location of the camera carries minimal or no impact on the performance of human activity recognition task. To fulfill this goal, we envisioned an intermediate representation of a video sequence, that can be view-invariant intrinsically. In other words, we aimed to establish an operation that takes a video sequence recorded from any view angle as input, and outputs a standardized representation. Ideally, the goal was to achieve the same representation for a particular action sequence that is for instance recorded from different viewpoints.

To that end, we first analyzed the action recognition literature and reviewed existing approaches toward representation of action sequences. We have been inspired by motion history images [28] and volume motion templates [227] to consequently propose a robust variant that is constructed from depth video sequences. The robustness of the proposed method lies in the additional operations such as noise reduction, not relying on background subtraction and outlier removal. Moreover, in order to build a view invariant model we applied viewpoint normalization, which is actually a rotation operation that transforms any volume motion template to a canonical orientation. Corresponding angles for the rotation are calculated with respect to the dominant motion direction of the observed activity, which is a characteristic property for most action types. Obtaining a robust and normalized volume object is not sufficient

to accomplish the classification task, since the volume object can still be considered as raw data. After investigating several spatio-temporal (3D) feature extraction techniques, we employed HOG3D features [135] to finally represent action sequences as a collection of feature vectors. Classification task was carried out with the classical bag-of-words approach. We evaluated our proposition on the LIRIS human activities dataset, which contains a medium amount of realistic and complex activities and obtained significant recognition performance gain compared to the baseline method.

As for the second part, our motivation was to establish a framework that can leverage image data acquired from different viewpoints and to use those in a collaborative manner to accomplish the pose estimation task efficiently and accurately. For this reason, we first studied and reviewed the state-of-the-art methods on articulated pose estimation. We evaluated the available methods based on various criteria such as input modalities, runtime requirements and particularly compatibility with multi-view extensions and conducted a preliminary research. According to the findings, flexible mixture of parts [340], which is basically a tree shaped graph representation of the human body, was considered the most suitable and adaptable one that fits our needs. We modeled the pose estimation problem with multiple graphs that are associated to each other by several constraints, and the inference problem was treated as a global optimization problem. We imposed strong prior on this problem, both by classical geometrical constraints and by the novel appearance constraints. Essentially, appearance constraints are modeled as latent variables which reward some appearance combinations (sets of appearance filters) for a part, while penalizing others. The inference problem, which consists of loopy graphs due to additional constraints between the viewpoints, is handled with an iterative scheme where parameters from all viewpoints are fixed except for one on each iteration. Evaluation of this method was completed on two publicly available datasets, namely HumanEva [261] and UMPM [303] and we determined that in both datasets our proposition provides 3-4% gain in the designated metrics, compared to the baseline method.

Additionally, we argued that such a method can yield even better pose estimation results if it is possible to evaluate the adequacy of individual viewpoints to support the consistency during the execution time. Therefore we were further motivated to investigate such a technique that will enable us control over the part based contributions of each viewpoint. Thus, we employed a convolutional neural network based approach to roughly estimate the expected error of a single-view estimation for each viewpoint. This allowed us to cancel out the multi-view constraints, i.e. geometric and appearance, in certain cases where a view angle provided poor visuals of a particular part. Consequently, we prevented propagation of erroneous data of poorly estimated part and improved the global pose estimation quality. This additional step, dubbed *adaptive viewpoint selection*, granted us an additional 2-4% of gain on both datasets and a total of 5-7% of improvement over the baseline method. Moreover, we compared our framework to another multi-view method and observed 5% of improvement on both datasets.

8.2 Discussion

In this section, we discuss the limitations of the proposed methods and identify the potential shortcomings under certain conditions.

Our first proposition takes depth imagery as input, acquired by commercial depth sensors. Even though they are affordable and made a great amount of research available widely, they have well-known limitations. Depth sensors can only operate in indoors and produce the best result under specific distances; acquired data quickly becomes unreliable once the conditions are no longer met. Furthermore, our method can only perform when the camera is stable and stationary. This is because the procedure to construct the volume motion template is highly sensitive and responsive on the any kind of movement in the scene, therefore the movement of the background with respect to moving camera will inevitably be translated into the point cloud. Currently, our method is not able to distinguish the ego-motion (the movement of the camera) from the motion of the individuals, as a consequence, we are limited to scenarios where the camera is absolutely immobile. This issue can be resolved by estimating the ego-motion from the video sequence and compensating it, as will be discussed in next section.

The second method that we proposed on the other hand, operates on RGB images and is not limited to indoors nor specific proximity requirements. However, it assumes calibrated cameras which may not be trivial in scenarios where the cameras are unstationary, such as when they are mounted on a mobile robot. Unlike the cases where cameras are pre-installed to an environment and are fixed, movable cameras requires to be calibrated before our method can perform. This is mainly because of the geometric constraints that we impose on the multi-view pose estimation scheme, where it is imperative to know the extrinsic parameters (position and direction of the cameras) in order to establish an association between cameras and their respective field of views. This can be achieved thorough self-calibration once the cameras are stable, even for a short period of time. Or, an alternative would be to perform calibration jointly with pose estimation by solving a joint (either discrete or continuous) optimization procedure.

This section drew attention to the limitations of the methods that we proposed in this manuscript and remarked the possible scenarios where they may be inadequate. In the following section, we will propose potential solutions corresponding to these limitations.

8.3 Future Work

We would like to point out potential research tracks to follow the ideas presented in this thesis and to carry the scientific findings one step further. Thus, this section will first propose possible future work to address the limitations that are mentioned above, and imagine new related challenges.

Addressing the Limitations

Here, we raise two propositions to address the issues that are discussed in Section 8.2. The first one focus on the action recognition part, where the procedure to build volume motion template can be improved for the cases where camera is mounted on a mobile robotic platform. As mentioned earlier, the movement of the camera causes an undesirable artifact, that is, the otherwise fixed background is recorded as evolving and advancing. Volume motion template representation requires to encode solely the motion of the person for best recognition results, and therefore we ought to compensate for this *movement of the background*. In order to accomplish this, the movement of the camera, also known as *ego-motion*, needs

to be determined first. There is a rather large body of work about estimating the ego-motion, which is reviewed and discussed in this survey [219]. Once the ego-motion is estimated, calculated volume motion template can be modified and adjusted according to obtained vector field, so that the movement of the camera is ignored.

Secondly, the assumption for the known calibration parameters ought to be avoided in order to expand the application areas of the multi-view pose estimation scheme. Let us consider a scenario where several cameras are mounted to corresponding mobile robots which are freely roaming in a bounded environment to temporarily stop and observe a person from various viewpoints. Each time the robots are stationary, the calibration parameters, particularly the extrinsic ones, ought to be recalculated so that the proposed geometric constraints are applicable. It is possible but ill-favored to rely on known coordinates of the robots (via mechanical encoders or other positioning techniques) as well as pitch, yaw, roll movements of the mounted cameras. Instead, these parameters can be estimated by purely image based methods, even for the cases that lack special markers such as chess-board patterns or calibration plates. Such parameter estimation can be carried out with a well known iterative scheme [101] based on automatic homography estimation between two images: First, numerous interest points are detected on both images, commonly using SIFT [161] or SURF [17] feature descriptors and putative correspondences are computed. Then, using Random Sample Consensus (RANSAC) [84] and some additional optimization steps the homography can be estimated. For alternative camera calibration methods, reviews and accuracy evaluations please refer to this survey [237]. Regardless, on-the-fly calibration of available cameras would require extra execution time, but the method would be relieved from the burden of fixed cameras.

Possible Extensions

In this section, we will mention some new ideas as possible extensions to the methods proposed within the scope of this thesis, but somewhat more demanding (in terms of time and resources) to implement compared to the improvements indicated in previous section.

Hand crafted appearance features are less and less used in all computer vision tasks. They are replaced by features extracted by convolutional neural networks, where hierarchical features are automatically learned on large datasets. Accordingly, this work may adopt a similar approach toward the extraction of appearance features, namely the ϕ term in Equation 6.1 which are computed using the older HOG [55] features in our experiments. Such extension may improve the precision of the appearance features and subsequently yield better overall performance for multi-view pose estimation.

Another plausible idea, which is relatively less related to the approaches proposed in this manuscript, would be further exploiting a deep convolutional neural network to learn the multi-view relations between the poses obtained from separate view angles. An arguable workflow would consist of the following steps: For each viewpoint, single-view pose estimation would be carried out classically first. A multi-branch convolutional neural network would take each image and corresponding part coordinates as input, to ultimately regress the 3D position of the person in the scene. Experiments to evaluate several fusion strategies would be conducted to to determine what kind of fusion (early, late or slow) works best to infer the 3D part locations of the individual. Alternatively, the single-view pose estimation step might be com-

pletely abandoned and another convolutional neural network architecture might be designed to simply take images from multiple viewpoints as inputs, and regress 3D part positions as outputs. Arguably, this sort of end-to-end learning approach would require a very large amount of data and considerable training time, but might significantly shorten the inference time in test stage.

8.4 List of Related Publications

International Conferences

- Emre Dogan, Gonen Eren, Christian Wolf, Eric Lombardi, Atilla Baskurt, *Multi-view Pose Estimation with Flexible Mixtures-of-Parts*, 2017 (Submitted to ACIVS 2017, under review)
- Emre Dogan, Gonen Eren, Christian Wolf, Atilla Baskurt, *Activity Recognition with Volume Motion Templates and Histograms of 3D Gradients*, Image Processing (ICIP), 2015 IEEE International Conference on, Quebec City, QC, 2015, pp. 4421-4425.

Journal Articles

- Emre Dogan, Gonen Eren, Christian Wolf, Eric Lombardi, Atilla Baskurt, *Multi-view pose estimation with mixtures-of-parts and adaptive viewpoint selection*, 2017 (Submitted to IET-CV, under revision)
- Christian Wolf, Eric Lombardi, Julien Mille, Oya Celiktutan, Mingyuan Jiu, Emre Dogan, Gonen Eren, Moez Baccouche, Emmanuel Dellandrea, Charles-Edmond Bichot, Christophe Garcia, Bulent Sankur, *Evaluation of video activity localizations integrating quality and quantity measurements: application to the ICPR 2012 HARL competition*, Computer Vision and Image Understanding 127, 2014, pp. 14-30

References

[1] Abdelkader, M. F., Abd-Almageed, W., Srivastava, A., and Chellappa, R. (2011). Silhouette-based gesture and action recognition via modeling trajectories on riemannian shape manifolds. *Computer Vision and Image Understanding*, 115(3):439 – 455. Special issue on Feature-Oriented Image and Video Computing for Extracting Contexts and Semantics.

[2] Agarwal, A. and Triggs, B. (2004). 3d human pose from silhouettes by relevance vector regression. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–882. IEEE.

[3] Agarwal, A. and Triggs, B. (2006). Recovering 3d human pose from monocular images. *IEEE T. on PAMI*, 28(1):44–58.

[4] Aggarwal, J. K. and Ryoo, M. S. (2011). Human activity analysis: A review. *ACM Comput. Surv.*, 43:16.

[5] Ahmad, M. and Lee, S.-W. (2006). Hmm-based human action recognition using multiview image sequences. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 1, pages 263–266.

[6] Amin, S., Andriluka, M., Rohrbach, M., and Schiele, B. (2013). Multi-view pictorial structures for 3d human pose estimation. In *BMVC*.

[7] Andriluka, M., Roth, S., and Schiele, B. (2008). People-tracking-by-detection and people-detection-by-tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8.

[8] Andriluka, M., Roth, S., and Schiele, B. (2009). Pictorial structures revisited: People detection and articulated pose estimation. In *CVPR*.

[9] Andriluka, M., Roth, S., and Schiele, B. (2010). Monocular 3d pose estimation and tracking by detection. In *CVPR*.

[10] Andriluka, M., Roth, S., and Schiele, B. (2012). Discriminative appearance models for pictorial structures. *International Journal of Computer Vision*, 99:259–280.

[11] Anguita, D., Ghio, A., Oneto, L., Parra, X., and Reyes-Ortiz, J. L. (2012). Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In *IWAAL*.

[12] Baak, A., Müller, M., Bharaj, G., Seidel, H.-P., and Theobalt, C. (2011). A data-driven approach for real-time full body pose reconstruction from a depth camera. In *ICCV*.

[13] Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., and Baskurt, A. (2011). Sequential deep learning for human action recognition. In *International Workshop on Human Behavior Understanding*, pages 29–39. Springer.

[14] Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., and Baskurt, A. (2012). Spatio-temporal convolutional sparse auto-encoder for sequence classification. In *BMVC*.

[15] Baradel, F., Wolf, C., and Mille, J. (2017). Pose-conditioned spatio-temporal attention for human action recognition. *arxiv*, 1703.10106.

[16] Barron, J. L., Fleet, D. J., and Beauchemin, S. S. (1994). Performance of optical flow techniques. *International Journal of Computer Vision*, 12:43–77.

[17] Bay, H., Tuytelaars, T., and Gool, L. V. (2006). Surf: Speeded up robust features. In *ECCV*.

[18] Belagiannis, V., Amin, S., Andriluka, M., Schiele, B., Navab, N., and Ilic, S. (2014a). 3d pictorial structures for multiple human pose estimation. In *CVPR*.

[19] Belagiannis, V., Amin, S., Andriluka, M., Schiele, B., Navab, N., and Ilic, S. (2015). 3d pictorial structures revisited: Multiple human pose estimation. *IEEE T. on PAMI*, PP(99):1–1.

[20] Belagiannis, V., Wang, X., Schiele, B., Fua, P., Ilic, S., and Navab, N. (2014b). Multiple human pose estimation with temporally consistent 3D pictorial structures. In *ChaLearn Looking at People (ECCV workshop)*.

[21] Belagiannis, V. and Zisserman, A. (2016). Recurrent human pose estimation. *arXiv preprint arXiv:1605.02914*.

[22] Belongie, S., Malik, J., and Puzicha, J. (2000). Shape context: A new descriptor for shape matching and object recognition. In *NIPS*, volume 2, page 3.

[23] Ben-Arie, J., Pandit, P., and Rajaram, S. (2001). View-based human activity recognition by indexing and sequencing. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages II–78. IEEE.

[24] Bhanu, B. and Han, J. (2002). Individual recognition by kinematic-based gait analysis. In *ICPR*, volume 3, pages 343–346.

[25] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[26] Blank, M., Gorelick, L., Shechtman, E., Irani, M., and Basri, R. (2005). Actions as space-time shapes. In *The Tenth IEEE International Conference on Computer Vision (ICCV'05)*, pages 1395–1402.

[27] Bo, L., Sminchisescu, C., Kanaujia, A., and Metaxas, D. (2008). Fast algorithms for large scale conditional 3d prediction. In *CVPR*, pages 1–8.

[28] Bobick, A. and Davis, J. (2001). The recognition of human movement using temporal templates. *PAMI*, 23(3):257–267.

[29] Bodor, R., Jackson, B., Masoud, O., and Papanikolopoulos, N. (2003). Image-based reconstruction for view-independent human motion recognition. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 2, pages 1548–1553. IEEE.

[30] Bogo, F., Kanazawa, A., Lassner, C., Gehler, P., Romero, J., and Black, M. J. (2016). *Keep It SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image*, pages 561–578. Springer International Publishing, Cham.

[31] Bouguet, J.-Y. (2004). Camera calibration tool box for matlab [eb/ol].

[32] Bourdev, L. D., Maji, S., and Malik, J. (2011). Describing people: A poselet-based approach to attribute classification. In *ICCV*.

[33] Bourdev, L. D. and Malik, J. (2009). Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*.

[34] Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239.

[35] Brand, M., Oliver, N., and Pentland, A. (1997). Coupled hidden markov models for complex action recognition. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 994–999.

[36] Bregler, C. (1997). Learning and recognizing human dynamics in video sequences. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 568–574.

[37] Burenius, M., Sullivan, J., and Carlsson, S. (2013). 3d pictorial structures for multiple view articulated pose estimation. In *CVPR*.

[38] Campbell, L. W., Becker, D. A., Azarbayejani, A., Bobick, A. F., and Pentland, A. (1996). Invariant features for 3-d gesture recognition. In *Automatic Face and Gesture Recognition, 1996., Proceedings of the Second International Conference on*, pages 157–162.

[39] Campbell, L. W. and Bobick, A. F. (1995). Recognition of human body motion using phase space constraints. In *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pages 624–630.

[40] Canton-Ferrer, C., Casas, J., and Pardas, M. (2009). Voxel based annealed particle filtering for markerless 3d articulated motion capture. In *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, pages 1–4.

[41] Carlsson, S. and Sullivan, J. (2001). Action recognition by shape matching to key frames. In *IEEE Computer Society Workshop on Models versus Exemplars in Computer Vision*.

[42] Carreira, J., Agrawal, P., Fragkiadaki, K., and Malik, J. (2015). Human pose estimation with iterative error feedback. *CoRR*, abs/1507.06550.

[43] Celiktutan, O., Wolf, C., Sankur, B., and Lombardi, E. (2014). Fast exact hyper-graph matching with dynamic programming for spatio-temporal data. *Journal of Mathematical Imaging and Vision*, pages 1–21.

[44] Chaaraoui, A. A., Climent-Pérez, P., and Flórez-Revuelta, F. (2012). An efficient approach for multi-view human action recognition based on bag-of-key-poses. In *HBU*.

[45] Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[46] Chen, X. and Yuille, A. (2014). Articulated pose estimation by a graphical model with image dependent pairwise relations. In *CVPR*.

[47] Chen, Y., Zhu, L., Lin, C., Yuille, A. L., and Zhang, H. (2007). Rapid inference on a novel and/or graph for object detection, segmentation and parsing. In *NIPS*.

[48] Cherian, A., Mairal, J., Alahari, K., and Schmid, C. (2014). Mixing body-part sequences for human pose estimation. In *CVPR*.

[49] Cho, E. and Kim, D. (2015). Accurate human pose estimation by aggregating multiple pose hypotheses using modified kernel density approximation. *Signal Processing Letters, IEEE*, 22(4):445–449.

[50] Crispim-Junior, C. F., Buso, V., Avgerinakis, K., Meditskos, G., Briassouli, A., Benois-Pineau, J., Kompatsiaris, I. Y., and Bremond, F. (2016). Semantic event fusion of different visual modality concepts for activity recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(8):1598–1611.

[51] Cuntoor, N. P., Yegnanarayana, B., and Chellappa, R. (2008). Activity modeling using event probability sequences. *IEEE Transactions on Image Processing*, 17(4):594–607.

[52] Cutler, R. (1998). View-based interpretation of real-time optical flow for gesture recognition. In *International Conference on Automatic Face and Gesture Recognition*, pages 416–421.

[53] Cuzzolin, F. (2006). Using bilinear models for view-invariant action and identity recognition. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1701–1708.

[54] Cuzzolin, F., Sarti, A., and Tubaro, S. (2004). Action modeling with volumetric data. In *Image Processing, 2004. ICIP'04. 2004 International Conference on*, volume 2, pages 881–884. IEEE.

[55] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *CVPR*, volume 1.

[56] Dalal, N., Triggs, B., and Schmid, C. (2006). Human Detection Using Oriented Histograms of Flow and Appearance. In *ECCV*, volume 3952, pages 428–441.

[57] Dantone, M., Gall, J., Leistner, C., and Van Gool, L. (2014). Body parts dependent joint regressors for human pose estimation in still images. *IEEE T. on PAMI*, 36(11):2131–2143.

[58] Darrell, T. and Pentland, A. (1993). Space-time gestures. In *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR '93., 1993 IEEE Computer Society Conference on*, pages 335–340.

[59] Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., aurelio Ranzato, M., Senior, A., Tucker, P., Yang, K., Le, Q. V., and Ng, A. Y. (2012). Large scale distributed deep networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 1223–1231. Curran Associates, Inc.

[60] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255.

[61] Dollar, P., Rabaud, V., Cottrell, G., and Belongie, S. (2005). Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, pages 65–72.

[62] Dollár, P., Wojek, C., Schiele, B., and Perona, P. (2012). Pedestrian detection: An evaluation of the state of the art. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34:743–761.

[63] Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., and Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[64] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. (2014). Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*.

[65] Edwards, M., Deng, J., and Xie, X. (2016). From pose to activity: Surveying datasets and introducing {CONVERSE}. *Computer Vision and Image Understanding*, 144:73 – 105. Individual and Group Activities in Video Event Analysis.

[66] Efros, A. A., Berg, A. C., Mori, G., and Malik, J. (2003). Recognizing action at a distance. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, volume 2, pages 726–733.

[67] Eichner, M. and Ferrari, V. (2009). Better appearance models for pictorial structures. In *BMVC*.

[68] Eichner, M. and Ferrari, V. (2013). Appearance sharing for collective human pose estimation. In *ACCV*, pages 138–151.

[69] Eichner, M., Marin-Jimenez, M., Zisserman, A., and Ferrari, V. (2012). 2d articulated human pose estimation and retrieval in (almost) unconstrained still images. *International Journal of Computer Vision*, 99(2):190–214.

[70] Elgammal, A., Harwood, D., and Davis, L. (2000). Non-parametric model for background subtraction. In *European conference on computer vision*, pages 751–767. Springer.

[71] Elgammal, A., Shet, V., Yacoob, Y., and Davis, L. S. (2003). Learning dynamics for exemplar-based gesture recognition. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–571–I–578 vol.1.

[72] Elhayek, A., de Aguiar, E., Jain, A., Tompson, J., Pishchulin, L., Andriluka, M., Bregler, C., Schiele, B., and Theobalt, C. (2015). Efficient convnet-based marker-less motion capture in general scenes with a low number of cameras. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3810–3818.

[73] Everingham, M., Eslami, S. M. A., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136.

[74] Fan, X., Zheng, K., Lin, Y., and Wang, S. (2015). Combining local appearance and holistic view: Dual-source deep neural networks for human pose estimation. In *ICCV*.

[75] Fathi, A. and Mori, G. (2008). Action recognition by learning mid-level motion features. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8.

[76] Felzenszwalb, P., Girshick, R., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *IEEE T. on PAMI*, 32(9):1627–1645.

[77] Felzenszwalb, P. F. and Huttenlocher, D. P. (2000). Efficient matching of pictorial structures. In *CVPR*.

[78] Felzenszwalb, P. F. and Huttenlocher, D. P. (2005). Pictorial structures for object recognition. *IJCV*, 61(1):55–79.

[79] Felzenszwalb, P. F. and Huttenlocher, D. P. (2012). Distance transforms of sampled functions. *Theory of computing*, 8(1):415–428.

[80] Ferrari, V., Marín-Jiménez, M. J., and Zisserman, A. (2008). Progressive search space reduction for human pose estimation. In *CVPR*.

[81] Filipovych, R. and Ribeiro, E. (2008). Learning human motion models from unsegmented videos. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–7.

[82] Finley, T. and Joachims, T. (2008). Training structural svms when exact inference is intractable. In *ICML*.

[83] Fischler, M. and Elschlager, R. (1973). The representation and matching of pictorial structures. *IEEE Transactions on Computers*, 22(1):67–92.

[84] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395.

[85] Forsyth, D. and Ponce, J. (2011). *Computer Vision: A Modern Approach*. Pearson Education.

[86] Gall, J., Rosenhahn, B., Brox, T., and Seidel, H.-P. (2010). Optimization and filtering for human motion capture. *IJCV*, 87(1-2):75–92.

[87] Gammeter, S., Ess, A., Jäggli, T., Schindler, K., Leibe, B., and Gool, L. (2008). Articulated multi-body tracking under egomotion. In *Proceedings of the 10th European Conference on Computer Vision: Part II*, ECCV '08, pages 816–830, Berlin, Heidelberg. Springer-Verlag.

[88] Gavrila, D. M. and Davis, L. S. (1995). Towards 3-d model-based tracking and recognition of human movement: a multi-view approach. In *In International Workshop on Automatic Face- and Gesture-Recognition. IEEE Computer Society*, pages 272–277.

[89] Gilbert, A., Illingworth, J., and Bowden, R. (2008). Scale invariant action recognition using compound features mined from dense spatio-temporal corners. In *European Conference on Computer Vision*, pages 222–233. Springer.

[90] Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587.

[91] Gkioxari, G., Arbeláez, P. A., Bourdev, L. D., and Malik, J. (2013). Articulated pose estimation using discriminative armlet classifiers. In *CVPR*.

[92] Gkioxari, G., Hariharan, B., Girshick, R. B., and Malik, J. (2014a). R-cnns for pose estimation and action detection. *CoRR*, abs/1406.5212.

[93] Gkioxari, G., Hariharan, B., Girshick, R. B., and Malik, J. (2014b). Using k-poselets for detecting people and localizing their keypoints. In *CVPR*.

[94] Gkioxari, G. and Malik, J. (2015). Finding action tubes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[95] Goh, H., Thome, N., Cord, M., and Lim, J. (2012). Unsupervised and supervised visual codes with restricted boltzmann machines. In *ECCV*.

[96] Grauman, K. and Darrell, T. (2005). Pyramid match kernels: Discriminative classification with sets of image features. In *International Conference on Computer Vision (ICCV)*.

[97] Green, R. D. and Guan, L. (2004). Quantifying and recognizing human movement patterns from monocular video images-part i: a new framework for modeling human motion. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(2):179–190.

[98] Gritai, A., Sheikh, Y., and Shah, M. (2004). On the use of anthropometry in the invariant analysis of human actions. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 923–926. IEEE.

[99] Guo, Y., Xu, G., and Tsuji, S. (1994). Understanding human motion patterns. In *Pattern Recognition, 1994. Vol. 2 - Conference B: Computer Vision amp; Image Processing., Proceedings of the 12th IAPR International. Conference on*, volume 2, pages 325–329 vol.2.

[100] Hariyono, J., Kurnianggoro, L., Wahyono, Hernandez, D. C., and Jo, K.-H. (2014). *Ego-Motion Compensated for Moving Object Detection in a Mobile Robot*, pages 289–297. Springer International Publishing, Cham.

[101] Hartley, R. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press.

[102] Harville, M. and Li, D. (2004). Fast, integrated person tracking and activity recognition with plan-view templates from a single stereo camera. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–398–II–405 Vol.2.

[103] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[104] Hernández-Vela, A., Sclaroff, S., and Escalera, S. (2016). Poselet-based contextual rescoring for human pose estimation via pictorial structures. *International Journal of Computer Vision*, 118:49–64.

[105] Hess, R., Fern, A., and Mortensen, E. (2007). Mixture-of-parts pictorial structures for objects with variable part sets. In *ICCV*, pages 1–8.

[106] Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554.

[107] Hofmann, M. and Gavrila, D. M. (2009). Multi-view 3d human pose estimation combining single-frame recovery, temporal integration and model adaptation. In *CVPR*, pages 2214–2221.

[108] Hofmann, M. and Gavrila, D. M. (2012). Multi-view 3d human pose estimation in complex environment. *International Journal of Computer Vision*, 96(1):103–124.

[109] Hogg, D. (1983a). Model-based vision: a program to see a walking person. *Image and Vision Computing*, 1(1):5 – 20.

[110] Hogg, D. C. (1983b). Model-based vision: a program to see a walking person. *Image Vision Comput.*, 1:5–20.

[111] Holte, M., Moeslund, T., and Fihl, P. (2010). View-invariant gesture recognition using 3d optical flow and harmonic motion context. *CVIU*, 114(12):1353 – 1361.

[112] Holte, M. B., Moeslund, T. B., and Fihl, P. (2008). View invariant gesture recognition using the csem swissranger sr-2 camera. *International Journal of Intelligent Systems Technologies and Applications*, 5(3-4):295–303.

[113] Holte, M. B., Tran, C., Trivedi, M. M., and Moeslund, T. B. (2011). Human action recognition using multiple views: A comparative perspective on recent developments. In *Proceedings of the 2011 Joint ACM Workshop on Human Gesture and Behavior Understanding*, J-HGBU '11, pages 47–52, New York, NY, USA. ACM.

[114] Holte, M. B., Tran, C., Trivedi, M. M., and Moeslund, T. B. (2012). Human pose estimation and activity recognition from multi-view videos: Comparative explorations of recent developments. *IEEE Journal of Selected Topics in Signal Processing*, 6(5):538–552.

[115] Horn, B. K. and Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence*, 17(1):185 – 203.

[116] Ikizler, N. and Duygulu, P. (2009). Histogram of oriented rectangles: A new pose descriptor for human action recognition. *Image and Vision Computing*, 27(10):1515 – 1526. Special Section: Computer Vision Methods for Ambient Intelligence.

[117] Ikizler, N. and Forsyth, D. (2007). Searching video for complex activities with finite state models. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

[118] Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M., and Schiele, B. (2016). *DeeperCut: A Deeper, Stronger, and Faster Multi-person Pose Estimation Model*, pages 34–50. Springer International Publishing, Cham.

[119] Ji, S., Xu, W., Yang, M., and Yu, K. (2013). 3d convolutional neural networks for human action recognition. *PAMI*, 35(1):221–231.

[120] Jia, K. and Yeung, D.-Y. (2008). Human action recognition using local spatio-temporal discriminant embedding. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8.

[121] Jiu, M., Wolf, C., and Baskurt, A. (2013). Integrating spatial layout of object parts into classification without pairwise terms: application to fast body parts estimation from depth images. In *VISAPP*, pages 626–631.

[122] Jiu, M., Wolf, C., Garcia, C., and Baskurt, A. (2012). Supervised learning and codebook optimization for bag of words models. *Cognitive Computation*, 4:409–419.

[123] Johnson, S. and Everingham, M. (2010). Clustered pose and nonlinear appearance models for human pose estimation. In *BMVC*.

[124] Johnson, S. and Everingham, M. (2011). Learning effective human pose estimation from inaccurate annotation. In *CVPR*.

[125] Junejo, I. N., Dexter, E., Laptev, I., and Pérez, P. (2008). *Cross-View Action Recognition from Temporal Self-similarities*, pages 293–306. Springer Berlin Heidelberg, Berlin, Heidelberg.

[126] Kale, A., Chowdhury, A. K. R., and Chellappa, R. (2003). Towards a view invariant gait recognition algorithm. In *Advanced Video and Signal Based Surveillance, 2003. Proceedings. IEEE Conference on*, pages 143–150.

[127] Kanaujia, A., Kittens, N., and Ramanathan, N. (2013). Part segmentation of visual hull for 3d human pose estimation. In *CVPR Workshop*.

[128] Karpathy, A. and Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[129] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Li, F.-F. (2014). Large-scale video classification with convolutional neural networks. In *CVPR*.

[130] Kazemi, V., Burenius, M., Azizpour, H., and Sullivan, J. (2013). Multi-view body part recognition with random forests. In *BMVC*.

[131] Kazhdan, M. (2004). *Shape representations and algorithms for 3D model retrieval*. PhD thesis, Princeton University.

[132] Ke, Y., Sukthankar, R., and Hebert, M. (2005). Efficient visual event detection using volumetric features. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 166–173 Vol. 1.

[133] Kern, N., Schiele, B., and Schmidt, A. (2003). Multi-sensor activity context detection for wearable computing. In *European Symposium on Ambient Intelligence*, pages 220–232. Springer.

[134] Kiefel, M. and Gehler, P. (2014). Human pose estimation with fields of parts. In *ECCV*, pages 331–346.

[135] Klaser, A., Marszalek, M., and Schmid, C. (2008). A Spatio-Temporal Descriptor Based on 3D-Gradients. In *BMVC*, pages 275:1–10.

[136] Kolb, A., Barth, E., Koch, R., and Larsen, R. (2010). Time-of-flight cameras in computer graphics. *Computer Graphics Forum*, 29(1):141–159.

[137] Kovashka, A. and Grauman, K. (2010). Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2046–2053.

[138] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc.

[139] Kwapisz, J. R., Weiss, G. M., and Moore, S. A. (2011). Activity recognition using cell phone accelerometers. *SIGKDD Explor. Newsl.*, 12(2):74–82.

[140] Laptev, I. (2005). On space-time interest points. *IJCV*, 64(2-3):107–123.

[141] Laptev, I. and Lindeberg, T. (2003). Space-time interest points. In *ICCV*.

[142] Laptev, I., Marszalek, M., Schmid, C., and Rozenfeld, B. (2008). Learning realistic human actions from movies. In *CVPR*.

[143] Laptev, I. and Perez, P. (2007). Retrieving actions in movies. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8.

[144] Lara, O. D. and Labrador, M. A. (2013). A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys and Tutorials*, 15:1192–1209.

[145] Lassner, C., Romero, J., Kiefel, M., Bogo, F., Black, M. J., and Gehler, P. V. (2017). Unite the people: Closing the loop between 3d and 2d human representations. *CoRR*, abs/1701.02468.

[146] Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*.

[147] Le, Q. V. (2013). Building high-level features using large scale unsupervised learning. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8595–8598.

[148] Le, Q. V., Zou, W. Y., Yeung, S. Y., and Ng, A. Y. (2011). Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3361–3368.

[149] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

[150] Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[151] LeCun, Y. and Ranzato, M. (2013). Deep learning tutorial. In *Tutorials in International Conference on Machine Learning (ICML13), Citeseer*. Citeseer.

[152] Lee, C.-S. and Elgammal, A. (2007). Modeling view and posture manifolds for tracking. In *ICCV*, pages 1–8.

[153] Li, B., Ayazoglu, M., Mao, T., Camps, O. I., and Sznaier, M. (2011). Activity recognition using dynamic subspace angles. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3193–3200.

[154] Li, B., Camps, O. I., and Sznaier, M. (2012). Cross-view activity recognition using hankelets. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1362–1369.

[155] Li, S., Zhang, W., and Chan, A. B. (2015). Maximum-margin structured learning with deep networks for 3d human pose estimation. In *ICCV*.

[156] Lifshitz, I., Fetaya, E., and Ullman, S. (2016). Human pose estimation using deep consensus voting. *CoRR*, abs/1603.08212.

[157] Liu, J. and Shah, M. (2008). Learning human actions via information maximization. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8.

[158] Liu, J., Shah, M., Kuipers, B., and Savarese, S. (2011). Cross-view action recognition via view knowledge transfer. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3209–3216.

[159] Lloyd, S. (1982). Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137.

[160] López-Quintero, M. I., Marín-Jiménez, M. J., Muñoz-Salinas, R., Madrid-Cuevas, F. J., and Carnicer, R. M. (2015). Stereo pictorial structure for 2d articulated human pose estimation. *Machine Vision and Applications*, 27(2):157–174.

[161] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110.

[162] Lu, W.-L. and Little, J. J. (2006). Simultaneous tracking and action recognition using the pca-hog descriptor. In *The 3rd Canadian Conference on Computer and Robot Vision (CRV'06)*, pages 6–6.

[163] Lv, F. and Nevatia, R. (2006). Recognition and segmentation of 3-d human action using hmm and multi-class adaboost. In *European conference on computer vision*, pages 359–372. Springer.

[164] Lv, F. and Nevatia, R. (2007). Single view human action recognition using key pose matching and viterbi path searching. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

[165] Mahalanobis, P. C. (1936). On the generalized distance in statistics. *Proceedings of the National Institute of Sciences (Calcutta)*, 2:49–55.

[166] Maji, S., Bourdev, L., and Malik, J. (2011). Action recognition from a distributed representation of pose and appearance. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3177–3184.

[167] Marr, D. and Nishihara, H. K. (1978). Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings of the Royal Society of London B: Biological Sciences*, 200(1140):269–294.

[168] Matas, J., Chum, O., Urban, M., and Pajdla, T. (2004). Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761 – 767. British Machine Vision Computing 2002.

[169] Messing, R., Pal, C., and Kautz, H. (2009). Activity recognition using the velocity histories of tracked keypoints. In *2009 IEEE 12th International Conference on Computer Vision*, pages 104–111.

[170] Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1615–1630.

[171] Mikolajczyk, K. and Uemura, H. (2008). Action recognition with motion-appearance vocabulary forest. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8.

[172] Mikolajczyk, K. and Uemura, H. (2011). Action recognition with appearanceâĂŞmotion features and fast search trees. *Computer Vision and Image Understanding*, 115(3):426 – 438. Special issue on Feature-Oriented Image and Video Computing for Extracting Contexts and Semantics.

[173] Mittal, A., Zhao, L., and Davis, L. S. (2003). Human body pose estimation using silhouette shape analysis. In *AVSS*.

[174] Natarajan, P. and Nevatia, R. (2008). View and scale invariant action recognition using multiview shape-flow models. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8.

[175] Natarajan, P., Singh, V. K., and Nevatia, R. (2010). Learning 3d action models from a few 2d videos for view invariant action recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 20006–2013.

[176] Neverova, N., Wolf, C., , Taylor, G., and Nebout, F. (2015a). Moddrop: adaptive multi-modal gesture recognition. *Pre-print: arXiv:1501.00102*.

[177] Neverova, N., Wolf, C., Lacey, G., Fridman, L., Chandra, D., Barbello, B., and Taylor, G. (2016a). Learning human identity from motion patterns. *IEEE Access*, 4:1810–1820.

[178] Neverova, N., Wolf, C., Taylor, G., and Nebout, F. (2015b). Hand pose estimation through weakly-supervised learning of a rich intermediate representation. *Pre-print: arxiv:1511.06728*.

[179] Neverova, N., Wolf, C., Taylor, G., and Nebout, F. (2016b). Moddrop: Adaptive multi-modal gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(8):1692–1706.

[180] Newell, A., Yang, K., and Deng, J. (2016). Stacked hourglass networks for human pose estimation. *CoRR*, abs/1603.06937.

[181] Ni, B., Pei, Y., Moulin, P., and Yan, S. (2013). Multilevel depth and image fusion for human activity detection. *Cybernetics, IEEE Transactions on*, 43(5):1383–1394.

[182] Niebles, J. C. and Fei-Fei, L. (2007). A hierarchical model of shape and appearance for human action classification. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

[183] Niyogi, S. A. and Adelson, E. H. (1994). Analyzing and recognizing walking figures in xyt. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, pages 469–474.

[184] Ofli, F., Chaudhry, R., Kurillo, G., Vidal, R., and Bajcsy, R. (2013). Berkeley mhad: A comprehensive multimodal human action database. In *Applications of Computer Vision (WACV), 2013 IEEE Workshop on*, pages 53–60.

[185] Ogale, A. S., Karapurkar, A., Guerra-filho, G., and Aloimonos, Y. (2004). View-invariant identification of pose sequences for action recognition. In *In VACE*.

[186] Oquab, M., Bottou, L., Laptev, I., and Sivic, J. (2014). Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*.

[187] Oreifej, O. and Liu, Z. (2013). Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[188] O'Rourke, J. and Badler, N. I. (1980). Model-based image analysis of human motion using constraint propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-2(6):522–536.

[189] Ouyang, W., Chu, X., and Wang, X. (2014). Multi-source deep learning for human pose estimation. In *CVPR*.

[190] Parameswaran, V. and Chellappa, R. (2006). View invariance for human action recognition. *International Journal of Computer Vision*, 66(1):83–101.

[191] Park, D. and Ramanan, D. (2015). Articulated pose estimation with tiny synthetic videos. In *CVPR Workshop*, pages 58–66.

[192] Park, S. and Aggarwal, J. K. (2003). Recognition of two-person interactions using a hierarchical bayesian network. In *First ACM SIGMM International Workshop on Video Surveillance*, IWVS '03, pages 65–76. ACM.

[193] Pearl, J. (1989). Probabilistic reasoning in intelligent systems - networks of plausible inference. In *DAGLIB*.

[194] Peng, X., Wang, L., Wang, X., and Qiao, Y. (2016). Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. *Computer Vision and Image Understanding*, 150:109 – 125.

[195] Pentland, A. P. (1987). A new sense for depth of field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(4):523–531.

[196] Perez-Sala, X., Escalera, S., and Angulo, C. (2012). Survey on spatio-temporal view invariant human pose recovery. In *Proceedings of the 15th International Conference of the Catalan Association of Artificial Intelligence (CCIA2012), Catalonia, Spain*, pages 24–26.

[197] Perez-Sala, X., Escalera, S., Angulo, C., and Gonzalez, J. (2014). A survey on model based approaches for 2d and 3d visual human pose recovery. *Sensors*, 14(3):4189–4210.

[198] Peursum, P., Venkatesh, S., and West, G. (2007). Tracking-as-recognition for articulated full-body human motion analysis. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

[199] Pfister, T., Charles, J., and Zisserman, A. (2015). Flowing convnets for human pose estimation in videos. *CoRR*, abs/1506.02897.

[200] Pierobon, M., Marcon, M., Sarti, A., and Tubaro, S. (2006). 3-d body posture tracking for human action template matching. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 2, pages II–II.

[201] Pishchulin, L., Andriluka, M., Gehler, P., and Schiele, B. (2013a). Poselet conditioned pictorial structures. In *CVPR*.

[202] Pishchulin, L., Andriluka, M., Gehler, P. V., and Schiele, B. (2013b). Strong appearance and expressive spatial models for human pose estimation. In *ICCV*.

[203] Plagemann, C., Ganapathi, V., Koller, D., and Thrun, S. (2010). Real-time identification and localization of body parts from depth images. In *ICRA*.

[204] Polana, R. and Nelson, R. (1994). Low level recognition of human motion (or how to get your man without finding his body parts). In *Motion of Non-Rigid and Articulated Objects, 1994., Proceedings of the 1994 IEEE Workshop on*, pages 77–82.

[205] Poppe, R. (2010). A survey on vision-based human action recognition. *Image Vision Comput.*, 28:976–990.

[206] Poppe, R. W. (2007). Evaluating example-based pose estimation: Experiments on the humaneva sets. Technical Report TR-CTIT-07-72, Centre for Telematics and Information Technology University of Twente, Enschede.

[207] Porrill, J., Pollard, S., and Mayhew, J. (1987). 2nd alvey vision meeting optimal combination of multiple sensors including stereo vision. *Image and Vision Computing*, 5(2):174 – 180.

[208] Prazdny, K. (1980). Egomotion and relative depth map from optical flow. *Biological Cybernetics*, 36(2):87–102.

[209] Puwein, J., Ballan, L., Ziegler, R., and Pollefeys, M. (2014). Joint camera pose estimation and 3d human pose estimation in a multi-camera setup. In *ACCV*.

[210] Qinfeng, S., Li, C., Li, W., and Smola, A. (2011). Human action segmentation and recognition using discriminative semi-markov models. *International Journal of Computer Vision*, 93(1):22 – 32.

[211] Rahmani, H., Mahmood, A., Huynh, D., and Mian, A. (2016). Histogram of oriented principal components for cross-view action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1.

[212] Rahmani, H. and Mian, A. (2015). Learning a non-linear knowledge transfer model for cross-view action recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[213] Rahmani, H. and Mian, A. (2016). 3d action recognition from novel viewpoints. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[214] Ramanan, D. (2006). Learning to parse images of articulated bodies. In *NIPS*.

[215] Ramanan, D. (2011). Part-based models for finding people and estimating their pose. In Moeslund, B. T., Hilton, A., Krüger, V., and Sigal, L., editors, *Visual Analysis of Humans: Looking at People*, pages 199–223, London. Springer London.

[216] Ramanan, D. and Forsyth, D. A. (2004). Automatic annotation of everyday movements. In Thrun, S., Saul, L. K., and Schölkopf, B., editors, *Advances in Neural Information Processing Systems 16*, pages 1547–1554. MIT Press.

[217] Rao, C., Yilmaz, A., and Shah, M. (2002). View-invariant representation and recognition of actions. *Int. J. Comput. Vision*, 50(2):203–226.

[218] Rapantzikos, K., Avrithis, Y., and Kollias, S. (2009). Dense saliency-based spatiotemporal feature points for action recognition. In *CVPR*, pages 1454–1461.

[219] Raudies, F. and Neumann, H. (2012). A review and evaluation of methods estimating ego-motion. *Computer Vision and Image Understanding*, 116(5):606 – 633.

[220] Razavian, A. S., Azizpour, H., Sullivan, J., and Carlsson, S. (2014). Cnn features off-the-shelf: An astounding baseline for recognition. In *CVPR*.

[221] Riboni, D. and Bettini, C. (2011). Cosar: hybrid reasoning for context-aware activity recognition. *Personal and Ubiquitous Computing*, 15:271–289.

[222] Rittscher, J. and Blake, A. (1999). Classification of human body motion. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 634–639 vol.1.

[223] Rius, I., Gonzàlez, J., Varona, J., and Roca, F. X. (2009). Action-specific motion prior for efficient bayesian 3d human body tracking. *Pattern Recognition*, 42:2907–2921.

[224] Robertson, N. and Reid, I. (2005). Behaviour understanding in video: a combined method. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 808–815 Vol. 1.

[225] Rodriguez, M. D., Ahmed, J., and Shah, M. (2008). Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*.

[226] Rogez, G., Guerrero, J. J., Martínez, J., and Orrite-Urunuela, C. (2006). Viewpoint independent human motion analysis in man-made environments. In *BMVC*, volume 6, page 659.

[227] Roh, M., Shin, H., Lee, S., and Lee, S. (2006). Volume motion template for view-invariant gesture recognition. In *ICPR*, volume 2, pages 1229–1232.

[228] Rohr, K. (1994). Towards model-based recognition of human movements in image sequences. *CVGIP: Image Understanding*, 59(1):94 – 115.

[229] Ronfard, R., Schmid, C., and Triggs, B. (2002). Learning to parse pictures of people. In *ECCV*.

[230] Rosales, R. and Sclaroff, S. (2001). Learning body pose via specialized maps. In *Advances in neural information processing systems*, pages 1263–1270.

[231] Rosales, R. and Sclaroff, S. (2006). Combining generative and discriminative models in a framework for articulated pose estimation. *International Journal of Computer Vision*, 67(3):251–276.

[232] Rosenblatt, F. (1960). Perceptron simulation experiments. *Proceedings of the IRE*, 48(3):301–309.

[233] Rusu, R. B., Marton, Z. C., Blodow, N., Dolha, M., and Beetz, M. (2008). Towards 3d point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56(11):927 – 941.

[234] Rybok, L., Friedberger, S., Hanebeck, U. D., and Stiefelhagen, R. (2011). The KIT Robo-Kitchen Data set for the Evaluation of View-based Activity Recognition Systems. In *IEEE-RAS International Conference on Humanoid Robots*.

[235] Ryoo, M. S. and Aggarwal, J. K. (2009). Spatio-temporal relationship match: video structure comparison for recognition of complex human activities. In *ICCV*.

[236] Saha, S., Singh, G., Sapienza, M., Torr, P. H. S., and Cuzzolin, F. (2016). Deep Learning for Detecting Multiple Space-Time Action Tubes in Videos. *ArXiv e-prints*.

[237] Salvi, J., ArmanguÃČÂl', X., and Batlle, J. (2002). A comparative review of camera calibrating methods with accuracy evaluation. *Pattern Recognition*, 35(7):1617 – 1635.

[238] Sapp, B., Jordan, C., and Taskar, B. (2010). Adaptive pose priors for pictorial structures. In *CVPR*, pages 422–429.

[239] Sapp, B. and Taskar, B. (2013). Modec: Multimodal decomposable models for human pose estimation. In *CVPR*.

[240] Sapp, B., Weiss, D. J., and Taskar, B. (2011). Parsing human motion with stretchable models. In *CVPR*.

[241] Schick, A. and Stiefelhagen, R. (2015). 3d pictorial structures for human pose estimation with supervoxels. In *IEEE Winter Conf. on Applications of Computer Vision*, pages 140–147.

[242] Schindler, K. and van Gool, L. (2008). Action snippets: How many frames does human action recognition require? In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8.

[243] Schölkopf, B., Smola, A. J., Williamson, R. C., and Bartlett, P. L. (2000). New support vector algorithms. *Neural Comput.*, 12(5):1207–1245.

[244] Schuldt, C., Laptev, I., and Caputo, B. (2004). Recognizing human actions: a local svm approach. In *ICPR*, volume 3, pages 32–36 Vol.3.

[245] Scovanner, P., Ali, S., and Shah, M. (2007). A 3-dimensional sift descriptor and its application to action recognition. In *15th International Conference on Multimedia*, pages 357–360.

[246] Seitz, S. M. and Dyer, C. R. (1997). View-invariant analysis of cyclic motion. *International Journal of Computer Vision*, 25(3):231–251.

[247] Senst, T., Evangelio, R. H., Keller, I., and Sikora, T. (2012). Clustering motion for real-time optical flow based tracking. In *IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS 2012)*, pages 410–415, Beijing, China. ISBN: 978-1-4673-2499-1 DOI: 10.1109/AVSS.2012.20.

[248] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y. (2013a). Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*.

[249] Sermanet, P., Kavukcuoglu, K., Chintala, S., and LeCun, Y. (2013b). Pedestrian detection with unsupervised multi-stage feature learning. In *CVPR*.

[250] Shahroudy, A., Ng, T.-T., Gong, Y., and Wang, G. (2016). Deep multimodal feature analysis for action recognition in rgb+ d videos. *arXiv preprint arXiv:1603.07120*.

[251] Shakhnarovich, G., Lee, L., and Darrell, T. (2001). Integrated face and gait recognition from multiple views. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–439–I–446 vol.1.

[252] Shakhnarovich, G., Viola, P., and Darrell, T. (2003a). Fast pose estimation with parameter-sensitive hashing. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 750–757. IEEE.

[253] Shakhnarovich, G., Viola, P., and Darrell, T. (2003b). Fast pose estimation with parameter-sensitive hashing. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 750–757 vol.2.

[254] Shao, L., Gao, R., Liu, Y., and Zhang, H. (2011). Transform based spatio-temporal descriptors for human action recognition. *Neurocomputing*, 74(6):962 – 973.

[255] Sheikh, Y., Sheikh, M., and Shah, M. (2005). Exploring the space of a human action. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 144–149 Vol. 1.

[256] Shen, Y. and Foroosh, H. (2008). View-invariant action recognition using fundamental ratios. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–6.

[257] Shimosaka, M., Sagawa, Y., Mori, T., and Sato, T. (2009). 3d voxel based online human pose estimation via robust and efficient hashing. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3577–3582. IEEE.

[258] Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. (2011). Real-time human pose recognition in parts from single depth images. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1297–1304.

[259] Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., and Moore, R. (2013). Real-time human pose recognition in parts from single depth images. *Commun. ACM*, 56(1):116–124.

[260] Sigal, L., Balan, A., and Black, M. J. (2008). Combined discriminative and generative articulated pose and non-rigid shape estimation. In Platt, J. C., Koller, D., Singer, Y., and Roweis, S. T., editors, *Advances in Neural Information Processing Systems 20*, pages 1337–1344. Curran Associates, Inc.

[261] Sigal, L., Balan, A. O., and Black, M. J. (2010). Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *IJCV*, 87(1-2):4–27.

[262] Sigal, L., Bhatia, S., Roth, S., Black, M. J., and Isard, M. (2004). Tracking loose-limbed people. In *CVPR*.

[263] Sigal, L. and Black, M. J. (2006a). Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. Technical Report CS-06-08, Department of Computer Science, Brown University, Providence, Rhode Island 02912.

[264] Sigal, L. and Black, M. J. (2006b). Predicting 3d people from 2d pictures. In *International Conference on Articulated Motion and Deformable Objects*, pages 185–195. Springer.

[265] Sigal, L., Isard, M., Haussecker, H., and Black, M. J. (2011). Loose-limbed people: Estimating 3D human pose and motion using non-parametric belief propagation. *IJCV*, 98(1):15–48.

[266] Sigal, L., Isard, M., Sigelman, B. H., and Black, M. J. (2003). Attractive people: Assembling loose-limbed models using non-parametric belief propagation. In *NIPS*.

[267] Simo-Serra, E., Ramisa, A., Alenyà, G., Torras, C., and Moreno-Noguer, F. (2012). Single image 3d human pose estimation from noisy observations. In *CVPR*.

[268] Simonyan, K. and Zisserman, A. (2014a). Two-stream convolutional networks for action recognition in videos. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 568–576. Curran Associates, Inc.

[269] Simonyan, K. and Zisserman, A. (2014b). Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556.

[270] Slama, R., Wannous, H., Daoudi, M., and Srivastava, A. (2015). Accurate 3d action recognition using learning on the grassmann manifold. *Pattern Recognition*, 48(2):556 – 567.

[271] Sminchisescu, C., Kanaujia, A., Li, Z., and Metaxas, D. (2005). Discriminative density propagation for 3d human motion estimation. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 390–397. IEEE.

[272] Sminchisescu, C., Kanaujia, A., and Metaxas, D. (2006). Conditional models for contextual human motion recognition. *Computer Vision and Image Understanding*, 104(2âĂŞ3):210 – 220.

[273] Sminchisescu, C. and Triggs, B. (2003). Kinematic jump processes for monocular 3d human tracking. In *CVPR*.

[274] Song, Y., Morency, L. P., and Davis, R. (2012). Multi-view latent variable discriminative models for action recognition. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2120–2127.

[275] Souvenir, R. and Babbs, J. (2008). Learning the viewpoint manifold for action recognition. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–7.

[276] Sridhar, S., Oulasvirta, A., and Theobalt, C. (2013). Interactive markerless articulated hand motion tracking using rgb and depth data. In *ICCV*.

[277] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

[278] Starner, T. and Pentland, A. (1995). Real-time american sign language recognition from video using hidden markov models. In *Computer Vision, 1995. Proceedings., International Symposium on*, pages 265–270.

[279] Stoll, C., Hasler, N., Gall, J., Seidel, H.-P., and Theobalt, C. (2011). Fast articulated motion tracking using a sums of gaussians body model. In *ICCV*.

[280] Sturm, P. F. and Maybank, S. J. (1999). On plane-based camera calibration: A general algorithm, singularities, applications. In *CVPR*.

[281] Sun, J., Wu, X., Yan, S., Cheong, L. F., Chua, T. S., and Li, J. (2009). Hierarchical spatio-temporal context modeling for action recognition. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2004–2011.

[282] Sundaram, N., Brox, T., and Keutzer, K. (2010). Dense point trajectories by gpu-accelerated large displacement optical flow. In *European Conference on Computer Vision (ECCV)*, Lecture Notes in Computer Science. Springer.

[283] Syeda-Mahmood, T., Vasilescu, A., and Sethi, S. (2001). Recognizing action events from multiple viewpoints. In *Detection and Recognition of Events in Video, 2001. Proceedings. IEEE Workshop on*, pages 64–72.

[284] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9.

[285] Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*. Texts in Computer Science. Springer London.

[286] Ta, A., Wolf, C., LavouÃl', G., and Baskurt, A. (2010a). Recognizing and localizing individual activities through graph matching. In *AVSS*, pages 196–203.

[287] Ta, A. P., Wolf, C., Lavoue, G., Baskurt, A., and Jolion, J. M. (2010b). Pairwise features for human action recognition. In *2010 20th International Conference on Pattern Recognition*, pages 3224–3227.

[288] Taylor, G., Sigal, L., Fleet, D., and Hinton, G. (2010a). Dynamical binary latent variable models for 3d human pose tracking. In *CVPR*, pages 631–638.

[289] Taylor, G. W., Fergus, R., LeCun, Y., and Bregler, C. (2010b). Convolutional learning of spatio-temporal features. In *European conference on computer vision*, pages 140–153. Springer.

[290] Thurau, C. and Hlavac, V. (2008). Pose primitive based human action recognition in videos or still images. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8.

[291] Tian, Y., Sukthankar, R., and Shah, M. (2013). Spatiotemporal deformable part models for action detection. In *CVPR*, pages 2642–2649.

[292] Tian, Y., Zitnick, C. L., and Narasimhan, S. G. (2012). Exploring the spatial hierarchy of mixture models for human pose estimation. In *ECCV*.

[293] Tompson, J. J., Jain, A., LeCun, Y., and Bregler, C. (2014). Joint training of a convolutional network and a graphical model for human pose estimation. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 1799–1807. Curran Associates, Inc.

[294] Toshev, A. and Szegedy, C. (2014). Deeppose: Human pose estimation via deep neural networks. In *CVPR*.

[295] Tran, D. and Sorokin, A. (2008). Human activity recognition with metric learning. In *European conference on computer vision*, pages 548–561. Springer.

[296] Tsai, R. Y. (1987). A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE J. Robotics and Automation*, 3:323–344.

[297] Turaga, P., Chellappa, R., Subrahmanian, V. S., and Udrea, O. (2008a). Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1473–1488.

[298] Turaga, P., Veeraraghavan, A., and Chellappa, R. (2008b). Statistical analysis on stiefel and grassmann manifolds with applications in computer vision. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8.

[299] Urtasun, R. and Darrell, T. (2008). Sparse probabilistic regression for activity-independent human pose inference. In *CVPR*, pages 1–8.

[300] Urtasun, R. and Fua, P. (2004a). 3d human body tracking using deterministic temporal motion models. In *ECCV*.

[301] Urtasun, R. and Fua, P. (2004b). 3d tracking for gait characterization and recognition. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pages 17–22.

[302] Valmadre, J. and Lucey, S. (2010). Deterministic 3d human pose estimation using rigid structure. In *ECCV*.

[303] van der Aa, N. P., Luo, X., Giezeman, G. J., Tan, R. T., and Veltkamp, R. C. (2011a). Umpm benchmark: A multi-person dataset with synchronized video and motion capture data for evaluation of articulated human motion and interaction. In *Workshop on Human Interaction in Computer Vision (HICV), in conjunction with ICCV, Barcelona, Spain*, pages 1264–1269.

[304] van der Aa, N. P., Luo, X., Giezeman, G. J., Tan, R. T., and Veltkamp, R. C. (2011b). Utrecht multi-person motion (umpm) benchmark. Technical Report UU-CS-2011-027, Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands.

[305] Vishwakarma, S. and Agrawal, A. (2013). A survey on activity recognition and behavior understanding in video surveillance. *The Visual Computer*, 29(10):983–1009.

[306] Wang, C., Wang, Y., Lin, Z., Yuille, A. L., and Gao, W. (2014a). Robust estimation of 3d human poses from a single image. In *CVPR*.

[307] Wang, F. and Li, Y. (2013). Beyond physical connections: Tree models in human pose estimation. *CoRR*, abs/1305.2269.

[308] Wang, H., Klaser, A., Schmid, C., and Liu, C. (2011a). Action recognition by dense trajectories. In *CVPR*, pages 3169–3176.

[309] Wang, H. and Schmid, C. (2013). Action recognition with improved trajectories. In *ICCV*, pages 3551–3558.

[310] Wang, H., Ullah, M. M., Klaser, A., Laptev, I., and Schmid, C. (2009). Evaluation of local spatio-temporal features for action recognition. In Cavallaro, A., Prince, S., and Alexander, D., editors, *BMVC 2009 - British Machine Vision Conference*, pages 124.1–124.11, London, United Kingdom. BMVA Press.

[311] Wang, J., Chen, Z., and Wu, Y. (2011b). Action recognition with multiscale spatio-temporal contexts. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3185–3192.

[312] Wang, J., Liu, Z., Wu, Y., and Yuan, J. (2012). Mining actionlet ensemble for action recognition with depth cameras. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1290–1297.

[313] Wang, J., Nie, X., Xia, Y., Wu, Y., and Zhu, S.-C. (2014b). Cross-view action modeling, learning, and recognition. In *CVPR*.

[314] Wang, L. and Suter, D. (2007). Recognizing human activities from silhouettes: Motion subspace and factorial discriminative graphical model. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

[315] Wang, Y., Jiang, H., Drew, M. S., Li, Z.-N., and Mori, G. (2006). Unsupervised discovery of action classes. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1654–1661.

[316] Wang, Y. and Mori, G. (2008). Multiple tree models for occlusion and spatial constraints in human pose estimation. In *Computer Vision–ECCV 2008*, pages 710–724. Springer.

[317] Wang, Y., Sabzmeydani, P., and Mori, G. (2007). Semi-latent dirichlet allocation: A hierarchical model for human action recognition. In *Human Motion–Understanding, Modeling, Capture and Animation*, pages 240–254. Springer.

[318] Wang, Y., Tran, D., and Liao, Z. (2011c). Learning hierarchical poselets for human parsing. In *CVPR*.

[319] Wei, X. K. and Chai, J. (2009). Modeling 3d human poses from uncalibrated monocular images. In *ICCV*.

[320] Weinland, D. and Boyer, E. (2008). Action recognition using exemplar-based embedding. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–7.

[321] Weinland, D., Boyer, E., and Ronfard, R. (2007). Action recognition from arbitrary views using 3d exemplars. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–7.

[322] Weinland, D., Ronfard, R., and Boyer, E. (2006). Free viewpoint action recognition using motion history volumes. *Computer vision and image understanding*, 104(2âĂŞ3):249 – 257.

[323] Weinland, D., Ronfard, R., and Boyer, E. (2011). A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 115(2):224 – 241.

[324] Weng, J., Cohen, P. R., and Herniou, M. (1992). Camera calibration with distortion models and accuracy evaluation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14:965–980.

[325] Willems, G., Tuytelaars, T., and Gool, L. (2008). An efficient dense and scale-invariant spatio-temporal interest point detector. In *ECCV*, pages 650–663.

[326] Wilson, A. D. and Bobick, A. F. (1999). Parametric hidden markov models for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(9):884–900.

[327] Wolf, C., Lombardi, E., Mille, J., Celiktutan, O., Jiu, M., Dogan, E., Eren, G., Baccouche, M., DellandrÃl'a, E., Bichot, C., Garcia, C., and Sankur, B. (2014). Evaluation of video activity localizations integrating quality and quantity measurements. *CVIU*, 127:14 – 30.

[328] Wolf, C., Taylor, G. W., and Jolion, J.-M. (2010). Learning individual human activities from short binary shape sequences. Technical Report RR-LIRIS-2010-010, LIRIS UMR 5205 CNRS/INSA de Lyon/UniversitÃl' Claude Bernard Lyon 1/UniversitÃl' LumiÃÍre Lyon 2/ÃL'cole Centrale de Lyon.

[329] Wong, S. F., Kim, T. K., and Cipolla, R. (2007). Learning motion categories using both semantic and structural information. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6.

[330] Wrobel, B. P. (2001). Multiple view geometry in computer vision. *KI*, 15:41.

[331] Wu, Q., Wang, Z., Deng, F., Chi, Z., and Feng, D. D. (2013). Realistic human action recognition with multimodal feature selection and fusion. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 43(4):875–885.

[332] Wu, X., Xu, D., Duan, L., and Luo, J. (2011). Action recognition using context and appearance distribution features. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 489–496.

[333] Xia, L., Chen, C., and Aggarwal, J. (2012). View invariant human action recognition using histograms of 3d joints. In *CVPRW*, pages 20–27.

[334] Xiaohan Nie, B., Xiong, C., and Zhu, S.-C. (2015). Joint action recognition and pose estimation from video. In *CVPR*.

[335] Xu, G. and Zhang, Z. (2013). *Epipolar geometry in stereo, motion and object recognition: a unified approach*, volume 6. Springer Science & Business Media.

[336] Xu, X. and Li, B. (2007). Learning motion correlation for tracking articulated human body with a rao-blackwellised particle filter. In *ICCV*, pages 1–8.

[337] Yacoob, Y. and Black, M. J. (1998). Parameterized modeling and recognition of activities. In *Computer Vision, 1998. Sixth International Conference on*, pages 120–127.

[338] Yamato, J., Ohya, J., and Ishii, K. (1992). Recognizing human action in time-sequential images using hidden markov model. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference on*, pages 379–385.

[339] Yan, P., Khan, S. M., and Shah, M. (2008). Learning 4d action feature models for arbitrary view action recognition. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–7.

[340] Yang, Y. and Ramanan, D. (2011). Articulated pose estimation with flexible mixtures-of-parts. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1385–1392. IEEE.

[341] Yang, Y. and Ramanan, D. (2013). Articulated human detection with flexible mixtures of parts. *IEEE T. on PAMI*, 35(12):2878–2890.

[342] Yao, A., Gall, J., Fanelli, G., and Gool, L. V. (2011). Does human action recognition benefit from pose estimation? In *BMVC*.

[343] Yao, B. and Fei-Fei, L. (2010). Modeling mutual context of object and human pose in human-object interaction activities. In *CVPR*.

[344] Yeung, S., Russakovsky, O., Jin, N., Andriluka, M., Mori, G., and Li, F. (2015). Every moment counts: Dense detailed labeling of actions in complex videos. *CoRR*, abs/1507.05738.

[345] Yilmaz, A. and Shah, M. (2005a). Actions sketch: a novel action representation. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 984–989 vol. 1.

[346] Yilmaz, A. and Shah, M. (2005b). Recognizing human actions in videos acquired by uncalibrated moving cameras. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 150–157 Vol. 1.

[347] Yu, J., Guo, Y., Tao, D., and Wan, J. (2015). Human pose recovery by supervised spectral embedding. *Neurocomputing*, 166:301 – 308.

[348] Zanfir, M., Leordeanu, M., and Sminchisescu, C. (2013). The Moving Pose: An Efficient 3D Kinematics Descriptor for Low-Latency Action Recognition and Detection. In *ICCV*.

[349] Zelnik-Manor, L. and Irani, M. (2001). Event-based analysis of video. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages II–123–II–130 vol.2.

[350] Zhang, D. and Shah, M. (2015). Human pose estimation in videos. In *ICCV*, pages 2012–2020.

[351] Zhang, N., Paluri, M., Ranzato, M., Darrell, T., and Bourdev, L. D. (2014). Panda: Pose aligned networks for deep attribute modeling. *CoRR*, abs/1311.5591.

[352] Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22:1330–1334.

[353] Zhang, Z. (2012). Microsoft kinect sensor and its effect. *IEEE MultiMedia*, 19(2):4–10.

[354] Zhang, Z., Hu, Y., Chan, S., and Chia, L.-T. (2008). Motion context: A new representation for human action recognition. In *European Conference on Computer Vision*, pages 817–829. Springer.

[355] Zhang, Z., Wang, C., Xiao, B., Zhou, W., Liu, S., and Shi, C. (2013). Cross-view action recognition via a continuous virtual path. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[356] Zhao, T. and Nevatia, R. (2002). 3d tracking of human locomotion: a tracking as recognition approach. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 1, pages 546–551 vol.1.

[357] Zhou, X., Zhu, M., Leonardos, S., Derpanis, K., and Daniilidis, K. (2015). Sparseness meets deepness: 3d human pose estimation from monocular video. *arXiv preprint arXiv:1511.09439*.

[358] Zhu, A., Snoussi, H., Wang, T., and Cherouat, A. (2015). Human pose estimation with multiple mixture parts model based on upper body categories. *J. Electronic Imaging*, 24:043021.

[359] Zuffi, S. and Black, M. (2015). The stitched puppet: A graphical model of 3d human shape and pose. In *CVPR*, pages 3537–3546.

FOLIO ADMINISTRATIF

THESE DE L'UNIVERSITE DE LYON OPEREE AU SEIN DE L'INSA LYON

NOM : DOGAN DATE de SOUTENANCE : 07/07/2017

Prénoms : Emre

TITRE : Estimation de pose humaine et reconnaissance d'action par un système multi-robots

NATURE : Doctorat Numéro d'ordre : 2017LYSEI060

Ecole doctorale : EDA 512, INFORMATIQUE ET MATHEMATIQUES

Spécialité : Informatiques

RESUME : L'estimation de la pose humaine et la reconnaissance des activités humaines sont des étapes importantes dans de nombreuses applications comme la robotique, la surveillance et la sécurité, etc. Actuellement abordées dans le domaine, ces tâches ne sont toujours pas résolues dans des environnements non-coopératifs particulièrement. Ces tâches admettent de divers défis comme l'occlusion, les variations des vêtements, etc. Les méthodes qui exploitent des images de profondeur ont l'avantage concernant les défis liés à l'arrière-plan et à l'apparence, pourtant, l'application est limitée pour des raisons matérielles. Dans un premier temps, nous nous sommes concentrés sur la reconnaissance des actions complexes depuis des vidéos. Pour ceci, nous avons introduit une représentation spatio-temporelle indépendante du point de vue. Plus précisément, nous avons capturé le mouvement de la personne en utilisant un capteur de profondeur et l'avons encodé en 3D pour le représenter. Un descripteur 3D a ensuite été utilisé pour la classification des séquences avec la méthodologie bag-of-words. Pour la deuxième partie, notre objectif était l'estimation de pose articulée, qui est souvent une étape intermédiaire pour la reconnaissance de l'activité. Notre motivation était d'incorporer des informations à partir de capteurs multiples et de les fusionner pour surmonter le problème de l'auto-occlusion. Ainsi, nous avons proposé un modèle de flexible mixtures-of-parts multi-vues inspiré par la méthodologie classique de structure pictural. Nous avons démontré que les contraintes géométriques et les paramètres de cohérence d'apparence sont efficaces pour renforcer la cohérence entre les points de vue, aussi que les paramètres classiques. Finalement, nous avons évalué ces nouvelles méthodes sur des datasets publics, qui vérifie que l'utilisation de représentations indépendantes de la vue et l'intégration d'informations à partir de points de vue multiples améliore la performance pour les tâches ciblées dans le cadre de cette manuscrit.

MOTS-CLÉS : Activity recognition, articulated pose estimation, multi-view settings

Laboratoire (s) de recherche : IMAGINE

Directeur de thèse: Atilla BASKURT
Co-directeur de thèse: Christian WOLF
Co-directeur de thèse: Gonen EREN

Président de jury : Pellerin, Denis

Composition du jury :

Fofi, David	PRU, Université de Bourgogne	Rapporteur
Vincent, Nicole	PRU, Université Paris Descartes	Rapporteure
Pellerin, Denis	PRU, Polytechnique de Grenoble	Examinateur
Ducottet, Christophe	PRU, Université Jean Monnet	Examinateur
Teulière, Céline	MC, Université Clermont-Ferrand	Examinatrice
Baskurt, Atilla	PRU, INSA Lyon	Directeur de thèse
Wolf, Christian	MC/HDR, INSA Lyon	Co-directeur de thèse
Eren, Gonen	MC, Université Galatasaray	Co-directeur de thèse