3D Motion Data aided Human Action Recognition and Pose Estimation

A dissertation submitted by **Wenjuan Gong** at Universitat Autònoma de Barcelona to fulfil the degree of **Doctor en Informàtica**.

Bellaterra, February 2013

Director	**Dr. Jordi Gonzàlez i Sabaté** Centre de Visió per Computador Dept. de Ciències de la Computació, Universitat Autònoma de Barcelona
Co-director	**Dr. Xavier Roca i Marvà** Dept. de Ciències de la Computació, Universitat Autònoma de Barcelona Centre de Visió per Computador

This document was typeset by the author using LaTeX 2_ε.

The research described in this book was carried out at the Computer Vision Center, Universitat Autònoma de Barcelona.

ISBN 978-84-940530-6-1

Printed by Ediciones Gráficas Rey, S.L.

To my family,

and all those who made a better me.

Acknowledgements

I would like to thank my supervisor Dr. Jordi Gonzàlez i Sabaté and all other members in the Image Sequence Evaluation group leaded by Dr. Xavier Roca i Marvà for providing their consistent support in my Phd study. Without their guidance, advises, and help along the way, I would never possibly be able to make it. Thanks to their considerations of my further career, I was given precious opportunities of collaborating with peers and professors in other institutes, including Fraunhofer-IOSB, Aalborg University and Chinese Academy. I would also like to thank Prof. Andrew D. Bagdanov as a short-term supervisor and also as a friend. He gave me so much courage in research and affected me with his passionate way of working.

Other than those knowledge that I learned from my supervisor and co-supervisors, I benefit tremendously from the colleagues and professors who I have collaborated with: Phd student Jürgen Brauer and his supervisor Dr. Michael Arens in Fraunhofer-IOSB, Germany, who guided and assisted my work there and taught me how to solve problems on solid reasoning; Prof. Preben Fihl and Prof. Thomas B. Moeslund in Aalborg University, Denmark, who encouraged daring ideas, provided in-depth discussions and organized barbecues; Dr. Yongzhen Huang and Prof. Wang Liang in Chinese Academy, who gave me an overall perspective on my research area and help to push my limit to a new level; my colleague Nataliya Shapovalova, who collaborated with me in Pascal Challenge 2010 and kindly left me the winner T-shirt; and master student Adela Bărbulescu who worked with me in her master project and fun exploring. Specially, I would like to thank Ariel Amato in Image Sequence Evaluation group for providing me background subtraction tool which is an indispensable module of this work.

I would also like to thank Dr. François Brémond, Dr. Grégroy Rogez and Dr. El-hadi Zahzah for taking their valuable time reviewing my thesis and sparing their value time to come to CVC as my thesis evaluators. I am also very thankful to my European Mention evaluators: Dr Ralph Martin and Dr Roberto Vezzani. In the last episode of my Phd study, I am not able to forget those who lead my way and helped me so much in the middle: my supervisor in my master study, Prof. Changhe Tu in Shandong University, who showed me how to be a restrict and honest researcher and led me into the research area; Prof. Ralph Martin in Cardiff University, who helped me a lot in applying Phd study opportunities; Gisele Kohatsu, Montse Culleré and all other staff in CVC administration who assisted me to prepare countless documents for coming to Spain and legally staying here.

I always owe my thanks to my dearest family and best friends. My father An

Gong, my mother Juying Pan and my brother Lei Gong, who gave me their endless love and care, are the reason that I keep fighting to become stronger. Their company and emotional support stimulate me to become a better self. My best friends Noha Elfiky and Happy Five group members, who shared so many precious moments in my life, are always there during good or bad times. And José M. Álvarez, who is special to me, gives me the courage to overcome difficulties. Lastly but not the least, I would like to thank my colleagues in Computer Vision Center, Barcelona, Spain: Hilda Caballero, Ahmed Mounir, Hongxing Gao, Nùria Ciera Turigues, Xu Hu, Hany SalahEldeen, Shida Beigpour, Dr. Jorge Bernal del Nozal, Dr. Pep Gonfaus, Dr. Fernando Barrera, David A. Rojas, Javier M. Tur for their help in my daily work and entertainment during coffee breaks; Dr. Carles Fernandez Tena, Dr. Pau Baiget, Dr. Marco Pedersoli who were senior Phd students in Computer Vision Center and gave me wise advises, my Chinese friends in Barcelona: Mengye Han, Su Yan, Ying Li, Shuzhen Li, Zhan Zhao and my friends in China: Haiyan Lv, Jie Lin, Xiuhong Song and all those who gave a helping hand but I forgot to mention.

Abstract

In this work, we explore human action recognition and pose estimation problems. Different from traditional works of learning from 2D images or video sequences and their annotated output, we seek to solve the problems with additional 3D motion capture information, which helps to fill the gap between 2D image features and human interpretations.

We first compare two different schools of approaches commonly used for 3D pose estimation from 2D pose configuration: modeling and learning methods. By looking into experiments results and considering our problems, we fixed a learning method as the following approaches to do pose estimation. We then establish a framework by adding a module of detecting 2D pose configuration from images with varied background, which widely extend the application of the approach. We also seek to directly estimate 3D poses from image features, instead of estimating 2D poses as a intermediate module. We explore a robust input feature, which combined with the proposed distance measure, provides a solution for noisy or corrupted inputs. We further utilize the above method to estimate *weak poses*,which is a concise representation of the original poses by using dimension deduction technologies, from image features. *Weak pose* space is where we calculate vocabulary and label action types using a bog of words pipeline. Temporal information of an action is taken into consideration by considering several consecutive frames as a single unit for computing vocabulary and histogram assignments.

To validate the proposed methods, we use HumanEva data set, IXMAS data set and TUM kitchen data set. The experiments we conducted includes: compare the performances of modeling and learning methods for estimating 3D poses from 2D poses with the training set of HumanEva data set and TUM kitchen data set under different conditions, like different performers, different viewpoint, different action types and so on; using state-of-art body part detectors, we detect 2D pose configurations from HumanEva data set and take 2D pose configurations as inputs for the pose estimation framework, which was validated with HumanEva data set; compare several popular input features for describing silhouettes for a learning method in pose estimation problem and with the feature that scores the best performance, we compare the performance of the most robust feature with the proposed feature combined with the distance measure; for action recognition, we use cross validation to fix the dimension of *weak poses* and the size of temporal steps; also in action recognition experiments, we compare action recognition accuracies from only 2D image features and incorporating 3D motion information.

From the work, we conclude that 3D motion data, which solve the ambiguity of 2D representation itself, could be utilized directly for accurate pose estimation and aids to enhance action recognition accuracies from 2D image sequences compared with using solely 2D image features. In our future work, we would like to explore how to improve the mapping mechanism from feature space to pose or action space that would hopefully fill the semantic gap.

Resum

En aquest treball s'explora el reconeixement d'accions humanes i la estimaci de la seva postura en seqncies dimatges. A diferncia de les tcniques tradicionals d'aprenentatge a partir d'imatges 2D o vdeo amb la sortida anotada, en aquesta Tesi abordem aquest objectiu amb la informaci de moviment 3D capturat, que ens ajudar a tancar el lla entre les caracterstiques 2D de la imatge i les interpretacions sobre el moviment hum.

En primer lloc, es comparen dos enfocaments diferents tpicament aplicats per a obtenir lestimaci de la posici 3D a partir de les configuracions 2D de la imatge: mtodes basats en la modelitzaci o basats en l'aprenentatge de moviment. Comparant i avaluant els resultats, es determina continuar amb els mtodes basats en laprenentatge de moviment per trobar estratgies de millora del seu rendiment. De fet, sestableix a continuaci un marc de treball afegint un mdul de detecci de parts 2D del cos hum per a refinar les estimacions de la postura 3D. Ls del mdul de detecci ens permet generalitzar el nostre mtode a entorns amb el fons no controlat. A continuaci passem a estimar directament la configuraci de la postura 3D a partir de la imatge, en comptes d'estimar la postura 2D en algun mdul intermedi, com es fa tpicament. Aix, avaluem un conjunt de descriptors robustos de la imatge, els quals combinats amb una nova mesura de distncia proposada en aquesta Tesi, permet obtenir resultats menys sorollosos o erronis. Amb aquests resultats, passem a avaluar com podem estimar postures dbils, o representacions molt compactes i redudes de la postura complerta original, obtingudes mitjanant tcniques de reducci de la dimensionalitat. s en lespai de postures dbils on calculem el vocabulari i les etiquetes de les accions humanes, procs estndard en el sistemes bags-of-words com aquest. Es t en compte la informaci temporal duna acci considerant diversos frames consecutius com a una nica unitat atmica per calcular el vocabulari i la seva assignaci a la representaci final basada en histogrames.

Per a validar els mtodes proposats, shan considerat les bases de dades HumanEva, IXMAS i TUM-Kitchen. Els experiments inclosos en aquesta Tesi comparen exhaustivament els resultats dutilitzar una estratgia de modelitzaci o una daprenentatge sota diferents condicions, com lactor, lescena, el punt de vista, lacci, etc. Aix, sanalitza el resultat dincorporar un dels detectors ms utilitzats en la literatura per localitzar les parts 2D del cos hum en imatges, per detectar robustament configuracions 2D de la postura. Tamb validem diferents descriptors populars dimatges per a que un mtode basat en laprenentatge pugui seleccionar aquell descriptor que li permet en cada moment obtenir el millor rendiment. Per a avaluar el nostre mtode de reconeixement daccions humanes, utilitzem validaci creuada per a fixar el nmero de dimensions necessari per a lespai de postures dbils i per calcular el pas temporal. Per ltim, incorporem moviment 3D per comparar els resultats del reconeixement daccions utilitzant nicament descriptors 2D.

A partir dels resultats obtinguts, es conclou que la utilitzaci de descriptors de moviment 3D, que de fet ja sutilitzen per solucionar la ambigitat inherent de les representacions 2D, pot ser una bona alternativa per a obtenir una estimaci acurada i robusta de la postura humana. De la mateixa manera, ens pot ajudar a millorar la precisi del reconeixement daccions humanes en seqncies dimatges 2D. Com a treball futur, es proposa explorar els mecanismes de mapeig des de lespai de caracterstiques

de la imatge a un espai de postures o accions humanes que ens ajuda a omplir la bretxa semntica.

Resumen

En este trabajo se exploran el reconocimiento de acciones humanas y la estimacin de su postura en secuencias de imgenes. A diferencia de las tcnicas tradicionales de aprendizaje a partir de imgenes 2D o vdeo con la salida anotada, en esta Tesis abordamos este objetivo con la informacin de movimiento 3D capturado, que nos ayudar a cerrar el lazo entre las caractersticas 2D de la imagen y las interpretaciones sobre el movimiento humano.

En primer lugar, se comparan dos enfoques diferentes tpicamente aplicados para obtener la estimacin de la posicin 3D a partir de las configuraciones 2D de la imagen: mtodos basados en la modelizacin o basados en el aprendizaje de movimiento. Comparando y evaluando los resultados, se determina continuar con los mtodos basados en el aprendizaje de movimiento para encontrar estrategias de mejora de su rendimiento. De hecho, establece a continuacin un marco de trabajo aadiendo un mdulo de deteccin de partes 2D del cuerpo humano para refinar las estimaciones de la postura 3D. El uso del mdulo de deteccin nos permite generalizar nuestro mtodo a entornos con el fondo no controlado. A continuacin pasamos a estimar directamente la configuracin de la postura 3D a partir de la imagen, en lugar de estimar la postura 2D en algn mdulo intermedio, como se hace tpicamente. As, evaluamos un conjunto de descriptores robustos de la imagen, los cuales combinados con una nueva medida de distancia propuesta en esta Tesis, permite obtener resultados menos ruidosos o errneos. Con estos resultados, pasamos a evaluar cmo podemos estimar posturas dbiles, o representaciones muy compactas y reducidas de la postura completa original, obtenidas mediante tcnicas de reduccin de la dimensionalidad. Es en el espacio de posturas dbiles donde calculamos el vocabulario y las etiquetas de las acciones humanas, proceso estndar en sistemas bags-of-words como ste. Se tiene en cuenta la informacin temporal de una accin considerando diversos frames consecutivos como una nica unidad atmica para calcular el vocabulario y su asignacin a la representacin final basada en histogramas.

Para validar los mtodos propuestos, se han considerado las bases de datos HumanEva, IXMAS y TUM-Kitchen. Los experimentos incluidos en esta Tesis comparan exhaustivamente los resultados de utilizar una estrategia de modelizacin o una de aprendizaje bajo diferentes condiciones, como el actor, la escena, el punto de vista, la accin, etc. As, se analiza el resultado de incorporar uno de los detectores ms utilizados en la literatura para localizar las partes 2D del cuerpo humano en imgenes, para detectar robustamente configuraciones 2D de la postura. Tambin validamos diferentes descriptores populares de imgenes para que un mtodo basado en el aprendizaje pueda seleccionar aquel descriptor que le permite en cada momento obtener el mejor rendimiento. Para evaluar nuestro mtodo de reconocimiento de acciones humanas, utilizamos validacin cruzada para fijar el nmero de dimensiones necesario para el espacio de posturas dbiles y para calcular el paso temporal. Por ltimo, incorporamos movimiento 3D para comparar los resultados del reconocimiento de acciones utilizando nicamente descriptores 2D.

A partir de los resultados obtenidos, se concluye que la utilizacin de descriptores de movimiento 3D, que de hecho ya se utilizan para solucionar la ambigedad inherente de las representaciones 2D, puede ser una buena alternativa para obtener una estimacin

precisa y robusta de la postura humana. De la misma manera, nos puede ayudar a mejorar la precisin del reconocimiento de acciones humanas en secuencias de imgenes 2D. Como trabajo futuro, se propone explorar los mecanismos de mapeo desde el espacio de caractersticas de la imagen a un espacio de posturas o acciones humanas que nos ayuda a llenar la brecha semntica.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Human action recognition and pose estimation have been intensively studied due to their wide applications in security surveillance, video indexing and human computer interaction in video games albeit intrinsic hard and challenging. In surveillance systems installed in places requiring high security, such as banks, human action recognition can be applied to detect abnormal human actions and potentially dangerous situations before they become truly dangerous. Human action characterization is also making inroads in the area of security and safety monitoring. Behavior analysis systems are being built to monitor the safety of children and the elderly, and in such scenarios, abnormal action detection can be used to detect dangerous situations like falling down. Automatic video indexing for video and image libraries can be enhanced using human action recognition and by allowing semantics-based access to multimedia content. Human pose estimation is applicable in similar areas. Human action recognition and pose estimation are also applied in human computer interaction, where estimation results can be used as noninvasive control signals so computers can react accordingly.

Despite increased interest in recent years, human action recognition and pose estimation remain challenging problems. Due to their close relation and resemblance albeit differences inhabit in these two problems, we explore the possible enhancements of these two problems. Although the improvement of one single problem is difficult, we explore these two problems together because we believe that the effective solution of one problem aids to improve the solution of another.

1.1 Problem Formulation

We can frame the problem of action recognition in computer vision as following:

$$\rho(\mathbf{X}, \mathbf{T}) = Y, \tag{1.1}$$

where $\mathbf{X} = \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ is the set of features extracted from images or image sequences, $\mathbf{T} = \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_n$ is the set of temporal information, if any, extracted from image sequences, n is the number of training samples and Y is the annotated labels

which specifies the action labels. And pose estimation problem can be framed as following:

$$\rho(\mathbf{X}) = Y, \tag{1.2}$$

where $\mathbf{X} = \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ is the set of features extracted from images and $Y = \mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n$ is the set of annotated human poses and every sample point y_i in the target space represents a configuration of human body limbs whose dimension should be no less than the degree of freedom of human motion.

Figure 1.1: Example images from [50]. From left to right, actions are running, walking, kicking, crouching, throwing, and catching.

The challenges are due to the following defects:

1. The flexibility of the human body results in a huge set of possible human poses and the different styles of performing the same actions by different performer might also results different human body poses of the same semantic meaning. Figure 1.1 and figure 1.2 shows examples from two data set with various human poses and human actions.

2. Lack of depth information in 2D images or video sequences. While actions and poses are performed in 3D space and losing this information could cause confusions or even failures, most of available data set for validating these problems only contain 2D images or video sequences, for example, Weizmann action data set, Hollywood human action data set or Pascal action recognition data set due to the fact that it is easier and less expensive to collect images or image sequences. We argue that addition of 3D motion data aid to enhance action recognition accuracy. Note there are also a few data set like CMU data set and HumanEva data set which provide 3D motion data. And for pose estimation problems, introducing 3D motion data is useful for reducing ambiguous poses, like left or right leg confusion. If we correlate with the formulation in equation 1.1 and equation 1.2, in the case of without 3D motion information, variable X in the equation is not the ideal space where we want to frame our solution.

3. Illumination changes and background jitter. Although these are also a crucial reason that poison the estimation results, we are not concentrating on resolving this problem in our current work. In our experiment, we don't need to tackle these problems because the validation data set are explore are recoded in indoor surveillance environment.

Figure 1.2: Example images from a human interaction data set [29]. Images from the first row to the fourth correspond to the four action classes: shaking hands, hugging, kissing and punching.

4. A proper processing and a suitable mapping mechanism to transform the feature space into target space, ρ in equation 1.1 and equation 1.2. This is intrinsically very hard problem, for example, for action recognition, this function is supposed to fill the gap between the features represented as digits and human interpretations. For pose estimation problem, it is relatively easier mathematically formulated due to the possible precise descriptions of human poses, but researchers are still struggling to find a good transformation that is practical for resolving the curse of dimensions resulted from high dimensions of freedoms of human bodies. Relating to equation 1.1 and equation 1.2,

1.2 Precedent Works and Inspirations

The origin of the above mentioned challenges can be traced back to the gap between the human interpretation of actions and poses and the digital representation of images. For example, while most of the researchers solve computer vision problems in

a Euclidean space, there are researchers who believe human vision is better modeled in a Riemannian space [58, 74, 62]. Here we don't aim to answer the question of which is a better space to model human vision, but we should bear in mind that this divergence in modeling space is critical and may shed a light in solutions to many computer vision problems including human action recognition and pose estimation. Since until now there is not a definitely conclusion in which space human vision is actually performed in, all our work is carried out in Euclidean space for simplicity of data representations.

Although we are not concentrating on the physiology of human vision, we seek additional information to minimize the gap between the output model and the real output data by introducing 3D motion data. For pose estimation problems, the advantage of 3D pose representation over 2D pose representation is obvious: unambiguous limb layout, direct applications for 3D visualization in Computer Graphics and so on. For example, 2D pose detection [89], which defines the left or right body part by their positions in the images (that is, body parts showed in the leftmost of the human blob is defined as left body parts without considering the human's facing direction), struggles to distinguish the left leg from the right one. This simplifies the solution, but may cause inconsistent among body parts, for example, a left leg appearing on the right side of the right leg would be labeled as the right leg, while the right arm appearing on the right side of the left arm would be labeled as the right arm which is semantically not from the same side of the detected right leg. Examples are showed in figure 1.3. This might result in ambiguous indexing in further applications.

In our work, we resort to 3D motion data for pose estimation and action recognition. We believe that 3D motion data, which is closer to human experience in the real world aids to solving action recognition and pose estimation problems. Human pose estimation allows for a wide field of applications such as video search, visual surveillance and human computer interfaces used e.g. in video games. Full body 3D human pose estimation from monocular images is a difficult problem since the depth information is lost when projecting from 3D space to 2D image plane. For this, a huge set of approaches have been suggested to recover the 3D pose based on monocular images.

One class of approaches tries to map image features directly to 3D poses. For example, Agarwal and Triggs [2] use a grid of local gradient orientation histograms, *i.e.* a dense sampling of interest points, and learn a mapping to 3D poses using direct regression. Another class of approaches first tries to map image features to 2D poses and then maps 2D pose estimates to 3D poses. For example, Andriluka etal. [7] first identify consistent sequences of 2D poses (called 'tracklets') and formulate the 3D pose estimation problem within a Bayesian framework while the prior probability of 3D poses is modeled using a hierarchical Gaussian process latent variable model.

For the later class there exist two subclasses that differ in the way in which 2D poses are lifted to 3D poses. Learning approaches try to learn this mapping using training examples and adapt some mapping using e.g. support vector machine, relevance vector machine [3], or Gaussian process Regressors. Modeling approaches try to model this mapping from 2D to 3D poses explicitly by using knowledge about the inverse of the 3D to 2D mapping. Although the learning and modeling approaches are quite different by concept for the 2D to 3D lifting task, it has not yet been in-

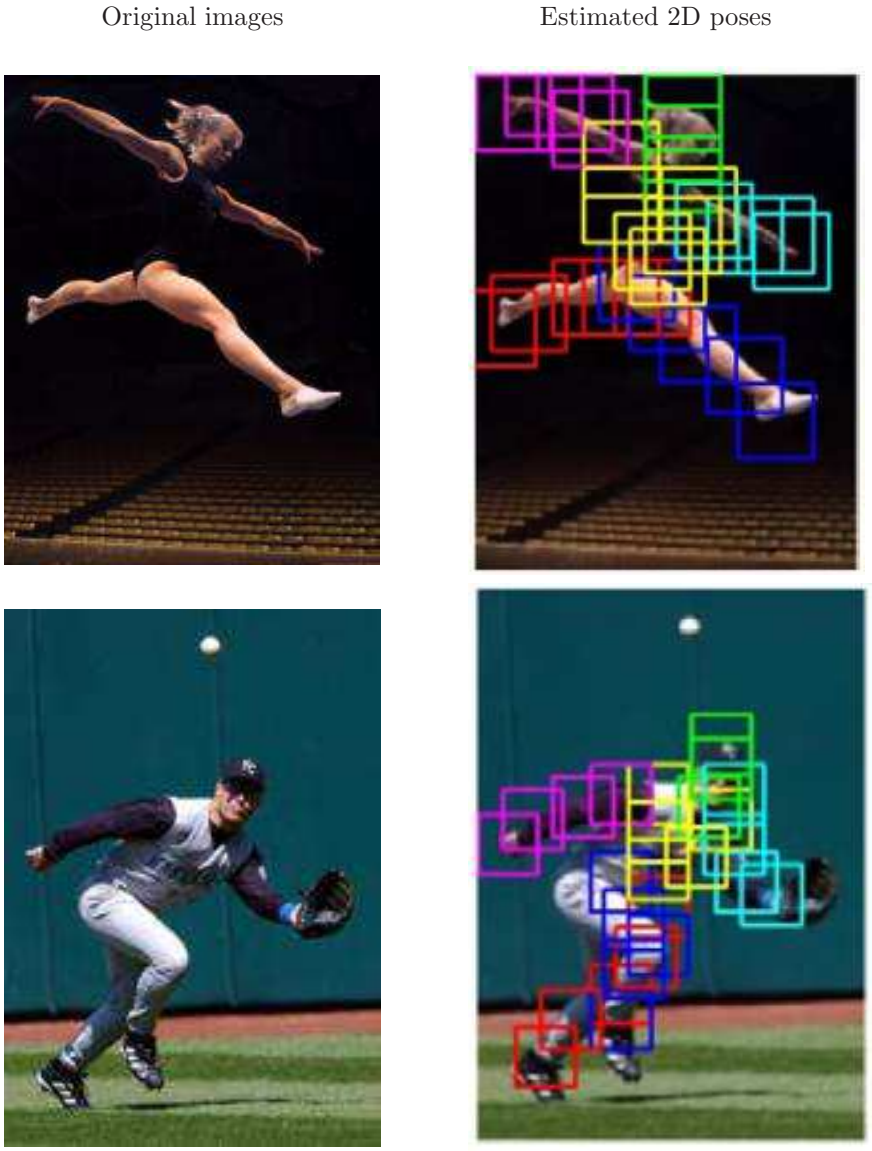


Figure 1.3: Examples of ambiguous 2D pose estimation.

vestigated systematically how the two classes of approaches differ and what are the advantages of each class.

One part of our work is to close this gap. For this, we present a systematic evaluation by choosing a typical representative method of each class and compare their 3D pose reconstruction performance directly using the same 2D input data. For the class of modeling approaches we choose a geometric reconstruction approach –

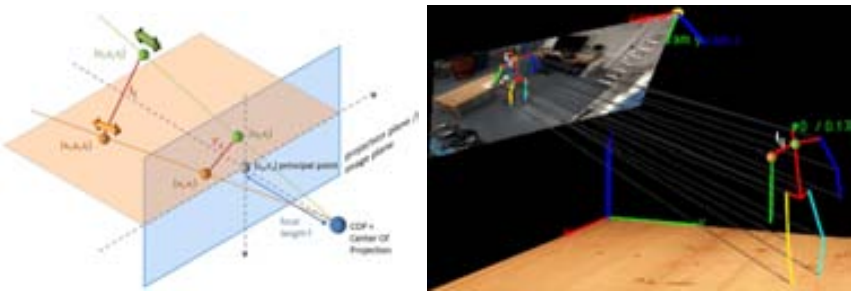


Figure 1.4: Geometric reconstruction of 3D poses. Using the foreshortening information of projected limb lengths l'_{ij} we can reconstruct the displacement in $\Delta_z = z_1 - z_2$ in z direction even for perspective camera models.

originally presented for a restricted parallel projection camera model [71], used in several following works (e.g. [34], [48]), and recently extended to a realistic perspective projection camera model [18]. Refer to chapter 3 for detailed explanation of the geometric method. For the class of learning approaches we choose the Gaussian process regression since it is successfully used in many pose estimation works (e.g. [78]). Support vector machine and relevance vector machine [3] are more efficient in training as they are picking the most representative training samples for the model. But due to a better predicting accuracy, we choose Gaussian process regressor. We evaluate both methods on the TUM kitchen and the HumanEva data set. Figure 1.4 shows a visualized illustration of a modeling method.

Based on the previous comparison results, we choose the learning method as the main approach for 2D poses to 3D poses mapping or later on 2D image features to 3D poses mapping. In order to deal with realistic situation, instead of simulated or 2D pose ground truth data, we introduce a module of 2D pose detection with a state-of-art method [89]. Despite the ambiguous indexing problem mentioned above, this is a robust and efficient method which deals with varied images even with cluttered backgrounds. The idea of the method is to learn a mixture model for each body part and represent the human body with a tree of these body parts. The detected human pose is optimized by calculating the best configuration of all candidate body parts with dynamic programming. After adding this 2D pose detector to our regression method, we set up a framework for 3D pose estimation from monocular image with varied backgrounds.

One important factor to close up the gap between human interpretation and image representation is accurately and sufficiently extracted image features. Take the above mentioned learning method for example, there are bulks of work concentrate on enhancing mapping models by learning output structure [11, 44, 60, 13] for a learning method. But rarely there are works on exploring the robustness of the feature extraction. Another focus of our work is to explore the feature robustness for extracted human silhouettes.

If we avoid using a intermediate layer of 2D pose estimation and estimate 3D poses directly from image features, usually silhouettes are first extracted with background subtraction algorithms. Then suitable input features are extracted from silhouettes

and sent to regression models for training and test. In most cases, extracted silhouettes are noisy due to camouflage and shadows. As we stated before, rarely there are studies on how noisy inputs influence human pose estimation accuracies and how robust are input features against noise. This is mainly due to the fact that traditional distance measures compute squared distance between two feature vectors, where points from camouflage and shadows show no difference from points from body parts.

The most commonly used features for describing extracted human silhouettes include shape context [3, 11], scale invariant feature transform (SIFT) [11], histograms of gradient (HOG) [90] and so on. Different features tend to capture variant attributes. For example, shape context describes point distributions from local points, SIFT combined with bag of words representation describes overall distribution of local features within region of interest, and PHOG portrait local features with location information. As a result, we select these three features. We compare these features based on human pose estimation accuracies on HumanEva data set.

We further propose a new image feature based on Iterative Closest Point algorithm in Computer Graphics. The proposed method is able to automatically discard noise from certain channels of input features. The basic idea is to automatically adjust a threshold value according to the noisy level of input feature, and use this threshold to discard those input channels that are considered as noise. Combined with this new feature, we devise a new distance measure within GPR framework. We test this combination on HumanEva data set and compare with a baseline method of PHOG inputs combined with standard squared exponential kernel in GPR.

Despite of its wide range of applications and the huge number of research works, action recognition from 2D image sequences remains a challenging problem. One of the reasons is due to the high variability of scenarios and situations which can be found in videos, thus resulting in very different image qualities and content. As a result, we need to choose robust features and classification methods which can work well in multiple scenarios and for different actions. While most of the related work are concentrating on exploring different input features and classification methods, few of them explores the use of 3D motion capture data for 2D action recognition. In our work, we will explore this possibility.

To do this, we introduce a module of *weak pose* estimation which explicitly incorporates human pose into action recognition problem. We believe that 3D motion information, after deducting redundant information, can be utilized to improve action recognition accuracies and the experiment results support this hypothesis. Most solutions for Human Action Recognition HAR learn action patterns from sequences of image features like Space-Time Interest Points [37, 63], temporal templates [24], 3D SIFT [64], optical flow [5, 4], Motion History Volume [86], among others. These features are commonly used to describe human actions which are subsequently classified using techniques like Hidden Markov Models [4, 17, 27, 85, 96], and Support Vector Machines [63]. Recent and exhaustive reviews of methods for HAR can be found in [54, 87].

One can categorize the scenarios found in the literature into several groups: single-human action [51], crowds [67], human-human interaction [61], and action recognition in aerial views [22], to cite but a few. Although the method proposed in our work

mainly concentrates on single-human action recognition, our work can be also applied to all the aforementioned scenarios, given that the 2D silhouettes of the agents are extracted from image sequences.

After confining the problem scenario, we tend to search for a solution for action recognition problem that incorporate 3D pose information. One exemplar work of utilizing 3D pose information for action recognition problem is stated in [51]. Authors in [51] propose a model by adding one hidden layer to Conditional Random Fields (CRF) containing pose information. One of the advantages is that every video frame has an action label, so that action segmentation is integrated with action recognition as a whole. However, the optimal number of consecutive frames which contribute to the decision of the action label of the current frame is given by the model. In our proposal, the optimal frame number is calculated from the training data. Also, while authors in [51] use CRFs to model relations between image features and action labels, we label motion sequences with a BoP model, an extension of BoW [39, 38, 15, 80, 31]. We will show that compared with BoW from only 2D image features, the incorporation of *weak poses* improves action recognition accuracy. Also, our method works better than state-of-art method validated on HumanEva dataset.

1.3 Outline of Our Method

Our work is aimed at making a minor step in tackling the challenges in pose estimation and action recognition problems. To resolve these challenges, we assume:

1. the problem of human action recognition is closely related to the problem of human pose estimation and effective solutions to one of the two problems can benefit another. In this work, we only explore the impact of introducing poses as additional information for action recognition.

2. for pose estimation problem, which is also a module in action recognition problem in our approach, the transform form feature space to target space is a linear mapping. In all cases, we model the mapping from extracted features to target poses with Gaussian process regression model which formulates regression as a linear regression.

In this work, we first tackle the problem of pose estimation and based on its modified solution, we further resolve the problem of action recognition.

The outline of our work is summarized as the following:

1. We tackle the problem of 3D human pose estimation based on monocular images from which 2D pose estimates are available. Some of the related works avoid to model the mapping from 2D poses to 3D poses explicitly but learn the mapping using training samples. In contrast, there also exist methods that try to use some knowledge about the connection between 2D and 3D poses to model the mapping from 2D to 3D explicitly. We present a comparison for the most commonly used learning approach for 3D pose estimation – the Gaussian process regressor – with the mostly used modeling approach – the geometric reconstruction of 3D poses. The results show that the learning based approach

outperforms the modeling approach when there are no big changes in viewpoint or action types compared to the training data. In contrast, modeling approaches show advantages over learning approaches when there are big differences between training and application data. With enough possible training poses, a learning methods give better precision, so in the following work, we concentrate on exploring learning methods. Then, we introduce a 2D pose detector which outputs 2D body part configurations from images with cluttered background. Combined with the learning method, the whole pipeline is capable of estimating the 3D pose of a person from single images or monocular image sequences in unconfined environment.

2. We explore the possibilities of estimating human poses directly from image features instead of resorting to 2D pose configurations. The first step of the solution is usually composed of a background subtraction method, where human silhouettes are isolated from the background. Then selected input features extracted from silhouettes and it corresponding output joint positions of this frame are used to train a mapping model. Although silhouettes from background subtraction methods are usually noisy, the effect of noisy inputs to pose estimation accuracies has been barely studied. In our work, we explore this issue: first, we compare several standard image features widely used for human pose estimation for comparing their performances. Second, a novel Iterative Closest Point algorithm is introduced as a filtering process of those foreground pixels which are false positives. Our method, in addition to automatically discard unwanted noise, like camouflage or shadows, allows us to differentiate between different noise levels to assess their effects in pose estimation accuracy.

3. We also present a method for human action recognition from image sequences based on human poses, which were estimated with the method mentioned above. We use 3D human pose data as additional information and propose a compact human pose representation, called a *weak pose*, in a low-dimensional space while still keeping the most discriminative information for a given pose. With predicted poses from image features, we map the problem from image feature space to pose space, where a Bag of Poses model is learned for the final goal of human action recognition. The Bag of Poses model is a modified version of the classical Bag of Words pipeline by building the vocabulary based on the most representative *weak poses* for a given action. Compared with the standard k-means clustering, our vocabulary selection criteria is proven to be more efficient and robust against the inherent challenges of action recognition. Moreover, since for action recognition the ordering of the poses is discriminative, the Bag of Poses model incorporates temporal information: in essence, groups of consecutive poses are considered together when computing the vocabulary and assignment.

The rest of the thesis is organized as following: chapter 2.6 explains the basics of GPR technology used throughout the work, including the definition, its attributes, main covariance matrix types, then we explain how to use GPR to solve non-linear mapping modeling with a simple example, and later on we show the algorithm to utilize GPR for pose estimation, finally, we compare GPR with multi-variate Gaussian, which can be considered a special case of GPR; in chapter 3.6, we show the

comparisons between modeling and learning methods for boosting 2D body part detections to 3D poses, and we explain the proposed method for 3D pose estimation from detected 2D body parts; in chapter 4.4, we propose a new robust feature against input feature noise, which introduce iterative closest point into the feature descriptor; in chapter 5.5, we explore the impact of incorporating estimated poses into action recognition problems, for which we use the standard bag of words pipelines to address; finally, we conclude our work on action recognition and pose estimation and discuss possible further work in chapter 6.

Chapter 2

Gaussian Process Regression Foundations

The problem of predicting 3D human postures from 2D silhouettes is highly non-linear. Gaussian processes have been effectively applied for modeling non-linear dynamics [68, 82, 32]. For example, Gaussian process has been applied to non-linear regression problems, like robot inverse dynamics [21] and nonrigid shape recovery [95]. Variations of Gaussian process model like Gaussian process latent variable model [73, 77, 25], Gaussian process models for structured output [11] and Gaussian process models for multi-task output [44] are also developed by adapting to application requirements.

In original Gaussian process regression (GPR) model, the core part is the definition of the mean and the covariance function. There are some works done on exploring new covariance definitions (kernel functions), like [45]. Also there are some works on multiple kernel learning for Gaussian processes. For example, A. Kapoor etal. [35] propose an algorithm to optimize kernel weights and hyper-parameters simultaneously for Gaussian processes. But the proposed algorithm is designed for pyramid match kernel applied to object categorization problems.

In virtue of the wide applications of Gaussian process regression model, we fix the model for learning mapping from images features to 3D pose data, from 2D poses to 3D poses, and from image features to *weak poses*. In this section, we explain the mechanism, the definition and important parameters of Gaussian process regression model.

2.1 Mechanism

For Gaussian process regression, it assume a linear mapping between the input and the output. By modeling parameters and noise with Gaussian priors, the model can be optimized to suit specific problems. In training, these parameters from the model and noise are learned from training data given pre-defined starting values. In the reference step, a correlation between the test input and each training inputs are calculated and this correlation is employed for deciding the weight of training outputs, which are interpolated to estimate the test output.

Here in training step, we can employ a famous kernel trick to avoid direct defining the mapping function (the mapping parameter in the linear mapping mechanism) and define a kernel (covariance matrix) with input as parameter. In this way, we deal with this kernel instead of model parameters directly, and all the final parameters from the GPR model is called hyper-parameters.

2.1.1 Probability over functions

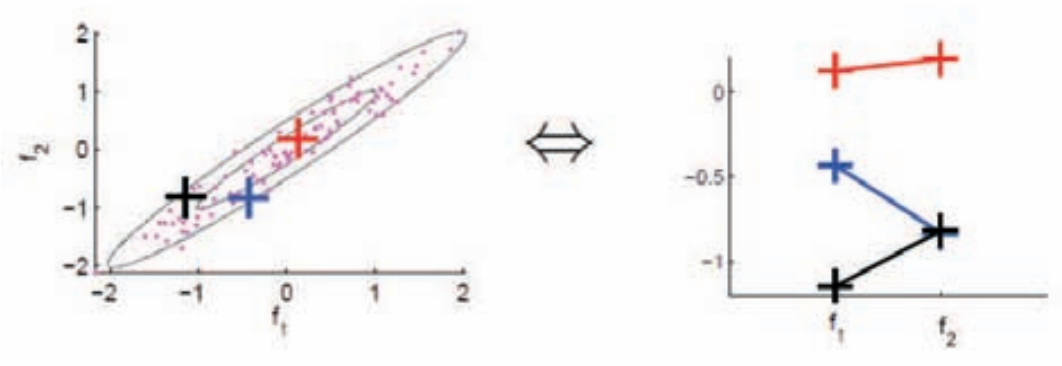


Figure 2.1: GPR with input dimension equals 2 in a function view. The figure shows the correspondence between 2D Gaussian and the function representation. A sample (left) corresponds to a connected line (right). The index of the input dimension (left) corresponds to the$x$ value (right). The value of the input in each dimension (left) corresponds to the y value (right). The probability of the sample point (left) corresponds to the probability of the line (right).

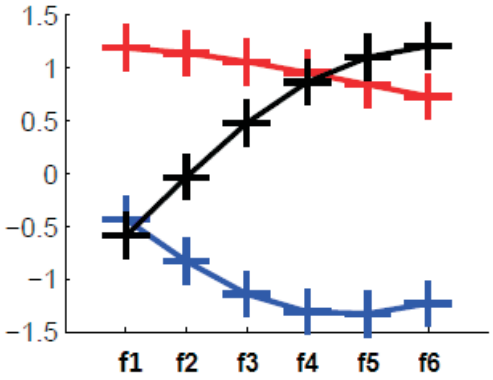


Figure 2.2: GPR with input dimension equals 6 in a function view.

GPR is a collection of random variables, any finite number of which have joint Gaussian distribution. For example, we show examples of random variables of size

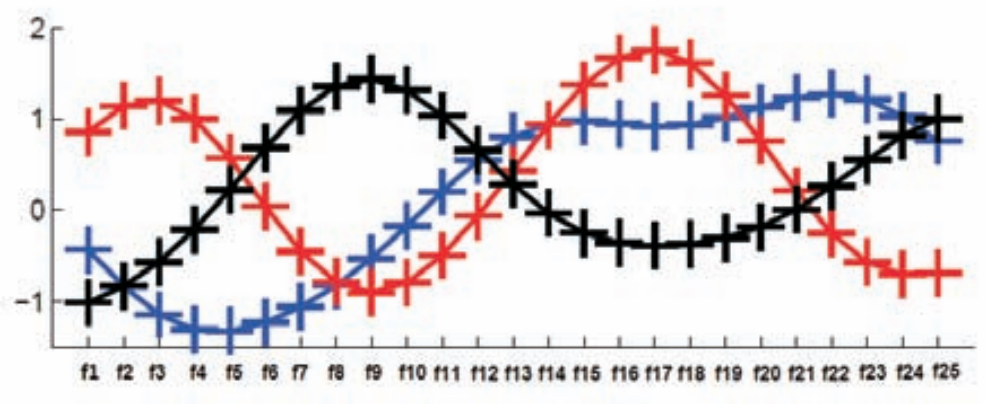


Figure 2.3: GPR with input dimension equals 25 in a function view.

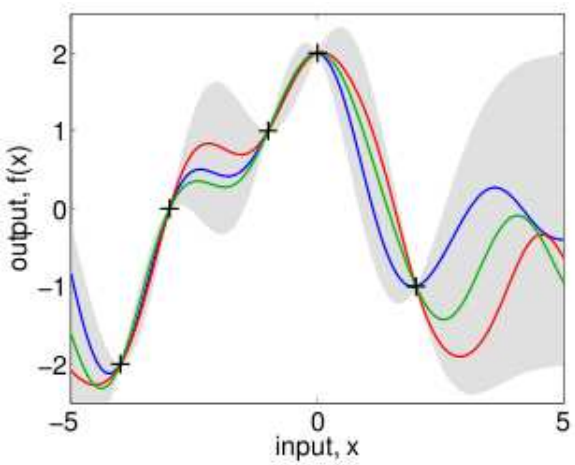


Figure 2.4: Gaussian process model in function viewpoint.

2(figure 2.1), 6(figure 2.2) and 25(figure 2.3). When the input dimensions increase to ∞, the mapping from the input to the output transforms into a function, where the distribution of the output can be considered as a distribution over a function. The functional viewpoint of GPR is showed by extending all input dimensions in GPR along one axis (right sub-figure in figure 2.1). Figure 2.1 shows the correspondence between two representations. A sample (left) corresponds to a connected line (right). The index of the input dimension (left) corresponds to the$x$ value (right). The value of the input in each dimension (left) corresponds to the y value (right). The probability of the sample point (left) corresponds to the probability of the line (right).

2.2 Definition of Gaussian Process Regression

According to [20], Gaussian process is defined as: *a collection of random variables, any finite number of which have (consistent) joint Gaussian distribution.* A Gaussian process is completely specified by its mean function and a covariance function. Integrating with our problem, we denote the mean function as $m(\mathbf{s})$ and the covariance function as $k(\mathbf{s}, \mathbf{s}')$, so a Gaussian process is represented as:

$$\zeta(\mathbf{s}) \sim \mathcal{GP}_j(m(\mathbf{s}), k(\mathbf{s}, \mathbf{s}')), \tag{2.1}$$

where

$$\begin{aligned} m(\mathbf{s}) &= E[\zeta(\mathbf{s})], \\ k(\mathbf{s}, \mathbf{s}') &= E[(\zeta(\mathbf{s}) - m(\mathbf{s}))(\zeta(\mathbf{s}') - m(\mathbf{s}'))], \end{aligned} \tag{2.2}$$

Normally, people set a zero-mean Gaussian process whose covariance is a squared exponential function with two hyperparameters controlling the amplitude θ_1 and characteristic length-scale θ_2:

$$k_1(\mathbf{s}, \mathbf{s}') = \theta_1^2 \exp(-\frac{(\mathbf{s} - \mathbf{s}')^2}{2\theta_2^2}). \tag{2.3}$$

We assume prediction noise as a Gaussian distribution and formulate finding the optimal hyperparameters as an optimization problem. We seek the optimal solution of hyperparameters by maximizing the log marginal likelihood (see [20] for details):

$$\log p(\Psi'|\mathbf{s}, \theta) = -\frac{1}{2}\Psi'^T K_{\Psi'}^{-1}\Psi' - \frac{1}{2}\log|K_{\Psi'}| - \frac{n}{2}\log 2\pi, \tag{2.4}$$

where $K_{\Psi'}$ is the calculated covariance matrix of the target vector (vector of training *weak poses* in *UaSpace*) Ψ' under the kernel defined in equation 5.8.

With the optimal hyperparameters, the prediction distribution is represented as:

$$\begin{aligned} \Psi'^*|\mathbf{s}^*, \mathbf{s}, \Psi' \sim\ & \mathcal{N}(\mathbf{k}(s^*, \mathbf{s})^T[K + \sigma_{noise}^2 I]^{-1}\Psi', \\ & k(s^*, s^*) + \sigma_{noise}^2 - \mathbf{k}(s^*, \mathbf{s})^T[K + \sigma_{noise}^2 I]^{-1}\mathbf{k}(s^*, \mathbf{s})), \end{aligned} \tag{2.5}$$

where K is the calculated covariance matrix from training 2D image features $\mathbf{s}$ and σ_{noise} is the covariance of Gaussian noise.

2.3 Attributes

Equation2.5 for referencing test data is deducted from marginal and conditional properties of Gaussian distributions. The following is the marginal property of Gaussian distributions: the marginals of a joint Gaussian are again Gaussian, that is,

$$p(\mathbf{x}, \mathbf{y}) = \mathcal{N}(\begin{bmatrix} a \\ b \end{bmatrix}, \begin{bmatrix} A & B \\ B^T & C \end{bmatrix}) \quad \Longrightarrow \quad p(\mathbf{x}) = \mathcal{N}(\mathbf{a}, A). \tag{2.6}$$

And the conditional property of Gaussian distributions are: the conditionals of a joint Gaussian are again Gaussian, that is,

$$p(\mathbf{x},\mathbf{y}) = \mathcal{N}(\begin{bmatrix} a \\ b \end{bmatrix}, \begin{bmatrix} A & B \\ B^T & C \end{bmatrix}) \tag{2.7}$$

$$\implies \quad p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{a} + BC^{-1}(\mathbf{y}-\mathbf{b}), A - BC^{-1}B^T). \tag{2.8}$$

Thus we are able to predict the distribution of x given the distribution y.

In most cases, we assume that Gaussian process priors have zero means, that is,

$$f(x)|M_i \sim \mathcal{GP}_j(m(\mathbf{x}) \equiv 0, k(\mathbf{s},\mathbf{s}')). \tag{2.9}$$

This leads to a Gaussian process posterior

$$f(x)|\mathbf{x},\mathbf{y},M_i \sim \mathcal{GP}_j(m_{post}(\mathbf{x}) = k(x,\mathbf{x})[K(x,\mathbf{x}) + \sigma^2_{noise}I]^{-1}\mathbf{y}. \tag{2.10}$$

With this posterior, we only need to define covariance matrices, known as kernel in machine learning community.

The most frequently used covariance matrices (kernels) include: squared exponential (SE), Rational quadratic (RQ), Matérn and Periodic, smooth covariance functions. The function of covariance function is define the distance measure in a newly transformed space where the original data samples have one to one correspondences with their mapped points and due to the transformation, data samples of different attribute classes in the new spaces are easier to classify or identify. The most frequently used covariance matrices in GPR are defined as following:

1. Squared exponential:

$$k(x,x') = \sigma_f^2 exp[\frac{-(x-x')^2}{2l^2}]. \tag{2.11}$$

2. Rational quadratic:

$$k_{RQ}(r) = (1 + \frac{r^2}{2\alpha l^2})^{-\alpha} \tag{2.12}$$

   with $\alpha > 0$ can be seen as a scale mixture (an infinite sum) of squared exponential (SE) covariance functions with different characteristic length-scales.

3. Matérn:

$$k(\mathbf{x},\mathbf{x}') = \frac{1}{\Gamma(\gamma)2^{\gamma-1}}[\frac{\sqrt{2\gamma}}{l}|\mathbf{x}-\mathbf{x}'|]^\gamma K_\gamma(\frac{\sqrt{2\gamma}}{l}|\mathbf{x}-\mathbf{x}'|) \tag{2.13}$$

   where K_γ is the modified Bessel function of second kind of order γ, and l is the characteristic length scale.

4. Periodic, smooth covariance function:

$$k_{periodic}(x,x') = exp(-2sin^2(\pi(x-x'))/l^2) \tag{2.14}$$

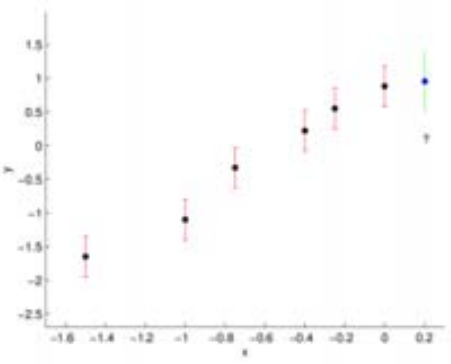


Figure 2.5: A simple problem that can be solved with GPR. Given six noisy data points (errors bars are indicated with vertical lines), we are interested in estimating a seventh at $x_* = 0.2$

2.4 One Simple Example

One example of applying Gaussian process: given six noisy data points showed in figure 2.5, we are interested in estimating a seventh at $x_* = 0.2$. One solution with GPR is showed as following.

Suppose, the mean of this partner GP is zero everywhere and the covariance function, $k(x, x')$ is the "squared exponential",

$$k(x, x') = \sigma_f^2 exp[\frac{-(x - x')^2}{2l^2}], \tag{2.15}$$

where the maximum allowable covariance is defined as σ_f^2. We also assume a Gaussian noise model:

$$y = f(x) + \mathcal{N}(0, \sigma_n^2). \tag{2.16}$$

And we can fold the noise into $k(x, x')$,

$$k(x, x') = \sigma_f^2 exp[\frac{-(x - x')^2}{2l^2}] + \sigma_n^2 \delta(x, x'), \tag{2.17}$$

where $\delta(x, x')$ is the Kronecker delta function.

According to equation 2.5, we need to calculate several covariance matrices: a covariance matrix between the training samples:

$$K = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \dots & k(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \dots & k(x_n, k_n)) \end{bmatrix}, \tag{2.18}$$

a covariance matrix between the training and testing samples:

$$K_* = [k(x_*, x_1) \quad k(x_*, x_2) \quad \dots \quad k(x_*, x_n)] \tag{2.19}$$

and a covariance matrix between the testing samples:

$$K_{**} = k(x_*, x_*) \tag{2.20}$$

With the three matrices, the mean and the variance are predicted accordingly:

$$\overline{y}_* = K_* K^{-1}\mathbf{y}, var(y_*) = K_{**} - K_* K^{-1} K_*^T. \tag{2.21}$$

According to the specific prediction problem shown in figure 2.5, we can carry out the following steps to get the prediction from a GPR:

1. collect observations $\mathbf{y}$ at

$$\mathbf{x} = [-1.50 - 1.00 - 0.75 - 0.40 - 0.250.00].$$

2. define noise covariance $\sigma_n = 0.3$
3. define hyperparameters, $l = 1$ and $\sigma_f = 1.27$.
4. calculate covariance matrices:

$$K = \begin{bmatrix} 1.70 & 1.42 & 1.21 & 0.87 & 0.72 & 0.51 \\ 1.42 & 1.70 & 1.56 & 1.34 & 1.21 & 0.97 \\ 1.21 & 1.56 & 1.70 & 1.51 & 1.42 & 1.21 \\ 0.87 & 1.34 & 1.51 & 1.70 & 1.59 & 1.48 \\ 0.72 & 1.21 & 1.42 & 1.59 & 1.70 & 1.56 \\ 0.51 & 0.97 & 1.21 & 1.48 & 1.56 & 1.70 \end{bmatrix},$$

$K_{**} = 1.70$ and $K_* = [\ 0.31\ 0.68\ 0.92\ 1.25\ 1.38\ 1.54\ ]$.

5. predict the mean and the variance: $y_* = 0.95$ and $var(y_*) = 0.21$.

2.5 GPR for pose estimation

Suppose we have input data: training image sequences (X, represented with feature descriptors of dimensions m) with synchronized 3D motion data sequences (Y, represented with direction cosines of dimension n) and testing image sequences ($Xstar$, represented with feature descriptors) and for pose estimation we want our output to be: reconstructed latent poses ($Ystar$) from testing image sequences. The algorithm of GPR applied for pose estimation is shown as below:

In our implementation, we choose the most frequently used covariance function: squared exponential, which is proved to be effective in most of the dealt problems. We can see from the algorithm that GPR is an elegant framework for non-linear mapping problem. It takes the normalized input and output data, and by training hyperparameters, fit the mapping model, and with test input, it can predict output with the optimized model. Note, that we do need to know the semantic meaning of input data for normalization, as shown in following applications of action recognition, different normalization ways results in different action recognition accuracies.

Algorithm 1 Gaussian process regression for pose estimation

Input:X (training inputs), X' (validation input), Y (training targets), Y' (validation targets), X^* (test inputs)

for i = 1:k **do**
1) Normalize input data X, Y to get normalized input X_{new}, Y_{new} and offset off_Y,
2) Train GPR with $< X_{new}, Y_{new} >$ to optimize hyperparameters,
3) Predict mean $\overline{Y} = K_* K^{-1} Y_{new}$ and variance $\mathbb{V}[Y^*] = K_{**} - K_* K^{-1} K_*^T$ from X' with the optimized hyperparamters,
3) Get the prediction: $\overline{Y^*} = \overline{Y} + off_Y$
end for
return $\overline{Y^*}$ (mean), $\mathbb{V}[Y^*]$ (variance)

2.6 Multi-variate Gaussian

The definition of multi-variate Gaussian is as following: Let X be an n-dim random vector with mean vector μ and covariance matrix Σ. Let Y be a random variable defined as a linear polynomial:

$$Y = bX + a. \tag{2.22}$$

A random vector has a joint-normal distribution (is a multi-variate Gaussian) if every non-trivial liner polynomial of the random vector is itself normal. If we compare multi-variate Gaussian and Gaussian process regression, we can see that multi-variate Gaussian is a special case of GPR, where covariance function can be considered a linear function. Note that although the number of input variables are fixed, as readers might argue that in GPR, X is infinite, in real problem, we are mostly interested in GPR solving problems with infinite number of input variables. In an intuitive way, we can interpret GPR as covariance functions (that is, kernels) plus multi-variate Gaussian.

Chapter 3

Pose Estimation

In this chapter, we first compare two standard ways of estimate 3D human poses from 2D body part positions and validate them on standard public data set; then, we add a module of 2D body part detector from a state-of-art method and propose a framework to estimate 3D human poses from 2D images with cluttered background.

Full body 3D human pose estimation from monocular images is a difficult problem since the depth information is lost when projecting from 3D space to 2D image plane. For this, a huge set of approaches have been suggested to recover the 3D pose based on monocular images. One class of approaches tries to map image features directly to 3D poses. Another class of approaches first tries to map image features to 2D poses and then maps 2D pose estimates to 3D poses. For the later class there exist two subclasses that differ in the way in which 2D poses are lifted to 3D poses. Learning approaches try to learn this mapping using training examples and adapt some mapping mechanism. Modeling approaches try to model this mapping from 2D to 3D poses explicitly by using knowledge about the inverse of the 3D to 2D mapping. Although the learning and modeling approaches are quite different by concept for the 2D to 3D lifting task it has not yet been investigated systematically how the two classes of approaches differ and what are the advantages of each class. In the following sections, we are going to explain in details about these two schools of methods and compare their performances under different conditions.

3.1 Geometric Reconstruction of 3D Poses

A typical example of a modeling approach is the work presented by Taylor [71]. Assuming that the 3D to 2D image formation process can be modeled by a scaled orthographic projection, a 3D object point (x, y, z) is mapped to its corresponding 2D image point (u, v) by $u = s{\cdot}x, v = s{\cdot}y$. This corresponds to a parallel projection with a subsequent scaling with scaling factor s. If we know the 3D length of a limb l between two body marker points (x_1, y_1, z_1) and (x_2, y_2, z_2) and their corresponding projected points (u_1, v_1) and (u_2, v_2) we can reconstruct the displacement $\Delta_z := z_1 - z_2$ of the limb in z direction based on the measured length of the foreshortened limb in the 2D

image. Since:

$$u_1 - u_2 = s \cdot (x_1 - x_2), \quad v_1 - v_2 = s \cdot (y_1 - y_2) \tag{3.1}$$

we can reformulate the Euclidian equation to get:

$$\begin{aligned} l^2 &= (x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2 && (3.2)\\ \Leftrightarrow (z1 - z2) &= \pm\sqrt{l^2 - (x_1 - x_2)^2 - (y_1 - y_2)^2} && (3.3)\\ \Leftrightarrow \Delta_z &= \pm\sqrt{l^2 - \frac{(u_1 - u_2)^2 + (v_1 - v_2)^2}{s^2}} && (3.4) \end{aligned}$$

Note that the displacement Δ_z can be reconstructed for one limb only up to a sign (+ or −) ambiguity in equation 3.4 since we cannot decide which of the limb endpoints (x_1, y_1, z_1), (x_2, y_2, z_2) is nearer to the camera. Having two reconstruction possibilities for one limb, for a body model with N limbs we have 2^N reconstruction possibilities for the whole body pose.

To solve this ambiguity, Taylor's original work assumed that a user labels which endpoint of each limb is nearer to the camera. Thus the original method was only a semi-automatic 3D pose reconstruction approach. To choose one of this 2^N solutions automatically different approaches have been suggested. Jiang [34] compares each of the candidate poses with over 4 million pose examples from the CMU motion capture database to assess the probability of each candidate pose. Mori and Malik [48] compare an unknown image with a sample database of images using shape context descriptor matching. The sample images are labeled with 2D body part locations and the information for each limb which of its endpoints is closer to the camera. Based on the shape context descriptor matches the 2D body part location and the information which limb endpoint is nearer to the camera is transferred to the unknown image. Wei and Chai [84] also tackle the problem of how to determine the unknown scale factor s. The set of limb projection constraints for all limbs of a body model in equation 3.4 are augmented by further constraints based on limb symmetries and fixed lengths on some rigid subparts of the human body such as the pelvis. Nevertheless, sometimes these additional constraints are not sufficient to solve the ambiguity. In such cases the pose reconstruction stops and the user has to solve the ambiguity manually.

Beside this ambiguity, Taylor's original method is more applicable in conditions that cameras are placed far from the captured objects. The model assumes that the projected size of a person or a limb does not depend on its distance to the camera which is not true when cameras are near. Note that the z coordinate has no influence on the resulting (u, v) coordinate. Parameswaran and Chellappa [52] therefore try to deal with a new camera model, *i.e.* perspective projections. Possible head orientations are reconstructed using a set of polynomial equations, epipolar geometry is recovered, and the rest of the body joint coordinates are computed using knowledge about the limb lengths in a recursive manner. But in their approach the authors have to make two strong assumptions: the torso twist has to be small – which is not true for many poses – and the locations of four markers on the head have to be given (e.g. forehead, chin, nose and left or right ear) – which is hard to be provided automatically since it would mean a very precise automatic localization of these markers on the head.

An approach that does not need to make such assumptions and nevertheless adopts a perspective projection camera model was presented recently [18]. In the perspective

camera model a 3D point (x, y, z) is mapped to the 2D point (u, v) with $u = f\frac{x}{z} + c_0$, $v = f\frac{y}{z} + c_1$. f is called focal length, (c_0, c_1) is called principal point. For a known calibrated camera, we know the principal point (and the focal length), and can correct the 2D image coordinates for the principal point translation vector ($u' = u - c_0$, $v' = v - c_1$) and therefore assume $(c_0, c_1) = (0, 0)$. Since

$$x_i = \frac{z_i u_i}{f}, \quad y_i = \frac{z_i v_i}{f} \tag{3.5}$$

we can reformulate the Euclidian equation into a quadratic equation for the z_i coordinate as following (refer to [18] for deduction details):

$$\begin{aligned} l_{ij}^2 &= (\frac{z_i u_i}{f} - \frac{z_j u_j}{f})^2 + (\frac{z_i v_i}{f} - \frac{z_j v_j}{f})^2 + (z_i - z_j)^2 && (3.6) \\ &\dots \\ \Leftrightarrow z_{i_{1/2}} &= -\frac{C z_j}{2A} \pm \sqrt{(\frac{C z_j}{2A})^2 - (\frac{B}{A} z_j^2 - \frac{f^2 l_{ij}^2}{A})} && (3.7) \end{aligned}$$

Equation 3.7 shows how to reconstruct the z coordinate of a child marker z_i given already reconstructed z_j coordinate of a parent marker. We further need to provide limb lengths l_{ij} (connecting marker i with j), and the 2D coordinates (u, v) of the markers within the image which are supposed to be provided by a 2D pose estimator.

Assuming a perspective projection camera model, we first have to start with an estimate for the z coordinate of the root marker of the kinematic tree, then we can apply equation 3.7 in a recursive manner: having computed the z coordinate for a parent marker, we can compute the two possible solutions for the z coordinate of the child marker and step down further in the kinematic tree. Since there are still two solutions for the z coordinate (either + and − in equation 3.7) we end up with a binary reconstruction tree with 2^N mathematically possible poses. To reduce the number of pose candidates already during the binary reconstruction tree traversal it was shown in [18] that it is possible to check for abnormal joint angles based on anatomical joint limits in the knees and elbows and prune branches of the reconstruction tree whenever we encounter anatomical violations.

To select a final 3D pose estimate from the remaining set of pose candidates, we can assign a probability

$$P(\vec{p}) = \prod P(\vec{j_i}) \tag{3.8}$$

for each pose candidate $\vec{p}$, where $P(\vec{j_i} = (\alpha, \beta, \gamma))$ is the probability to find a joint in a certain configuration $\vec{j_i} = (\alpha, \beta, \gamma)$ (the three Euler angles) which can be learned by observing motion capture sample data. The z coordinate of the root marker can be estimated by the distance of the person to the image plane. In [18] the proposed solution for the estimation of the person to camera distance was reconstructing all possible poses using different distance estimates and then choose the distance where the average pose probability takes on a maximum. This approach is successful for estimating the person $\leftrightarrow$ camera distance since for distances different from the ground truth distance, the reconstructed poses have to be squeezed (distance too small) or pulled apart (distance estimate too big) into the perspectives rays bundle which in turn results in unlikely joint angles and small pose probabilities. For further details we refer the reader to [18].

3.2 Regression of 3D poses

The Gaussian process regressor is currently the most widespread representative for learning approaches in the pose estimation community since it has been proved to be an effective approach for the nonlinear 2D to 3D pose mapping problem [82]. The main idea of Gaussian process regression is to map unknown test data to a prediction by interpolating the training data weighted by the correlation between the training and test data. In our method, we take normalized 2D body part positions as input and output a 3D pose prediction – represented as a vector of direction cosines of limb orientations. In the following subsections, we will explain detailed representations and settings for the Gaussian process regressor used here.

3.2.1 Normalized 2D Body Part Positions

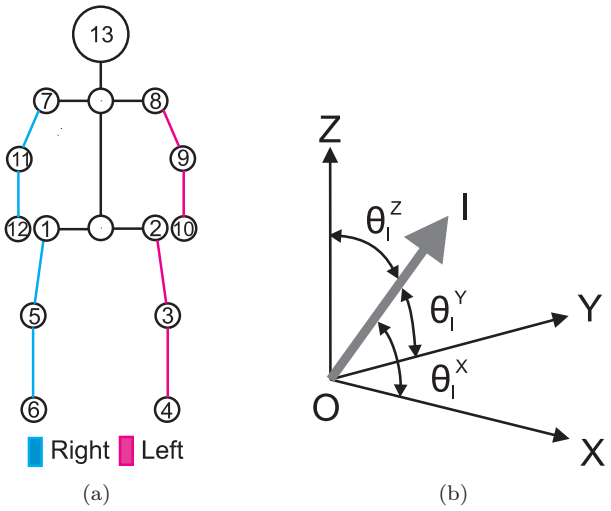


Figure 3.1: (a) The 3D stick figure model used for representing human pose with 2D body part indices. Thirteen body parts corresponding to the markers used in motion capture are used [65]. (b) The angles $(\theta_l^x, \theta_l^y, \theta_l^z)$ between the limb l and the axes [59].

From detected body part positions of a performer, we take 13 body parts. Correspondence of body parts and 3D stick figure model is shown in figure 5.3(a). The 2D body part positions are collected within a vector $BP = [x_1, y_1, x_2, y_2, \ldots, x_i, y_i, \ldots, x_{12}, y_{12}, x_{13}, y_{13}]$ where (x_i, y_i) is the 2D position of the i-th body part. For representing the 2D pose independently of the persons's size and distance to the camera, we normalize this 2D pose vector:

$$BP_{norm} = (BP + M_{off}) * M_{scale} \tag{3.9}$$

where $*$ means element-wise multiplication and

$$M_{scale} = [\frac{1}{y_{range}}, \frac{1}{y_{range}}, \ldots, \frac{1}{y_{range}}, \frac{1}{y_{range}}] \tag{3.10}$$

$$M_{off} = [x_{off}, y_{off}, x_{off}, y_{off}, \ldots, x_{off}, y_{off}] \tag{3.11}$$

$$x_{off} = -min(X) + (y_{range} - x_{range})/2 \tag{3.12}$$

$$y_{off} = -min(Y) \tag{3.13}$$

where X and Y are vectors of all x and y coordinate values of the 2D pose in the frame.

For upright standing persons, the range of y coordinate values is typically bigger compared to the range of x coordinate values. For this, we normalize both x and y coordinates by y range in each frame (M_{scale}). This makes sure, that we keep the aspect ratio of the performer and that normalized y coordinates range from 0 to 1. The normalized 2D part positions are the input of the regressor.

3.2.2 3D Human Pose Representation

The output of our regressor is 3D human poses. According to our experiments, with the dimension of the output from regressor increases, the regressor will take more time for training parameters. We use the same representations for human posture as in [59], considering this representation is concise and unambiguous. We model a human pose using twelve rigid body parts: hip, torso, shoulder, neck, two thighs, two lower legs, two upper arms and two forearms. These parts are connected by a total of ten inner joints, as shown in figure 5.3(a). For defining a local coordinate system in the hip, we use the direction of the torso for the y axis and the direction vector pointing from the left hip to the right hip as z axis. The x axis is then given by cross product of y axis and z axis.

The pose of an actor in an image frame is represented as a vector of direction cosines, i.e. the cosines of the angles between the limb direction vectors and the three coordinate axes of the root coordinate system. That is, limb orientation is modeled using three parameters, without modeling self rotation of limbs around its axes, as shown in figure 5.3(b). The overall posture of the subject for a frame is represented using a vector of direction cosines measured on twelve limbs. This results in a 36-dimensional representation of the pose:

$$\psi = [\cos\theta_1^x, \cos\theta_1^y, \cos\theta_1^z, \ldots, \cos\theta_{12}^x, \cos\theta_{12}^y, \cos\theta_{12}^z], \tag{3.14}$$

where θ_l^x, θ_l^y and θ_l^z are the angles between the limb l and the axes of the root coordinate system in the hip as shown in figure 5.3(b).

3.2.3 Gaussian Process Regression

For regression model, we choose Gaussian process regression (GPR) model due to its successful application in non-linear regression and prediction. Please refer to chapter 2.6 for its definition, detailed explanations and an example of applying GPR in solving prediction problem.

As we mentioned in chapter 2.6, a Gaussian process is completely specified by a mean function and covariance function. If we denote the mean function as $m(\mathbf{x})$ and the covariance function as $Cov[\mathbf{f}(\mathbf{x_1}), \mathbf{f}(\mathbf{x_2})] = k(\mathbf{x_1}, \mathbf{x_2})$, a Gaussian process is denoted as $\mathbf{f}(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x_1}, \mathbf{x_2}))$, where

$$m(\mathbf{x}) = E[\mathbf{f}(\mathbf{x})] \tag{3.15}$$

$$k(\mathbf{x_1}, \mathbf{x_2}) = E[(\mathbf{f}(\mathbf{x_1}) - m(\mathbf{x_1}))(\mathbf{f}(\mathbf{x_2}) - m(\mathbf{x_2}))] \tag{3.16}$$

The covariance function specifies how two function values $f(x_1)$ and $f(x_2)$ (the function values are considered as random variables) can change, given two arguments x_1,x_2. Since we want the Gaussian process to interpolate continuously between supporting points, a continuous covariance function is used as well. A typical covariance function that is used for a Gaussian process is the squared exponential:

$$k(\mathbf{x_1}, \mathbf{x_2}) = \theta_1^2 \exp(-\frac{(\mathbf{x_1} - \mathbf{x_2})^2}{2\theta_2^2}) \tag{3.17}$$

where θ_1, θ_2 are called the amplitude and lengthscale hyperparameters respectively. This covariance function makes sure that the covariance of two function values $f(x_1)$ and $f(x_2)$ of nearby x_1, x_2 is high (which will result in a smooth function), while the covariance of $f(x_1)$ and $f(x_2)$ is low, if x_1, x_2 are far away.

Given a 2D pose estimate which is represented as the 26 dimensional vector BP_{norm} we train one Gaussian process to predict each of the 36 dimensions of the 3D pose vector ψ separately. For the Gaussian process training and prediction we used a reference implementation[1].

3.3 Performance Comparisons

In this section, we describe the settings for the experiments and how we measure the error of an estimated pose both for the regression and the geometric reconstruction method. Based on a comparison of these errors, we analyze and conclude advantages and disadvantages of each method.

3.3.1 Training and Test Data Composition

We chose the public available HumanEva [65] and the TUM kitchen data set [72] for an exhaustive evaluation and comparison of both methods since both data set provide 3D motion capture ground truth data which allows to compute an error for each estimated pose. Furthermore, intrinsic and extrinsic camera parameters are provided as well which allows to project the 3D poses into the image and thereby provide 2D ground truth poses as well. Both data set contain sequences where different subjects (4 for HumanEva, 4 for TUM kitchen) perform different actions (walking, boxing, laying a kitchen table, etc.) recorded from different viewpoints (7 for HumanEva, 4 for TUM kitchen). Table 3.1 shows detailed configuration for all experiment settings. These variations of performers, action types and viewpoints between training and

[1] http://www.gaussianprocess.org/gpml/code/matlab/doc/

Exp.	training	testing	change of
1a	TUM, 0-0-cam3, S1	TUM-0-0-cam2, S1	viewpoint (weak)
1b	HE, walk-cam1, S1	HE, walk-cam2, S1	viewpoint (weak)
1c	TUM, 0-0-cam1, S1	TUM-0-2-cam3, S1	viewpoint (strong)
1d	HE, box-cam1, S1	HE, box-cam2, S1	viewpoint (strong)
2a	TUM, 0-0-cam3, S1	TUM-0-3-cam3, S2	person
2b	HE, walk-cam1, S1	HE, walk-cam1, S2	person
3a	HE, walk-cam1, S2	HE, box-cam1, S2	action
3b	HE, box-cam1, S2	HE, walk-cam1, S2	action
4a	HE, walk-cam2, S1	TUM, 0-2-cam3, S2	data set
4b	TUM, 0-2-cam3, S2	HE, walk-cam2, S1	data set

Table 3.1: Experiments definition for both the geometric reconstruction and the regression approach.

Exp.	training	testing
5a	TUM, 0-0-cam3, S1	TUM-0-0-cam2
5b	HE, walk-cam1, S1	HE, walk-cam2, S1

Table 3.2: Experiment settings of the geometric reconstruction method with noisy 2D input poses and different noise levels.

Exp.	training	testing
5a	TUM, 0-0-cam3, S1	TUM-0-0-cam2
5b	HE, walk-cam1, S1	HE, walk-cam2, S1

Table 3.3: Experiment settings of the regression method with noisy 2D input poses and different noise levels.

test data, allow to define a set of experiments in which different capabilities of both methods can be tested.
We use 4 categories of experiments:

1. *train* on a sequence recorded from one camera view $\rightarrow$ *test* on a sequence recorded from another view (1a/1b/1c/1d). The change of viewpoint can be weak (1a/1b) or strong (1c/1d).

2. *train* on a sequence comprising a subject $S_i \rightarrow$ *test* on a sequence comprising another subject S_j [2] (2a/2b),

3. *train* on a sequence showing one action class $A_1 \rightarrow$ *test* on a sequence showing another action class A_2 (3a/3b),

[2]Person S_i within the TUM kitchen data set is different from the person S_i within the HumanEva data set

4. *train* on a sequence from HumanEva (TUM kitchen) data set → *test* on a sequence from the other data set, i.e. TUM kitchen (HumanEva) (4a/4b)

Both approaches, the regression and the geometric reconstruction method, map 2D input poses to 3D pose estimates. In experiments 1a-4b we test on ground truth 2D input poses and estimated 2D poses as well (see table 3.4). The estimated 2D poses stem from a Implicit Shape Model based 2D pose estimator, that learns the spatial relation between SIFT features and 15 body parts and uses this learned relationship to vote for the location of each body part. The method and the quality of these estimated 2D poses is described in detail in [49].

The quality of estimated 2D input poses depends on the performance of the 2D pose estimator. To be independent from this quality of a 2D pose estimator we also added two further experiments 5a/5b (table 3.2 and table 3.3) in which we put more and more noise onto ground truth 2D poses and evaluate the capability of both the regression and geometric reconstruction method to estimated 3D poses with such 2D poses of different noise levels. Here, a noise level of n% means that we added a random translation vector (Δ_x, Δ_y) to each marker position where the length of this random vector is in the range of 0-n% of the person's height (measured in pixels) in the current frame.

In the learning phase the regression method uses the (2D ground truth pose, 3D ground truth pose) pairs of the training data to train the Gaussian processes and fix the hyper-parameters. In contrast, the geometric reconstruction method uses only the 3D ground truth poses of the training data to learn joint angle probabilities for all joints.

3.3.2 Error Measurements

Since we use the same input 2D poses (ground truth / estimated / noisy) for both experiments this allows us to compare both approaches - the regression and the geometric reconstruction - based on their 3D pose estimation performance which is measured by the average angular error and average absolute marker position error of the estimated 3D poses compared to the ground truth 3D poses. Suppose predicted limb angles $\hat{\Theta}$ and ground truth limb angles Θ are denoted as

$$\hat{\Theta} = [\hat{\theta}^x_{l_1}, \hat{\theta}^y_{l_1}, \hat{\theta}^z_{l_1}, \ldots, \hat{\theta}^x_{l_{14}}, \hat{\theta}^y_{l_{14}}, \hat{\theta}^z_{l_{14}}] \tag{3.18}$$

$$\Theta = [\theta^x_{l_1}, \theta^y_{l_1}, \theta^z_{l_1}, \ldots, \theta^x_{l_{14}}, \theta^y_{l_{14}}, \theta^z_{l_{14}}] \tag{3.19}$$

then the angular error is defined as:

$$Err_{Ang} = \frac{\sum_{i=1}^{J} |\Theta_i - \hat{\Theta}_i| \bmod 180^\circ}{J}, \tag{3.20}$$

where $J = 3 \cdot 14$ (3 Euler angles, 14 limbs) and "mod" is to deal with angle singularity problem.

An angular error in a joint at a high level of the kinematic tree (e.g. shoulder joint) will have a bigger impact on the resulting pose than an angular error in a

joint at a low level of the kinematic tree (e.g. wrist joint). For this, we do not only compute the angular error Err_{Ang} but also compare the absolute marker positions of the estimated poses with the ground truth pose.

Since the geometric reconstruction method first reconstructs 3D marker locations and then computes joint angles based on the reconstructed marker positions this comparison is straightforward. In contrast, the regression method maps a normalized 2D body part location vector BP_{norm} to a 3D joint angle vector ψ such that there are at first no estimated 3D marker locations at all. To compute 3D marker location estimates, we assume a person of average U.S. size [43] and use pre-computed relative limb length ratios to compute absolute limb length estimates. Based on the estimated joint angles and these estimated limb lengths we can then reconstruct 3D marker locations as well. If we denote these estimated marker positions $\hat{\mathbf{P}}$ and ground marker positions $\mathbf{P}$:

$$\hat{\mathbf{P}} = [\hat{x}_1, \hat{y}_1, \hat{z}_1, \ldots, \hat{x}_{15}, \hat{y}_{15}, \hat{z}_{15}] \tag{3.21}$$

$$\mathbf{P} = [x_1, y_1, z_1, \ldots, x_{15}, y_{15}, z_{15}], \tag{3.22}$$

then the average marker position error is defined as

$$Err_{pos} = \frac{\sum_{i=1}^{M} |\mathbf{P}_i - \hat{\mathbf{P}}_i|}{M}, \tag{3.23}$$

where $M = 3 \cdot 15$ (x/y/z coordinates, 15 markers). Err_{Ang} is specified in degrees, Err_{pos} in mm.

3.3.3 Results

The 3D pose estimation error results of all experiments are shown in table 3.4, table 3.5, and table 3.7. For the case of estimated 2D input poses (see table 3.4) and noisy 2D input poses (see table 3.5 and table 3.7) the results are obvious: the regression method outperforms the geometric reconstruction method in all scenarios. This shows that the Gaussian process learning based approach is able to use the training data samples sufficiently to interpolate to new data. The estimated 3D poses generated from the regression method are substantially better than for the 3D pose estimates obtained from the geometric reconstruction method. This shows that the geometric reconstruction approach presented in its puristic form here is not able to deal with noisy 2D input poses. As the 2D input poses get more and more noisy, errors from regression method increase slower compared to errors of the geometric reconstruction method (compare table 3.5 with table 3.7). The reason is that the wrong 2D body part locations will lead to wrong 2D limb lengths which in turn will lead to wrong displacement values (see equation 3.4 and equation 3.7). The working principle – using the foreshortening information of limbs to reconstruct the limb displacement in z direction – continuously loses its basis with increasing noise in the 2D input poses (see table 3.5). This underlines the need to augment modeling based approaches for 2D to 3D pose estimation with some explicit handling of noise while it is not necessary for the regression / learning based approaches.

Exp.	**Geometric reconstruction**				**Regression**			
2D input	Ground truth		Estimation		Ground truth		Estimation	
	[°]	[mm]	[°]	[mm]	[°]	[mm]	[°]	[mm]
1a	6.12	143.51	13.44	230.72	**0.12**	**1.82**	**7.358**	**146.83**
1b	7.47	155.77	10.74	187.57	**1.39**	**22.27**	**3.79**	**78.80**
1c	5.24	135.14	10.93	194.49	**5.48**	**96.69**	**5.85**	**102.71**
1d	8.15	159.14	11.59	189.33	**4.49**	**85.02**	**5.05**	**95.56**
2a	6.53	156.43	12.20	197.92	**5.52**	**89.09**	**7.92**	**140.93**
2b	8.61	158.41	11.71	194.97	**3.87**	**64.29**	**5.12**	**95.18**
3a	16.65	210.78	17.18	202.41	**11.48**	**197.23**	**11.53**	**192.57**
3b	**9.10**	**153.64**	12.02	197.35	9.34	166.02	**9.13**	**160.09**
4a	**7.57**	155.15	13.48	214.47	8.40	**137.11**	**8.47**	**139.70**
4b	8.07	160.06	10.98	188.34	**7.08**	**123.78**	**7.52**	**131.10**

Table 3.4: 3D pose reconstructions errors for both the geometric reconstruction and the regression approach. For each experiment we present the average angular and average marker position error of the estimated poses resulting from the geometric reconstruction and the regression approach compared to the ground truth poses.

Exp.	**Geometric reconstruction**									
	2%		4%		6%		8%		10%	
	[°]	[mm]	[°]	[mm]	[°]	[mm]	[°]	[mm]	[°]	[mm]
5a	6.69	149.19	7.83	160.43	9.14	175.17	10.51	191.87	11.75	205.82
5b	8.00	159.03	8.97	168.07	10.04	177.59	11.42	189.02	12.53	198.49

Table 3.5: 3D pose reconstruction errors for the geometric reconstruction method with noisy 2D input poses and different noise levels.

Exp.	**Regression**									
	2%		4%		6%		8%		10%	
	[°]	[mm]	[°]	[mm]	[°]	[mm]	[°]	[mm]	[°]	[mm]
5a	5.72	112.01	5.77	113.11	5.85	115.01	5.94	117.08	6.03	119.21
5b	1.55	24.70	1.90	31.34	2.27	39.53	2.59	47.53	2.85	54.73

Table 3.6: 3D pose reconstruction errors for regression method with noisy 2D input poses and different noise levels.

For the case of using 2D ground truth input poses, the average 3D pose error is for the regression method 5.7° (averaged over all experiments 1a-4b) compared to 8.3° for the geometric reconstruction approach. This shows that the regression method yields 3D pose estimates with an error of about 2.6° lower than the modeling approach used here. Especially due to the fact that the Gaussian process regressor

yields better 3D pose estimates in average, it is interesting that nevertheless in some cases, the geometric reconstruction method could outperform the regression method slightly (1c/3b/4a). We trace this back to the fact, that learning based approaches run into problems if the test data is substantially different from the training data. This is the case in experiment 1c where we find a big viewpoint change, in 3b where we find a change of action, and in experiment 4a where we switched from one data set to another. We expect an even bigger difference if we compute the joint angle probabilities on a bigger variety of motion capture data compared to the situation here in which we use just one data set to estimate the joint angle probabilities. In scenarios, in which there are new actions, viewpoints or other changes compared to the training data we expect the model based approaches to be the better choice since then interpolation capabilities of learning based approaches will not be sufficient to generalize to the new data.

When there is a big variance regarding the action type, the regression method has problems predicting the 3D poses correctly, because no similar poses are learned in training. If variances are only present for certain limbs, for example upper body limbs or lower body limbs, the regression method can correctly predict the body parts that have similar orientation as in training data, e.g. experiment 4b with estimated 2D body part input in figure 3.2. This is due to the fact that every limb orientation in the regression method is estimated separately from others, in contrast to the modeling approach. Thus, for the regression method errors from the root of the kinematic tree structure will not transmit to leave nodes. Another problem occurs when there are ambiguities in mapping, e.g. in experiment 1c with ground truth 2D body part input in figure 3.2, the predict pose is left-right flipped compared with ground truth 3D pose. This is due to lack of depth information in 2D images, the same 2D pose might correspond to more than one 3D poses.

Although the modeling method tends to be a more flexible modeling method independent of training samples, we choose a learning method with Gaussian process regression model as our model of learning 2D pose/3D pose mapping, or later 2D image feature/3D pose mapping. In the following sections, we propose a framework of estimating 3D poses from still images with cluttered backgrounds by introducing a state-of-art 2D body part detector.

3.4 Detector of 2D Poses

The dominant approach towards 2D human pose estimation implies articulated models in which parts are parameterized by pixel location and orientation. The approach used in [90] introduces a model based on a mixture of non-oriented pictorial structures. The main advantages of using the articulated mixture model consist in the fact that it is highly customizable, using a variable number of body parts, and that it reflects a large variability of poses and appearances without requiring background or temporal information. Also, it outperforms state-of-the-art 2D detectors while requiring less processing time. The next sections describe the model proposed in [90]:

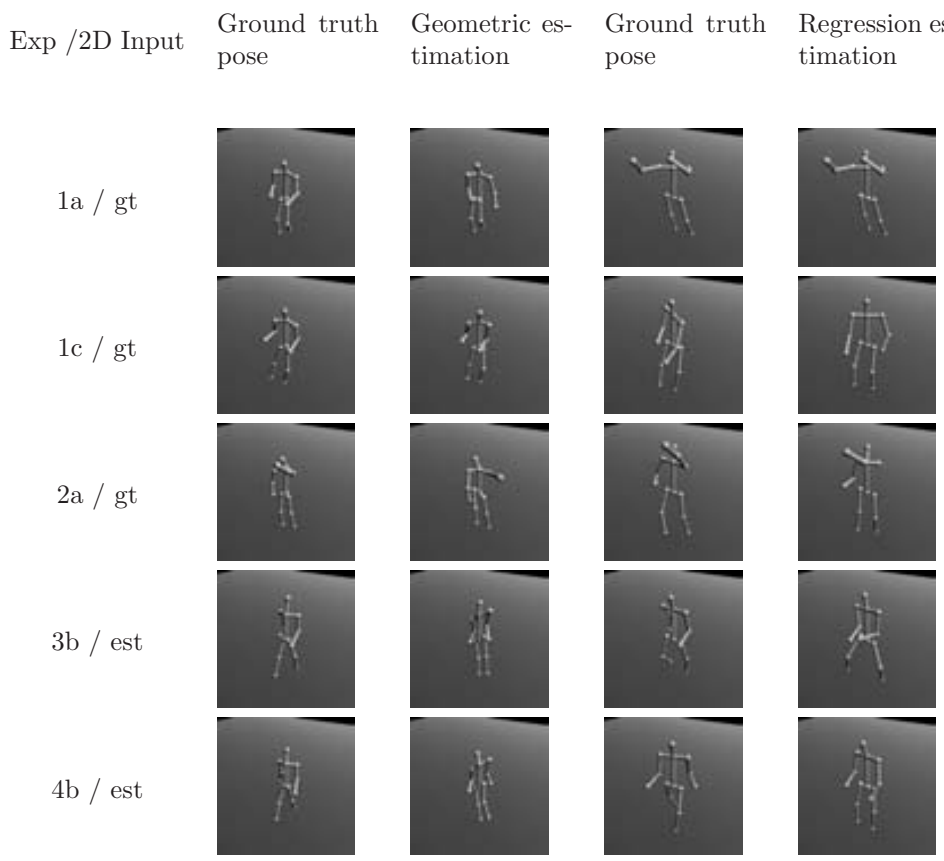


Figure 3.2: Qualitative 3D pose estimation samples. Column "Exp /2D Input" shows the experiment number and 2D pose input type. There are two types of 2D input poses. "gt" means ground truth 2D pose and "est" means estimated 2D pose. 1st+2nd column: ground truth 3D pose and corresponding estimated 3D pose by the geometric reconstruction approach. 3rd+4th column: ground truth 3D pose and corresponding estimated 3D pose by the Gaussian process approach.

3.4.1 Part-based Model for Human Detection

The mixture model implies mixtures of parts or part types for each body part, in our case spanning different orientations and modeling the implied correlations. The body model can be associated with a graph in which nodes are represented by body parts and edges connect parts with strong relations. Similar to the star-structured part-based model in [66], this mixture model involves a set of filters that are applied to a HOG feature map [23] extracted from the analyzed image. A configuration of parts for a part-based model specifies which part type is used from each mixture and its relative location. The score of a configuration of parts is computed according to

three model components: co-occurrence, appearance and deformation [90]:

$$S(I,p,t) = \sum_{i \in V} b_i^{t_i} + \sum_{ij \in E} b_{ij}^{t_i,t_j} + \sum_{i \in V} \omega_i^{t_i} \cdot \phi(I,p_i) + \sum_{ij \in E} \omega_{ij}^{t_i,t_j} \cdot \psi(p_i - p_j), \quad (3.24)$$

where the first term favors certain part type t_i for body part i, the second term favors certain co-occurrence body part type t_i of body part i and body part type t_j of body part j, the third term expresses the local appearance score by assigning weight templates associated to part i and part-type t_i to certain locations p_i, described by the extracted HOG descriptor, and the fourth term expresses the deformation score by assessing the part-type pair assignment parameters and the relative location between connected parts i and j.

Figure 3.3: Person detected using a 26-part model, highlighting body part locations with circles. The upper row presents successful detections and the bottom presents wrong limb detections.

As the model described is highly customizable, experiments have been deployed as to find a more efficient model structure by varying the number of part-types and mixtures. A full-body 26-part model (figure 3.3) is chosen, as it shows increased performance due to the capture of additional orientation.

3.4.2 Inference and Learning

Inference using the mixture model described is obtained by retrieving the highest-scoring configuration, precisely by maximizing the score at root position $S(I,p,t)$

(equation 3.24) over all parts and part-types. Building the associated relational graph G as a tree allows message passing from children to the root of the tree and also tracing back inference of body part positions with dynamic programming. Still we should keep in mind that the possible shortage of the tree representation is the double counting problem, where one body part might be detected as two overlapped body part positions due to its strong response to both trained templates.

The essence of mixture of parts model is to cluster each body part into different set and learn appearance templates for each set. This allows for a variety of different appearances for each body part which is crucial for pose estimation. Also the limb orientation is represented as the connection between these body parts. This representation eliminate the possible confusion caused by explicitly learning limb orientations. The solution used for training a model which generates high scores and outputs a set of parameters containing limb locations is a SVM, leading to a problem of quadratic programming (QP), which in this case is solved using dual coordinate-descent.

With the state-of-art method introduced in this section, we extend the 2D pose/3D pose estimation module to 2D image feature inputs. So now the overall framework is able to take 2D image features as input and estimate 3D human poses from trained GPR model. In the following section, we show the experiments on public data set for validating the proposed framework.

3.5 Experiments

All experiments are carried on the HumanEva I data set as it provides ground-truth 2D and 3D information on subjects performing different actions. For every action, the image frames are equally divided in training and test data, the input received being vectors of 2D coordinates. 3D estimation performance is measured using the average angular error and average absolute marker position error defined in equation 3.20 and equation 3.23.

Experiments are conducted by varying the dimension of the input vectors containing the normalized 2D coordinates from the 2D detector. The final results are compared with an approach that uses a similar Gaussian process regressor and, as input, histograms of shape contexts obtained from extracted silhouettes [11]. As the silhouette-based experiments are carried in controlled conditions, requiring fixed cameras and background information, we will consider the method as ground truth experiment.

The dimensions of the input are varied by manually choosing significant body parts and obtaining the associated coordinates by re-projecting the 2D coordinates. Ground truth data is obtained in a similar manner according to the HumanEva marker positions. The results show that using a simpler body representation for regression input performs better while training and prediction are less time consuming. Therefore, a 16-dimensional input is chosen containing normalized 2D coordinates corresponding to body parts: head, neck, upper and lower torso, two shoulders, two elbows, two wrists, two hips, two knees and two ankles.

The shape context-based solution [11] outperforms the two-stage framework because of the increased reliability of the features extracted from silhouettes. The biggest

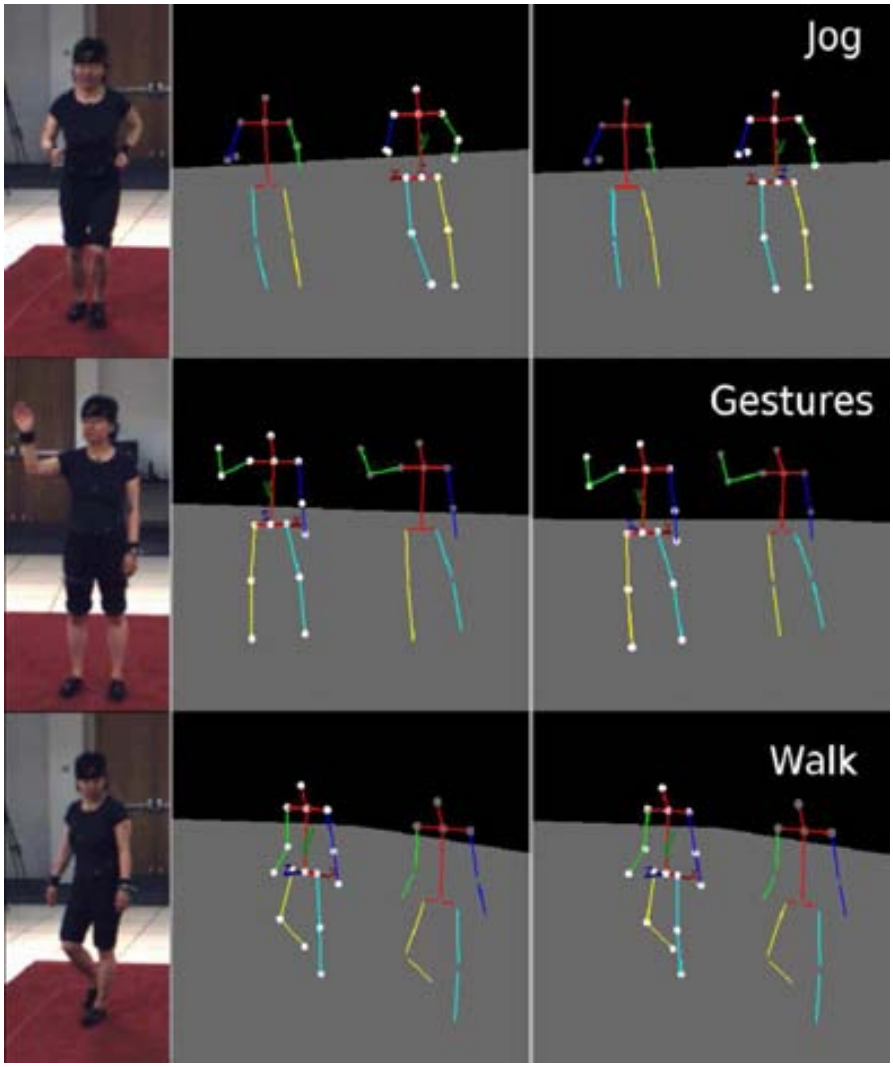


Figure 3.4: Visualized pose estimation results. The first column shows the original image inputs, the second column shows estimated poses using shape contexts and the third columns shows results of our approach. Estimated poses are highlighted with enlarged body joints, while the simple stick figure represents 3D ground truth data.

error rate is obtained for the "Jog" database, where a bigger number of frames present self-occlusions and generate double-counting and wrong limb detections. In the "Gestures" database the camera viewpoint is constant leading to a smaller error rate. Figure 3.4 presents visualizations of results for the HumanEva database.

Input	Motion	Err_{Ang}	Err_{Pos}
Our Method	Walking	1.85	41.50
	Box	2.68	45.45
	ThrowCatch	2.50	45.98
	Jog	2.64	49.93
	Gestures	0.89	12.07
GT	Walking	0.96	21.75
	Box	1.04	16.97
	ThrowCatch	1.08	19.19
	Jog	1.42	26.96
	Gestures	0.55	7.61

Table 3.7: Results obtained on the HumanEva data set. We compare the performance of two different inputs: detected body part positions and ground truth body part positions.

3.6 Conclusions and future work

This chapter compares two categories of methods for boosting from 2D body part detection to 3D poses: modeling and learning methods, then it presents a framework with learning approaches for the problem of 3D pose estimation from monocular images by exploiting the state-of-art 2D body part detector. Experiment results conducted on HumanEva dataset shows that learning method outperforms modeling method. And results from the state-of-art 2D body part detectors combined with the learning method is an effective way to estimate 3D poses.

For future work, it would be interesting to enhance the performance of the 2D detector within the temporal context, using a "tracklets" approach [7] for different frame window sizes [19]. Another alternative would be to incorporate multiple cues other than HOG in the original 2D body part detector. Within a general framework, we could optimize the overall gain computed from the each input feature cues. Under the proposition that multiple features provide more information than a single feature cue, we can utilize information from multiple cues. For example, we can multiply the probabilities from multiple cues to get a more robust detection.

Another problem the current state-of-art 2D body part detectors are facing is the mislabeling of detected 2D body parts, that is although body parts are detected correctly, which is considered a valid detection, it is not given the correct label, for example, a correctly localized right hand might be labeled as a left hand. A straightforward way to solve this problem is to impose physical constraints to the predicted 3D poses in the future, so that we can get back to 2D body part detections and enhance their precision.

Further exploitations could also be extended to input data with depth information. That is, instead of exploring more features and mapping models, we can resort to input with more information. For example, it would be interesting to see how state-of-art 2D body part detectors on color images works on range data. Or we can combine range data with image data to enrich the input information, but in that case, we

might need extra correspondence information between these two types of input data.

Chapter 4

Feature Robustness

However, the above features do not distinguish between noises (camouflage and shadows from background subtraction) and normal body parts. When input features from all images are passed to regressors, scalars from the same dimension have a unique weight. That is, noise from dimension d in frame f_1 is treated the same as normal feature from dimension d in frame f_2. To overcome this shortage, we introduce iterative closest point algorithm for point samples of noisy silhouettes. In the proposed measurement, noise from dimension d in frame f_1 could be discarded by filtering. The new distance measure is able to discard noisy parts like camouflage from the background and shadows. This is achieved by filtering points to be matched with an automatically adjusted threshold. With the proposed distance measue we divide the extracted silhouettes into different noise levels. Then we compare a robust input feature with the proposed distance measure for point samples. The comparisons are based on pose estimation accuracies in regression models. We further validate the performance of these two feature measurement on HumanEva dataset.

As mentioned before, there are several models like support vector machine (SVM), relevance vector regressor (RVR), Gaussian process regression (GPR) and their modified versions that can be utilized to learn mapping from 2D image features to 3D human poses. It is worth mentioning that Gaussian process models have been successfully applied to modeling non-linear problems [57, 32, 68, 82], predicting trends of stock market [12], head pose estimation problems [60], human pose estimation problems [94], tracking problems [76] and so on. Due to its successful applications in regression problems, we choose Gaussian process regression as our learning model. That is, the baseline method is trained with image feature descriptors and their corresponding 3D human poses. The trained GPR models are then tested with the test set.

The main contributions of this chapter are as following:

1. we compare 3D pose estimation accuracies from commonly used image features: shape context, SIFT with bag of words representation and PHOG;

2. we propose a new feature measure which compute distances between noisy silhouettes and non-noisy silhouettes;

3. we explore the effect of noisy inputs to human pose estimation accuracies;

4. we validate the proposed feature measure and PHOG (the most robust image feature from above) on a public dataset: HumanEva dataset.

In the rest of the chapter is organized as following: in section 4.1, we set up several commonly used image features for human pose estimation and compare their performances on training set of HumanEva and choose the most robust one for further analysis. Section 4.2 gives a detailed description about the proposed new feature measure which gives quantitive value of noisy conditions of extracted silhouettes. With this measure, we noisy silhouettes into two noise levels, and compares feature performance of different noisy inputs. In section 4.3, we validate our proposed feature measure and PHOG feature with the online evaluation from HumanEva dataset.

4.1 Image features for human pose estimation

From captured image sequences, we first use background subtraction [6] to get human silhouettes. Due to background camouflages, shadows and noisy edges, subtracted silhouettes vary with different levels of noises. For example, some of the silhouettes are uneven along the edge while others contains extra blobs from background camouflages or shadows. We first compare the overall performances of human pose estimation accuracies for all candidate image features, and select the one with the best performance for further robustness analysis. The most commonly used features for describing extracted human silhouettes include shape context [3, 11], scale invariant feature transform (SIFT) [11], histograms of gradient (HOG) [90]. Since pyramid of histogram of gradient (PHOG) incorporate position information into the descriptor in a more concise, it suits human pose estimation from extracted silhouettes better. So in our comparison, we consider shape context, SIFT and PHOG.

4.1.1 Shape Context

The shape context descriptor was proposed by S. Belongie and J. Malik [9]. It is first applied for shape matching and object recognition problems [8]. After sampling points from a shape, which is usually represented with its silhouette, the shape context descriptor describes the statical distribution of sampled points with respect to a point sample. To be specific, we place the origin of a local polar coordinate system on a sample point, define a grid with several angles and several radii, and count the number of sampled points in each bin. For human pose estimation problem, G. Mori and J. Malik [47] use the original set of shape context descriptors from all sampled points, while A. Agarwal and B. Triggs [3] propose to cluster all context descriptors and use a histogram representation over the cluster centers.

While shape context is a rich descriptor for extracted human silhouettes, users need to define radius parameter within which point samples are counted. And this parameter can vary according to user definition. For example, G. Mori and J. Malik [47] define the radius to include all sample points while A. Agarwal and B. Triggs [3] use a diameter of roughly a limb size. And there is no related work on what is the

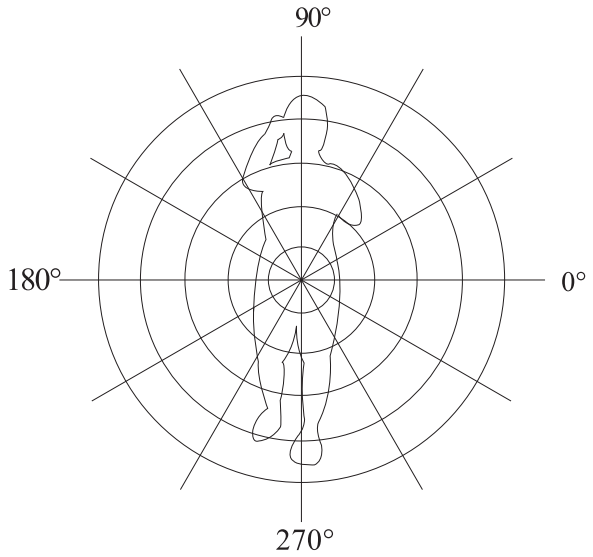


Figure 4.1: The Shape Context Descriptor in Our Method. The origin of the polar coordinate system is on the centroid of the extracted silhouette. The longest diameter is equal to the diagonal of the silhouette bounding box. The space is divided by five equally dividing radii.

best size of the radii. In our method, we place the origin of the local polar coordinate system at the centroid of the extracted silhouette with the longest diameter equal to the diagonal of the silhouette bounding box. We define five radii equally dividing the space and twelve angels. Origin placed at the centroid of the human silhouette, all sample points should be equally weighted. So we use polar coordinates instead of log-polar coordinates in the original shape context descriptor. The final dimensions of our shape context descriptor is $5 * 12$, as shown in figure 4.1.

4.1.2 PHOG

Histogram of gradient (HOG) was proposed by N. Dalal and B. Triggs for human detection [23] and is widely applied in pedestrian detection [70, 10] and deformable object detection [53]. The original HOG descriptor are computed with several steps:

1. gamma/colour normalization, which are reported to have modest effect;

2. gradient computation, where gradients are computed with a filter $[-1\ 0\ 1]$;

3. spatial/orientation binning, where each pixel votes for an orientation and votes are accumulated within a local region, called cells;

4. normalization and descriptor blocks, in which cells are grouped into blocked and normalized separately.

The successful applications of HOG descriptor in human detection problems rely on several factors. For example, overlaps in binning step make HOG a dense descriptor compared with SIFT and separate normalization within blocks allows local variations in illumination and foreground-background contrast. These advantages make it extremely useful for detecting deformable object from cluttered backgrounds.

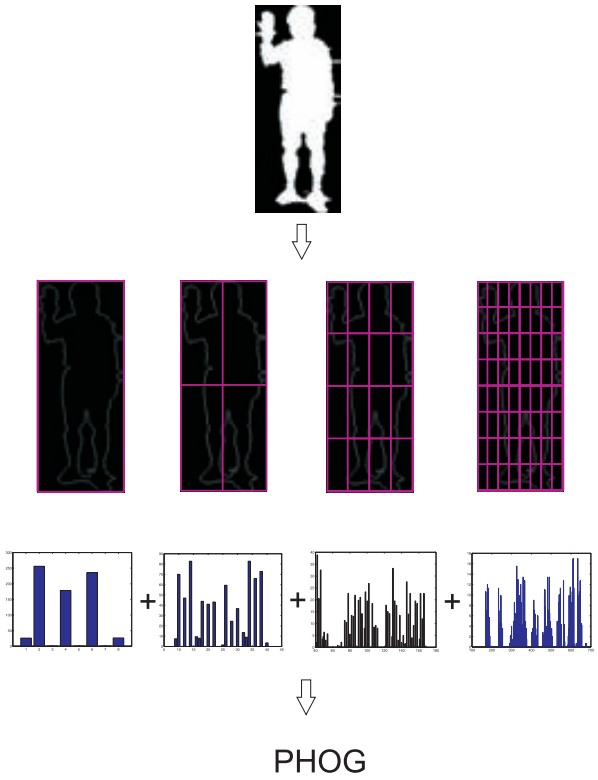


Figure 4.2: The pyramid of histogram of gradient descriptor in Our Method. We use four level of pyramids, eight orientation bins and 180 degrees of maximum orientation.

Although there is a related work which applies HOG for pose estimation [90], it is not terribly common that HOG is used for pose estimation. The reason might be the overhead of the descriptor. For example, in [23], the best size of the descriptor is $3 * 3 * 6 * 6 * 9$, where $3 * 3$ is the size of blocks, $6 * 6$ is the size of cells, and 9 is the number of orientation bins. In Gaussian process regression, a distance matrix should be computed among all training data, the length of the descriptor has an impact on the performance of the method. In our experiments, we choose pyramid of histogram of gradient (PHOG). PHOG descriptor was proposed by A. Bosch, A. Zisserman and X. Munoz [16]. It abandons both overlaps of cells in binning step and separate normalization within blocks in normalization step. But PHOG keeps the statistical representation of edge orientations and is more concise compared with HOG. To describe extracted silhouettes, where cluttered background is not a main problem, it is a better descriptor. In our experiment setting, we use four level of

pyramids, eight orientation bins and 180 degrees of maximum orientation, as shown in figure 4.2.

4.1.3 SIFT

Scale invariant feature transform (SIFT) was proposed by D. Lowe to detect and describe local features [40] and is widely applied in classification, object detection and recognition thanks to its multiple scale descriptions. As a local feature descriptor, the most common way to describe a region of interest is to use bag of words representation. That is, SIFT descriptors extracted from all regions of interest are clustered, a vocabulary is computed as cluster centers, and each region of interest is denoted as a histogram by binning extracted SIFT into the vocabulary. In our experiment, we also use bag of word representation. We use k-means for clustering and the vocabulary size is defined as 250.

4.1.4 Human pose estimation error comparisons

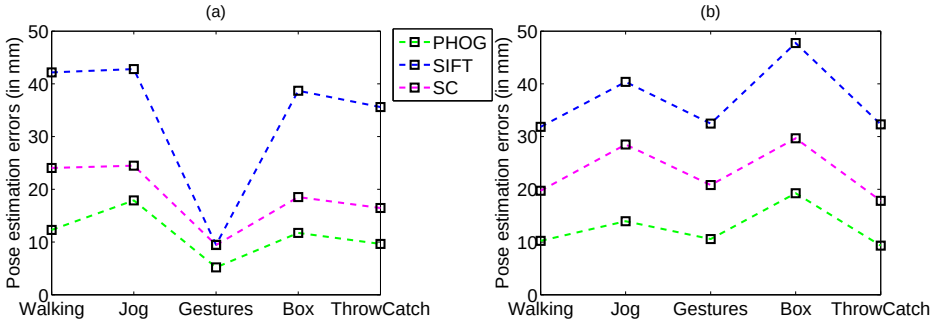


Figure 4.3: The Performance Comparison of three Feature Descriptor. (a) Pose estimation errors on sequences performed by actor "S1" and (b) actor "S2".

To compare the strongness of these descriptors with each other, we validate the aforementioned descriptors on HumanEva dataset. We take training data from the dataset including two actors ("S1" and "S2") performing five actions ("Walking", "Jogging", "Gestures", "Box" and 'ThrowCatch"). The training data are split into two sections: images with even frame numbers compose the training data and the rest of the images compose validation data. Every experiment has a specific action type and a certain performer. In each experiment, the regression model is trained with the training data in a motion sequence and validated on the test data from this sequence. The errors are average joint positions between the estimated poses and the ground truth 3D human poses. The pose estimation errors comparisons among all features are shown in figure 4.3.

From the figures, we can conclude that PHOG, among all the candidate feature descriptors gives the least pose estimation errors. The results are comprehensible, because PHOG with our setting is a denser descriptor compared with shape context and it has spatial information where line segments occur compared with bag of SIFT.

Shape context descriptor outperforms bag of SIFT descriptor in all experiments, although shape context has a much lower dimension (60) than bag of SIFT (250). This suggests that for pose estimation, the spatial information of a feature is a key factor. Another crucial factor that affects the performance is action type. All descriptors share similar performance trend when action types vary. In action "Gestures", where only one arm is moving, shape context (with pose estimation error of $9.43mm$) and bag of SIFT (with pose estimation error of 9.47) performs almost the same. The reason is when only one limb is moving, even without location information, descriptors from a motion sequences have a one-to-one mapping to the joint position output. While in other actions, like "Walking" and "Jogging", location information aids to decided which part of the body is in motion.

After the comparison, we can choose the feature descriptor with the best performance. Although extracted silhouettes with background subtraction are noisy because of camouflages and shadows, we didn't take any measure to process noise on the extracted silhouettes. In the following section, we propose a new feature descriptor designed to automatically discard noise from extracted silhouettes and compare with PHOG to check their robustness against different levels of noisy silhouettes.

4.2 Iterative Closest Points for Noisy Silhouettes

We sample points from extracted silhouettes as a compact representation from images. And a new distance measure is proposed for measuring distances between two set of point samples. The proposed measurement is modified based on iterative closet point (ICP) algorithm. The idea of ICP is to iteratively modify translation and rotation of a set of points, for example a point cloud, to match with another set of points. It is mainly used to register points between two point clouds from different scans. By automatically adjust a threshold, as in [93], iterative closest point algorithm is able to discard points on a point cloud that has no match to another point cloud. We introduce this idea and propose a new distance measurement which is able to discard unwanted noise on the extracted silhouette. The threshold by which 3D points are discarded is automatically adjusted according to noisy conditions of extracted silhouettes. Next, we are going to explain the new distance measurement and how to apply it to human pose estimation problem.

4.2.1 Iterative Closest Points as a Distance Measurement

Given two noisy silhouettes, represented by pixel positions along the silhouette edge, we first sample pixels to eliminate redundancy. The sampling ratio is set as one out of three so that no big gap is left between adjacent pixels. With the sparse pixel positions, we normalize x and y coordinates with the y range of the current frame. In this way, all pixel position are comparable with each other and also we keep the aspect ratio of the extracted silhouette.

With two noisy silhouettes represented as two clusters of normalized x and y coordinates, we can start the iterative comparison procedures. We fix one silhouette as a static template and another one move horizontally and vertically in a neighborhood area so that we can find the optimal translation of the moving silhouette. We con-

sider a point from the moving silhouette is matched to the static silhouette when their distance is below a certain threshold $\mathcal{D}$. And the distance between two silhouettes is computed as the average error of all matched points. Once the distance change between iterations is below a certain threshold, the iteration algorithm is terminated. We can use any kind of the optimization method for local search. In our implementation, we sample in different directions from the current values of the parameters and find the maximum decrease and set this direction as the gradient for the next movement.

By automatically adjusting the threshold, we can process point sets of different similarities. The automatic adjustment is as follows: the errors of all matched points are fit to a Gaussian distribution and the mean of the Gaussian is then compared with the threshold from the previous step, where μ is the fitted mean, σ is the fitted covariance and $\mathcal{D}$ is the threshold from the previous step (as in [93]): By adaptively

Algorithm 2 Automatic adjustment of $\mathcal{D}$

If $\mu < \mathcal{D}$
    $\mathcal{D} = \mu + 3 * \sigma$;
Else If $\mu < 3 * \mathcal{D}$
    $\mathcal{D} = \mu + 2 * \sigma$;
Else If $\mu < 6 * \mathcal{D}$
    $\mathcal{D} = \mu + \sigma$;
Else
    $\mathcal{D} = \epsilon$;

optimizing thresholds, the algorithm adapts to different noise levels. That is, if two point set are very different from each other, the threshold is enlarged according to fitted mean μ and covariance σ, so that still there are certain amount of points remained for matching after filtering. The automatic selection of the threshold is carried out in each iteration.

Matched silhouettes of different degree of similarities are as shown in figure 4.4 and figure 4.5. In the two figures, the upper left subfigure shows the original silhouettes to be matched, the right subfigure shows the static silhouette, the moved silhouette after optimal translation and the correspondences between matched points, and the lower left subfigure shows the fitted Gaussian distribution to the errors of all matched points. In figure 4.4, due to threshold filter, some points on the hand are filtered from the moving silhouette. Due to noisy parts along the silhouette other than the shadow, the shadow is kept after the filter. In the case where two silhouettes are very similar with each other but the part from the shadow, the shadow would be considered noise and eliminated partially or completely. Figure 4.5 shows an example where shadows are partially discarded because of small differences between two silhouettes.

4.2.2 Mapping Learning with Gaussian process regression model

As we introduce in chapter [**?**], for mapping 2D image features to 3D poses in this chapter, we also use Gaussian process regression (GPR) model. If we normalize the training output as zero mean, we only need to specify the covariance function for the

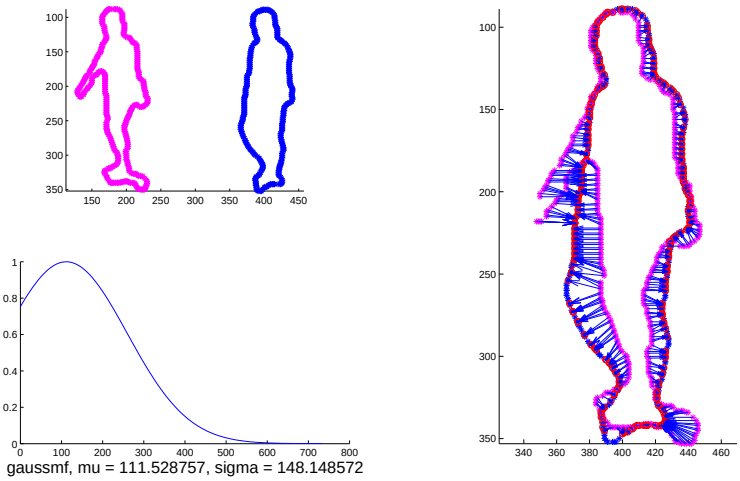


Figure 4.4: An example of two noisy silhouettes matched with AICPPSS method.

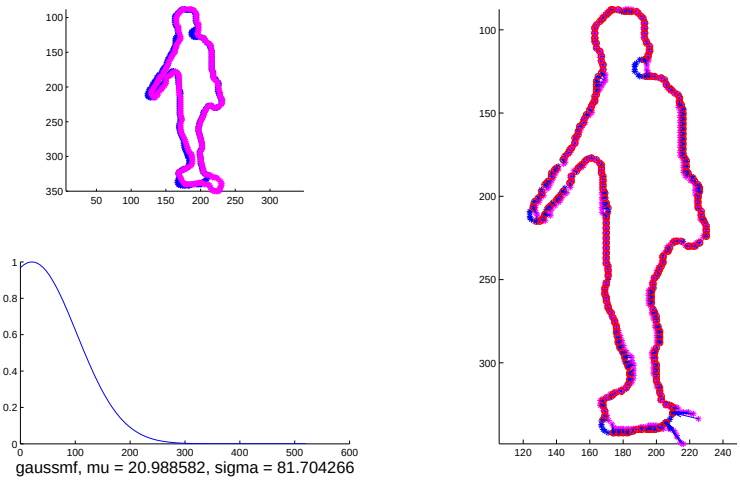


Figure 4.5: An example of two noisy silhouettes matched with AICPPSS method.

GPR model. A typical covariance function used for a Gaussian process is the squared exponential:

$$k(\mathbf{x_1}, \mathbf{x_2}) = \theta_1^2 \exp(-\frac{(\mathbf{x_1} - \mathbf{x_2})^2}{2\theta_2^2}) \tag{4.1}$$

where θ_1, θ_2 are called the amplitude and lengthscale hyperparameters respectively. This covariance function implicitly define the closeness of two function values $f(x_1)$

and $f(x_2)$ x_1, x_2. The covariance of $f(x_1)$ and $f(x_2)$ is a small value, if x_1, x_2 are very different from each other. We use this covariance function for all commonly used feature, for example, PHOG. However in the case of iterative closet point as a distance measurement, the distance measure itself specifies the closeness between data samples. To keep the properties of input features, we use a relatively simple covariance function: linear covariance function. In the following subsection, we compare the performance of the two combinations for pose estimation problem.

4.2.3 Comparisons between PHOG and ICPNS

From the above sections, we explain the two compared two methods. The first uses PHOG as input features, and set covariance function of GPR as squared exponential. The second one takes point samples from the extracted silhouettes as input, measures distance with iterative closest point and sets covariance function as linear covariance function. Here we explain the experiment setting and the performance comparison of these two methods.

The experiments are set up with the following steps:

1. We take a walking sequence (with 1171 frames) from HumanEva dataset and visually pick clean silhouettes without camouflage and shadows (81 silhouettes);
2. All the clean silhouettes are grouped into clusters, where every pair of silhouettes within a cluster is below a threshold (in our experiment, the threshold is 0.0092);
3. We randomly pick one from each cluster and compose the training set of our experiment (63 silhouettes);
4. All silhouettes expect the training silhouettes compose the test set.
5. We compute distances between all silhouettes in the test set and all silhouettes in the training set, and split the test set into two noise level. For those test silhouettes, whose minimum distance to the training set is below a threshold (0.015), compose noise level one (425 silhouettes) and the rest compose noise level two (683 silhouettes).

By comparing experimental results carried on these two test sets, we can see the response of a method to different levels of noisy silhouettes.

Table 4.1 shows the pose estimation performance of PHOG with point samples feature. "PHOG+SE" stands for the first method of using PHOG as input and squared exponential as the covariance function for GPR. "PS+ICPNS" stands for the second method of using point samples of silhouettes as input, iterative closest point as measurement and linear function as covariance function for GPR. The difference between the estimated poses and the ground truth poses are measured with joint position differences (in mm) and limb angle differences (in degree). From the table, we can see that with lower noise level "PHOG+SE" performs better than "PS+ICPNS" due the precise description of this feature and the effectiveness of squared exponentional kernel. But when the noise level increases, "PS+ICPNS" outperforms "PHOG+SE" thanks to its robustness against noise. In experiment section, we will further validate these two methods in a public dataset with variant experimental setting.

Noise Level	PHOG+SE		PS+ICPNS	
	mm	[°]	mm	[°]
Level 1	64.18	2.79	69.31	3.18
Level 2	93.83	3.61	87.82	3.59

Table 4.1: The Composition of PHOG with Point Samples Feature Measured with Iterative Closest Point. "PHOG+SE" stands for the first method of using PHOG as input and squared exponential as the covariance function for GPR. "PS+ICPNS" stands for the second method of using point samples of silhouettes as input, iterative closest point as measurement and linear function as covariance function for GPR.

4.3 Experiments

Action	S1 Training/Test	S2 Training/Test	S3 Training/Test
Walking	1171/999	871/1088	890/800
Jogging	434/869	790/722	826/859
Gestures	796/1065	681/1057	209/548
Box	497/601	463/984	928/748
ThrowCatch	217/946	804/1394	\

Table 4.2: The composition of experiment data from HumanEva-I dataset. Numbers of training and test poses from three actors performing five actions.

The proposed method of "PS+ICPNS" for 3D Human Pose Estimation is validated quantitatively on the HumanEva-I[1] and compared with "PHOG+SE" method. The experiments on the HumanEva-I are designed to include different five actions ("Walking", "Jogging", "Gestures", "Box" and "ThrowCatch") and three performers ("S1", "S2" and "S2").

The composition of the training and test data is shown in table 4.2. Methods are validated on 15 different exeriment settings: five different actions performed by three actors as aboved mentioned. For each experiment setting, we use the training set from HumanEva-I dataset and the online evluation system for test. The numbers of training and test frames for each experiment are shown in the table. Subsection 4.2.3 gives us a detailed comparison between two feature descriptors with different noisy levels. In this section, we aim to validate these two features on the aforementioned dataset. The method in subsection 4.2.3 is a little bit cumbersome for running on a batch of experiments, so we simply it and take all the training frames as it is. That is, we consider all extracted silhouettes from training frames are non-noisy input. Thus, the comparison of experiment results on test set also relfects the variance between the training and the test.

With the experiment setting shown in table 4.3, we compare the performance of the aforementioned two methods. We can see from the table that, in most cases, the

[1] http://vision.cs.brown.edu/humaneva/

Features	Actions	S1	S2	S3
PHOG+SE	Walking	124.5	150.2	**157.4**
	Jogging	131.8	**121.5**	93.6
	Gestures	29.5	125.7	74.1
	Box	94.0	138.1	157.3
	ThrowCatch	\	107.1	\
	Average	94.95	128.5	**120.6**
PS+ICPNS	Walking	**107.2**	**121.8**	199.3
	Jogging	**90.6**	140.7	**81.3**
	Gestures	**25.8**	**91.4**	**70.1**
	Box	**72.3**	**106.3**	**135.6**
	ThrowCatch	\	**90.8**	\
	Average	**73.98**	**110.2**	121.6

Table 4.3: Comparison of 3D pose reconstruction errors between the proposed AWGPR method and the original GPR model. For AWGPR, we use SIFT and PHOG features. GPR models are trained with SIFT and PHOG features separately. Errors are measured with joint angel difference (in degrees) and joint displacement measurement (in mm) between the ground truth data and the estimated data.

proposed "PS+ICPNS" method outperforms "PHOG+SE" method. The maximum boost is 41.2mm in "Jogging" performed by actor "S1". And the average boost with "PS+ICPNS" method for actor "S1" is 21.0mm. Note that there are also two cases where "PHOG+SE" outperforms "PS+ICPNS". We interpret that this is due to the different qualities of the training and the test data. From the comparisons, we conclude that the proposed "PS+ICPNS" method outperforms traditional "PHOG+SE" method in the experiment settings and the extracted test silhouettes are quite different from the extracted training silhouettes. We further show per frame error difference comparison in the following paragraph.

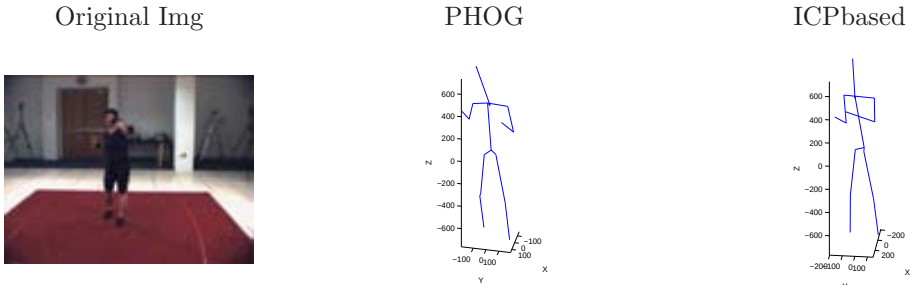


Figure 4.6: Examples of estimated 3D poses from frame 225 of actor "S1" performing action "Box". The first column shows the original image, the second column shows estimated poses from PHOG feature, the third columns shows estimated poses from the proposed ICP based method.

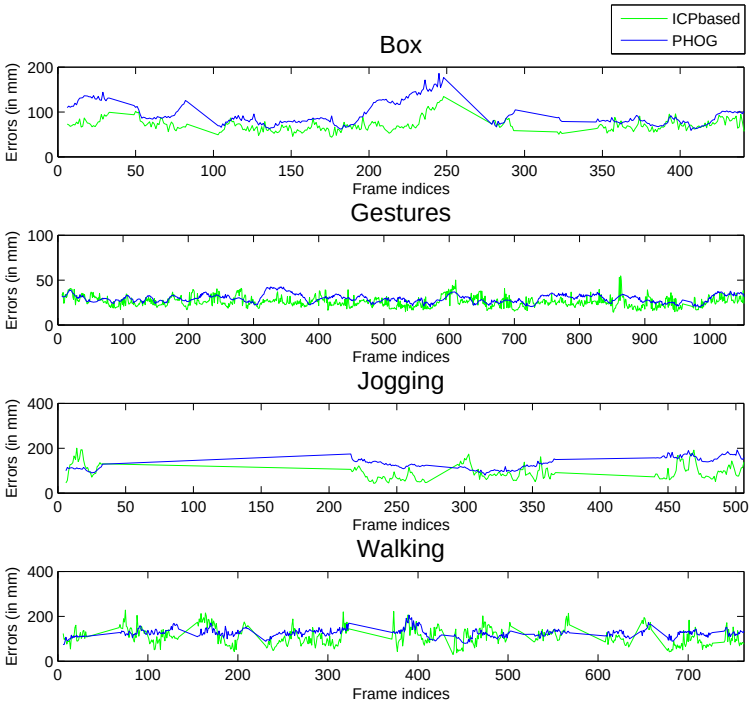


Figure 4.7: Average joint position test error per frame for two input features, four actions and one actor ("S1").

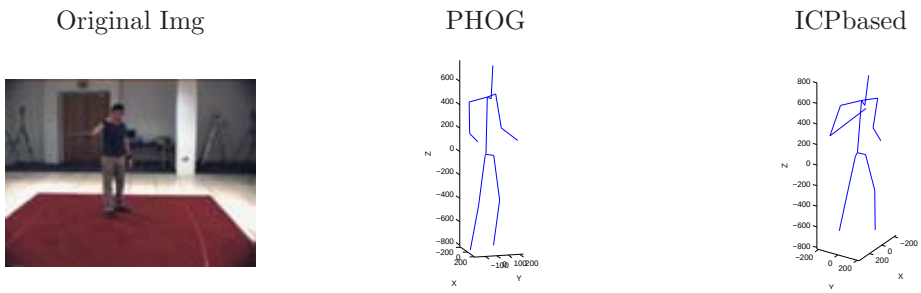


Figure 4.8: Examples of estimated 3D poses from frame 300 of actor "S2" performing action "Gestures".

Figure 4.7 and figure 4.9 compare average joint position errors per frame for "PHOG" and "ICP based" input features on all actions. "PHOG" corresponds to "PHOG+SE" and "ICP based" corresponds to "PS+ICPNS" in subsection 4.2.3. We can see that "ICP based" method has an absolute advantage compared with "PHOG" method in action "Box" for both actor "S1" for actor "S2". Also this holds for ac-

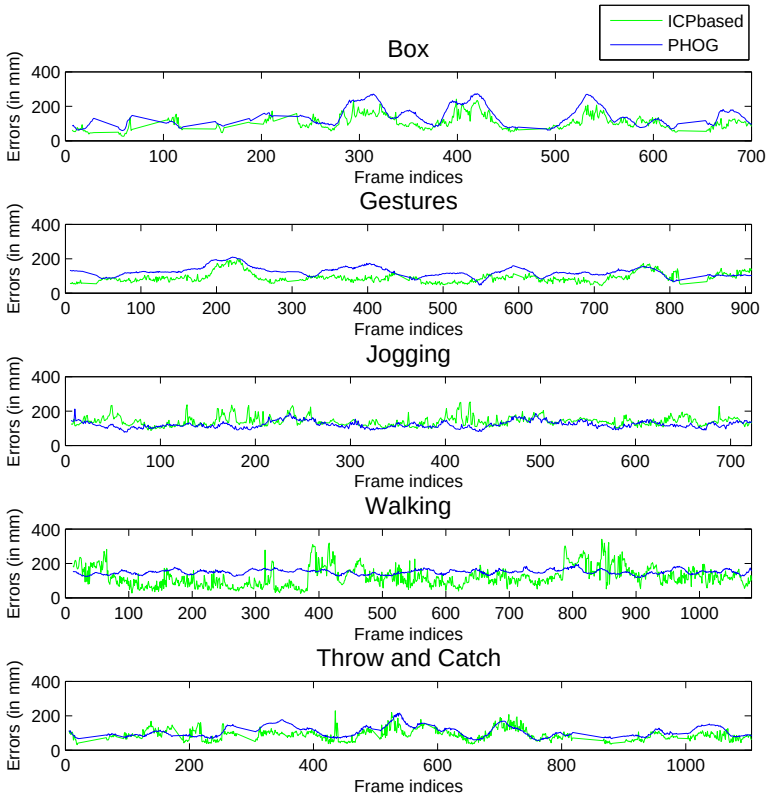


Figure 4.9: Average joint position test error per frame for two input features, four actions and one actor ("S2").

tion "Gestures" by actor "S2". We further visualize some estimated poses for certain frames in test sequence in figure 4.6 and figure 4.8. From the visualized results, we can see that the "ICP based" method outperforms "PHOG" method in estimating more accurate 3D poses.

4.4 Conclusions and future work

In this chapter, we compare several popular image features for monocular pose estimation problem, including shape context, SIFT with bag of words representation and PHOG. In our feature setting, PHOG outperforms all other image features based on pose estimation accuracies. Then we proposed a new distance measurement for sample points on extracted silhouettes. And split the validation silhouettes into two different noisy level and compare the performance of PHOG and the proposed measure

from point sample within a Gaussian process regression model. The experiment shows that the proposed feature measure is more robust against noisy inputs. We further validate these two features on a public dataset: HumanEva data. In all experiments except two, the proposed method outperforms PHOG.

In future work, it would be interesting to enhance GPR, by using more modified GPR model, like GPR with output structure, like in [11], to enhance prediction accuracies. The shortcoming of the original GPR model, is that each output dimension of GPR is separately trained and predicted. The idea of [11] is to maintain the structure between different dimensions of the output. Authors in [11] achieve this goal by fixing all output dimensions expect the current to-be-optimized one, and feed input together with output (expect one) to the input of the GPR. There are also other ways to keep the output structure of prediction, for example, proposing a kernel [44] with output structure. In the future work, we could first experiment on these two different types of methods and maybe modify or enhance the performances according to our problem.

Chapter 5

Action Recognition

One important application of pose estimation is per-frame initialization for human tracking. Also researchers solve action recognition problem by incorporating poses as a latent variable. In this chapter, we explore the effect of human poses explicitly encoded as a module in action recognition problem. In this way, we can figure out what is the explicit impact of human poses to action recognition problem.

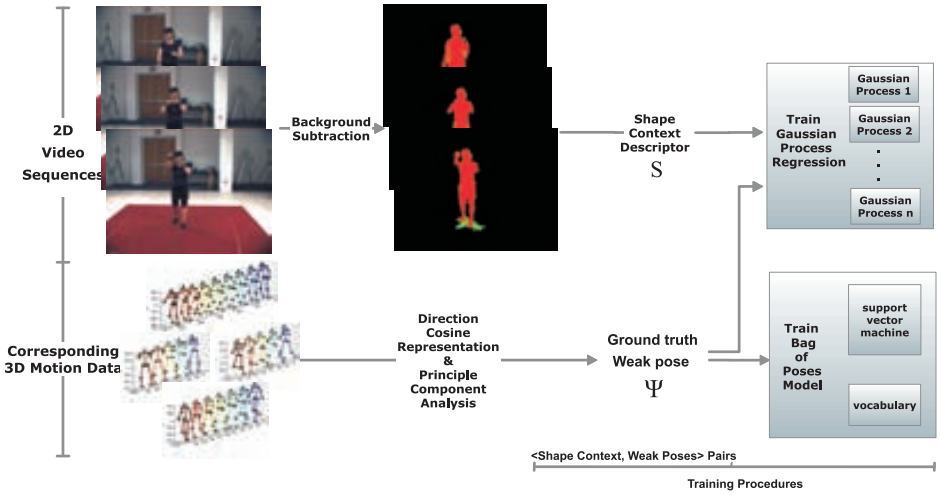


Figure 5.1: Learning step: we train Gaussian processes to learn the regression function from shape context descriptors to *weak poses*. In parallel, a BoP model is built for each action class by extracting key poses and training SVM classifiers.

The whole procedure presented in this chapter is shown in figs 5.1 and 5.2. In essence the method is composed of two steps: training and prediction. In training, a set of Gaussian processes (first row fig. 5.1) and the Bag of Poses (BoP) model (second row fig. 5.1) are learnt. In training, a set of Gaussian processes (first row fig. 5.1) and the Bag of Poses (BoP) model (second row fig. 5.1) are learnt. On

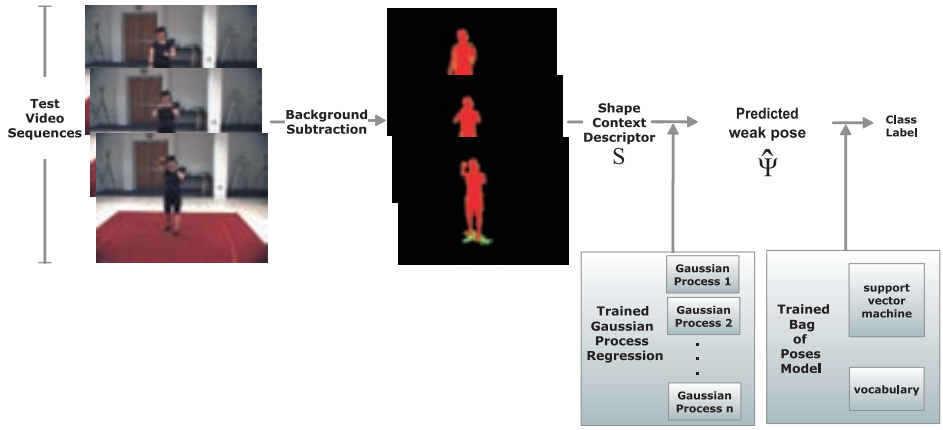


Figure 5.2: Predicting phase. The test video sequence is described using shape context descriptors as in the learning phase (see fig. 5.1). *Weak poses* are predicted from shape context descriptors using trained Gaussian processes and the video is represented as a histogram of the vocabulary learned in the training phase. The video is finally labeled using the ensemble of trained SVMs for each action class.

one hand, Gaussian processes are trained with pairs of 2D image features and our intermediate 3D pose representation or *weak poses*. For each dimension of the *weak pose* parameter space, we define a Gaussian process to map from 2D image features to this particular dimension. On the other hand, the BoP model is trained with *weak poses* and motion sequences. We introduce temporal information in BoW by grouping consecutive video frames. Similar to graphical models which account for the influence of neighboring data, in our case we take into account those neighboring frames by merging consecutive frames in a single word. After choosing the most representative *weak poses* for the vocabulary, each motion sequence is represented as a histogram and SVMs are finally trained. In the prediction step, given an unknown video sequence, we predict human poses with the trained set of Gaussian processes, and represent the video sequence using the histogram of the vocabulary. After that, we label the action by the trained SVMs.

The rest of the chapter is organized as follows: next section introduces our human body model and human posture representation; section 5.2 describes how we use a set of Gaussian processes for learning the mapping from 2D image features to 3D human poses; in section 5.3, we describe a procedure for incorporating temporal information in a BoW schema, showing the results in section 5.4. Finally section 5.5 presents the future avenues of research.

5.1 Data representation

The flexibility of the human body and the variability of human actions produce high-dimensional motion data. Given a number of video sequences of a single actor execut-

ing certain actions, in training each image has its corresponding 3D motion capture data. How to represent these data in a compact and effective way is also a challenge.

We select a compact representation of human postures in 3D, in our case a stick figure of twelve limbs. For representing 3D motion data, a human pose is defined using twelve rigid body parts: hip, torso, shoulder, neck, two thighs, two legs, two arms and two forearms. These parts are connected by a total of ten inner joints, as shown in fig. 5.3(a). Body segments are structured in a hierarchical manner, constituting a kinematic tree rooted at the hip, which determines the global rotation of the whole body.

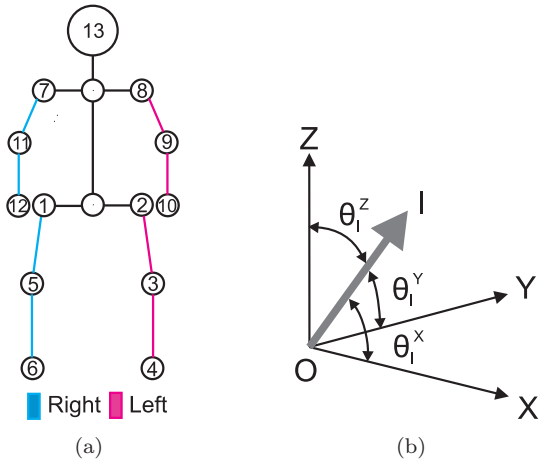


Figure 5.3: (a) The 3D stick figure model used for representing human pose. Ten principal joints corresponding to the markers used in motion capture are used [65]. (b) The angles $(\theta_l^x, \theta_l^y, \theta_l^z)$ between the limb l and the axes [59].

Although some works only consider the 3D position of the markers at each time step [42, 41, 56], others have explored representations like polar angles [30] or Direction Cosines (DCs) [59]. In the latter case, the orientation of each limb is represented by three direction cosines of the angles formed by the limb in the world coordinate system. DCs embed a number of useful invariants, and by using them we can eliminate the influence of different limb lengths. Compared to Euler angles, DCs do not lead to angle discontinuities in temporal sequences. Lastly , DCs have a direct geometric interpretation which is an advantage over quaternions [92].

So we use the same representations for human postures and human motions as in [59]: a limb orientation is represented using three parameters, without modeling self rotation of the limb around its axes, as shown in fig. 5.3(b). This results in a 36-D representation of the pose of the actor in frame j of video i:

$$\psi_j^i = [\cos\theta_1^x, \cos\theta_1^y, \cos\theta_1^z, \ldots, \cos\theta_{12}^x, \cos\theta_{12}^y, \cos\theta_{12}^z], \tag{5.1}$$

where θ_l^x, θ_l^y and θ_l^z are the angles between the limb l and the axes as shown in fig. 5.3(b).

With direction cosines, we represent the motion sequence of the i-th video as a sequence of poses:

$$\mathbf{\Psi}_O^i = [\psi_1^i, \psi_2^i, \ldots, \psi_{n_i}^i], \tag{5.2}$$

where n_i is number of poses (frames) extracted from video i.

5.1.1 Universal Action Space or *UaSpace*

Since natural constraints of human body motions lead to highly correlated data [91], we build a more compact, non-redundant representation of human pose by applying Principle Component Analysis (PCA). This universal action space (*UaSpace*) will become the basis for vocabulary selection and finally classification using BoP.

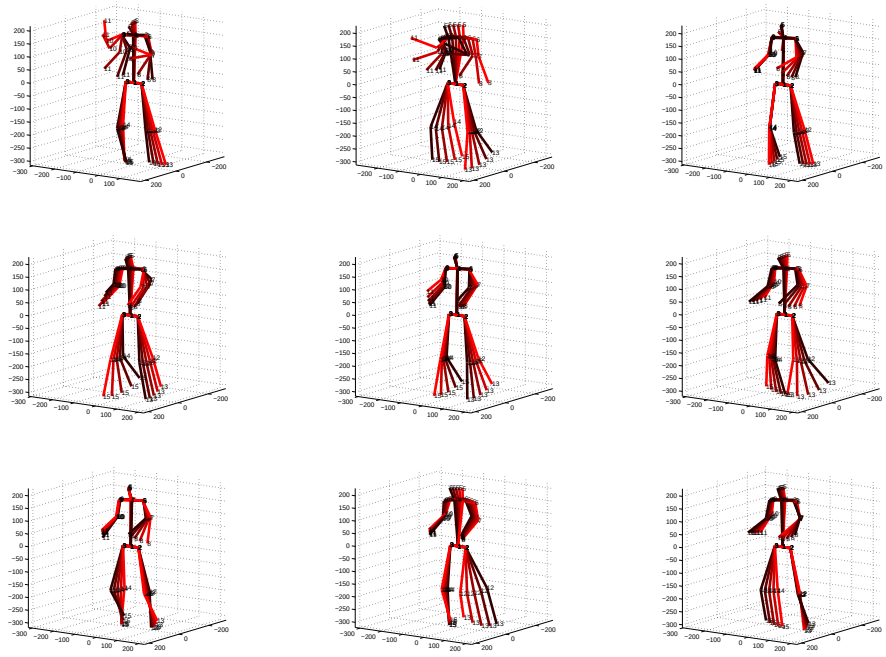


Figure 5.4: Visualizing the 9 principal variations of the pose within *UaSpace* learnt from HumanEva data. Each plotted stick figure is a re-projected pose by moving it in one eigenvector's dimension from -3 up to 3 times the standard deviation.

By projecting human postures into the *UaSpace*, distances between poses of different actions can be computed and used for classification. Fig. 5.4 shows pose variation corresponding to the top (in terms of eigenvalues) 9 eigenvectors in the *UaSpace*. From the figure, one can see which pose variations each eigenvector accounts for in the eigenspace decomposition. For example, one can see that the first eigenvector corresponds to the characteristic motion of the arms and the second eigenvector corresponds to the motion of the torso and the legs. In the following section, we describe

how *weak poses* are estimated from video frame feature descriptors using GPR.

We denote the pose representation in the reduced dimensionality space as *weak poses* or ψ', and the motion sequence of *UaSpace* the i-th video is represented as:

$$\mathbf{\Psi}^i = [\psi_1^{i\prime}, \psi_2^{i\prime}, \ldots, \psi_{n_i}^{i\prime}], \tag{5.3}$$

where $\psi_j^{i\prime}$ is the *weak pose* corresponding to the j-th image frame in i-th video sequence.

5.2 *Weak pose* estimation using GPR

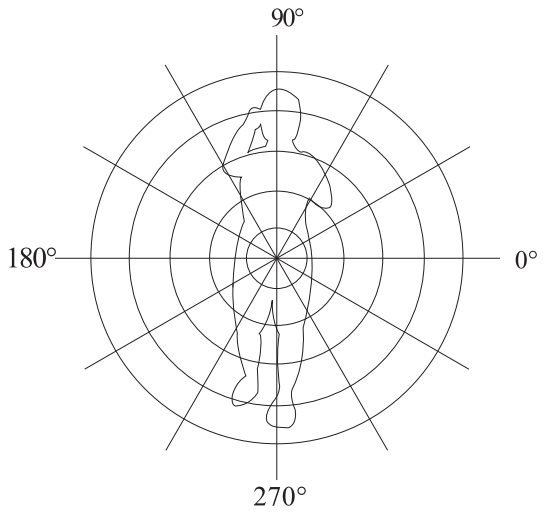


Figure 5.5: Radial coordinates for shape context descriptor. The origin of the polar coordinate system is placed on the centroid of the bounding box of the silhouette. The radius is divided equally into 5 bins and the circle is divided equally into 12 bins.

We use Shape Context Descriptor (SCD) to represent the human silhouette found using background subtraction [6]. Shape context is commonly applied to describe shapes given silhouettes [46, 3], and have been proven that it is an effective descriptor for human pose estimation [55].

The main idea of our SCD is to place a sampled point on a shape in the origin of a radial coordinate system and then to divide this space into different range of radius and angle. In this way, the number of points that fall in each bin of the radial coordinate system are counted and encoded into a bin of an histogram. In our experiments, we place the origin of radial coordination on the centroid of a silhouette and divide radius into 5 bins equally spaced and divide angle into 12 equally spaced bins, as shown in fig. 5.5. As a result, the SCD vector is 60-D. Figure 5.6 shows examples of extracted silhouettes of actor "S1" performing action "Box" and action "Gesture". From the figure, we can see that background subtraction with the method

(a) Box

(b) Gestures

Figure 5.6: Samples of extracted silhouettes of actor "S1" performing action "Box" and "Gesture" with the method in [6]. Silhouette centroids are marked in red square.

in [6] gives promising background results. We set the centroid of the silhouette as the center of the local coordinate system, and the largest diameter is set as 1.25 times the diagonal length of the silhouette bounding box.

The normalization of the resulting SCD has a significant impact on the performance of Gaussian process regression. We exploit two different ways of normalizing data: standard deviation and individual normalizations. Suppose $\mathbf{s}_{orig}$ denotes the original shape context descriptor from one image, and

$$\mathbf{s}_{orig} = [np^1, np^2, \ldots, np^i, \ldots, np^{60}], \tag{5.4}$$

where np^i is the number of pixels that fell in the i-th bin.

In standard deviation based normalization, we calculate standard deviations from all training shape context descriptors $\mathbf{std} = [std^1, std^2, \ldots, std^{60}]$. Then we normalize each dimension of the shape context descriptor by dividing it with the corresponding standard deviation. If we represent the normalized shape context descriptor as $\mathbf{s}_{normlized}$, then

$$\mathbf{s}_{norm1} = [\frac{np^1}{std^1}, \frac{np^2}{std^2}, \ldots, \frac{np^i}{std^i}, \ldots, \frac{np^{60}}{std^{60}}] \tag{5.5}$$

In individually normalizing method, we divide the pixel number in a bin by the total pixel number of the shape context descriptor. That is, if we represent the total number of pixels in one shape context descriptor as $npSum$, then in individually normalizing method, the normalized shape context descriptor is defined as:

$$\mathbf{s}_{norm2} = [\frac{np^1}{npSum}, \frac{np^2}{npSum}, \ldots, \frac{np^i}{npSum}, \ldots, \frac{np^{60}}{npSum}]. \tag{5.6}$$

We compare these two different ways of normalizing shape context descriptors in experimental results.

5.2.1 Gaussian Process Regression

The method described in [1] predicts 3D poses from 2D image features using Relevance Vector Machine (RVM). RVM is more efficient during learning, but less accurate since RVM is a special case of GPR: during the learning phase, RVM takes the most representative training samples while GPR takes all training samples. Additionally, GPR has been successfully applied to pose estimation and tracking problems, for example [76, 75]. So in our approach, we will use GPR for modeling the mapping between silhouettes and *weak poses*.

According to [20], Gaussian process is defined as: *a collection of random variables, any finite number of which have (consistent) joint Gaussian distribution.* A Gaussian process is completely specified by its mean function and a covariance function. Please refer to chapter 2.6 for its definition, detailed explanations and an example of applying GPR in solving prediction problem. Integrating with our problem, we denote the mean function as $m(\mathbf{s})$ and the covariance function as $k(\mathbf{s}, \mathbf{s}')$, so a Gaussian process is represented as:

$$\zeta(\mathbf{s}) \sim \mathcal{GP}_j(m(\mathbf{s}), k(\mathbf{s}, \mathbf{s}')), \tag{5.7}$$

where

$$\begin{aligned} m(\mathbf{s}) &= E[\zeta(\mathbf{s})], \\ k(\mathbf{s},\mathbf{s}') &= E[(\zeta(\mathbf{s})-m(\mathbf{s}))(\zeta(\mathbf{s}')-m(\mathbf{s}'))], \end{aligned} \tag{5.8}$$

We set a zero-mean Gaussian process whose covariance is a squared exponential function with two hyperparameters controlling the amplitude θ_1 and characteristic length-scale θ_2:

$$k_1(\mathbf{s},\mathbf{s}') = \theta_1^2 \exp(-\frac{(\mathbf{s}-\mathbf{s}')^2}{2\theta_2^2}). \tag{5.9}$$

We assume prediction noise as a Gaussian distribution and formulate finding the optimal hyperparameters as an optimization problem. We seek the optimal solution of hyperparameters by maximizing the log marginal likelihood (see [20] for details):

$$\log p(\Psi'|\mathbf{s},\theta) = -\frac{1}{2}\Psi'^T K_{\Psi'}^{-1}\Psi' - \frac{1}{2}\log|K_{\Psi'}| - \frac{n}{2}\log 2\pi, \tag{5.10}$$

where $K_{\Psi'}$ is the calculated covariance matrix of the target vector (vector of training *weak poses* in *UaSpace*) Ψ' under the kernel defined in equation 5.8.

With the optimal hyperparameters, the prediction distribution is represented as:

$$\begin{aligned} \Psi'^*|\mathbf{s}^*,\mathbf{s},\Psi' \sim\ & \mathcal{N}(\mathbf{k}(s^*,\mathbf{s})^T[K+\sigma_{noise}^2 I]^{-1}\Psi', \\ & k(s^*,s^*)+\sigma_{noise}^2 - \mathbf{k}(s^*,\mathbf{s})^T[K+\sigma_{noise}^2 I]^{-1}\mathbf{k}(s^*,\mathbf{s})), \end{aligned} \tag{5.11}$$

where K is the calculated covariance matrix from training 2D image features $\mathbf{s}$ and σ_{noise} is the covariance of Gaussian noise. We train a set of Gaussian processes to learn regression from SCD to each dimension of the *weak poses* separately.

So the number of Gaussian processes m equals dimensions of *weak poses*. In our method, m is an important factor because too few Gaussian processes will not be able to reach an ideal classification result while too many Gaussian processes are burdens for computation. We use cross validations on training data to fix the value of m. We will explain in detail how to fix m in section 5.4.2.

Given an unseen test video sequence, we extract silhouettes and describe them using shape context as for training data. Using trained Gaussian process, we predict weak poses in *UaSpace* (refer to [20] for detailed prediction procedures). Based on predicted weak poses in *UaSpace* for all frames of test video sequences, now we can label the test video sequence in *UaSpace*. In the next section, we explain how to compute vocabulary and train support vector machine and finally label actions in this space.

5.3 Bag of Poses for action recognition

Given a test video sequence, we extract SCDs from image sequences and then predict the *weak pose* by the set of trained Gaussian processes. With the predicted *weak poses*, the problem turns into a classification problem in the *UaSpace*.

Inspired by BoW [39, 38, 15], we apply the following steps for action recognition: compute descriptors for input data; compute representative *weak poses* to form vocabulary; quantize descriptors into representative *weak poses* and represent input data as histograms over the vocabulary, a Bag of Poses (BoP) representation. Next we explain how to compute the vocabulary and perform classification with our modified BoP model.

5.3.1 Vocabulary selection

The classic BoW pipeline uses k-means for calculating the vocabulary. But this way of calculating the vocabulary does not give promising action recognition results [28]. While energy-based method proposed in [28] gives comparatively better results when applied for each action separately, it is not applicable here. Because the number of key poses calculated from energy-based method is closely related with numbers of motion cycles. When we use one vocabulary for all actions, key pose numbers increases dramatically. While the number of training sequences stays the same. Even we use techniques to create new training sequences, the experiment results are not ideal.

We combine these two methods and propose a new method for computing the vocabulary. First, we select candidate key *weak poses* using energy optimization as in [28]. The key *weak poses* are pre-selected as:

$$F^i_{pre} = \{f^i_1, f^i_2, , \ldots, f^i_l\}, \tag{5.12}$$

where f^i_j corresponds to local maximum or local minimum energies in i-th motion sequence. And l is the total number of local maximum and local minimum values. Note, l is not a fixed value, and it depends on number of motion cycles and motion variations in the sequence.

Without taking into account temporal information, we cluster all preselected key *weak poses* from all performances: $F_{pre} = \{F^1_{pre}, F^2_{pre}, \ldots, F^p_{pre}\}$, where F^i_{pre} is calculated as in equation 5.12 and p is the number of training motion sequences. Then, we select k most representatives *weak poses* F_k from F_{pre} with k-means. So F_k makes the vocabulary. We call the proposed method as energy-k-means. We will show in experiment section comparisons between the energy-k-means, k-means and energy-based method.

To incorporate temporal information into our solution, we consider d consecutive frames as one unit. That is, key *weak poses* with temporal information are preselected as

$$F^t_{pre} = \{F^{t1}_{pre}, F^{t2}_{pre}, \ldots, F^{tl}_{pre}\}, \tag{5.13}$$

where

$$F^{tj}_{pre} = [f^{frm-d+1}_j, f^{frm-d+2}_j, \ldots, f^{frm}_j] \tag{5.14}$$

is the j-th candidate for key *weak poses*. F^{tj}_{pre} is a concatenation of d consecutive *weak poses* and f^{frm}_j corresponds to local maximum or local minimum energies in j-th motion sequence, and tl equals the total number of preselected key *weak poses*. Then, the vocabulary is calculated as k-means clustering centers F^t_k from F^t_{pre}.

Temporal step d is a critical factor. Experimental results show that, for *weak poses*, after temporal step d reaches a certain value, classification results remain comparatively steady. In section 5.4.2, we will show how we fix d using cross validation on training data.

5.3.2 Action Classification

A vocabulary is calculated as a collection of characteristic key *weak poses*. Then we represent our motion sequences statistically as occurrences of these characteristic key *weak poses*, that is, histograms over the vocabulary. To be specific, the i-th motion sequence $\mathbf{\Psi}^i$ represented as in equation 5.3 in *UaSpace* can be represented statistically as:

$$hist^i = [n_1, n_2, \ldots, n_j, \ldots, n_{tk}], \tag{5.15}$$

where n_j is the number of *weak poses* in $\mathbf{\Psi}^i$ that are nearest (Euclidean distance) to j-th word in vocabulary F_k. To incorporate temporal information, we start from d-th frame of video sequence V^i, and compare a concatenation of consecutive d *weak poses* with each entry of the vocabulary F_k^t. And tk in equation 5.15 is the number of words contained in vocabulary F_k^t.

For each action, we train a SVM with histograms and their corresponding action class labels. We choose a linear kernel according to experimental results and use cross validation to fix the cost value as 5. For measuring classification results, we use classification accuracy:

$$accuracy = \frac{tp + tn}{tp + tn + fp + fn}, \tag{5.16}$$

where tp, tn, fp, fn refer to true positive, true negative, false positive and false negative respectively. $tp + tn$ represents correctly classified samples, and $tp + tn + fp + fn$ is the total number of all samples. We use this criterion as the maximizing target when we do cross validation to fix parameters, for example, number of Gaussian process m and temporal step size d. With the computed vocabulary, we compute the After calculating vocabulary, we can represent sequences of motion capture data as a histogram of this vocabulary by counting the occurrence of the vocabulary. In learning phase, we train a support vector machine with training histogram (see figure 5.1) and in predicting phase, we predict action labels with the trained support vector machine (see figure 5.2).

Our hypothesis for incorporating temporal correlation between video frames is that for action recognition, a motion unit, here consecutive d poses, is more representative than a single human pose. We name d temporal step size. In our implementation, we start from d-th frames of each video sequence and for each assignment step, we take $d - 1$ historical frames together with the current frame so that we are not only considering the current pose but the variations of these d frames. We use cross validation to get the optimum d value for our motion capture data. Our experiment results will show that this way of incorporating temporal information gives promising improvement in human action recognition accuracy.

5.4 Experimental results

To verify robustness of our method, we choose two public datasets: HumanEva and IXMAS. [51] gives state of art action classification accuracy for HumanEva dataset. We will compare with this result with our experiments on this dataset. There are several related works on action recognition with IXMAS dataset, for example [79, 36, 69, 26]. Authors of [33] listed all state of art experimental results on this dataset. Among all, we will compare with experimental results in [26], because this method uses single viewpoint as input like our method while other methods need multiple viewpoints.

The composition of the data are:

1. HumanEva [1] dataset. This dataset contains six actions: "Walking", "Jog", "Gesture", "Throw/Catch", "Box", and "Combo". We consider the first five actions, since "Combo" is a combination of "Walking", "Jog", and "Balancing on each of two feet". Four actors perform all actions a total of three times each. Trial 1 has both video sequences and 3D motion data; in trial 2, 3D motion data are withheld for testing purposes; trial 3 contains only 3D motion data.

2. IXMAS [2] dataset. We further apply trained models from HumanEva dataset to IXMAS dataset, to test robustness of our method. From this dataset, we take four actions: "Walk", "Wave", "Punch" and "Throw A Ball". They correspond to actions "Walking", "Gesture", "Box" and "Throw/Catch" in HumanEva dataset.

There are in total 3 trials of the same action acted by every performer in Humaneva dataset. In our experiments, we only use the first trial for training, because this trial has both 2D image sequences and 3D motion capture data. Training frame number from a video depends on the length of the image sequence and the number of valid 3D motion capture data. Subsections are organized as follows: in subsection 5.4.1, first, we show the model training phase (refer to figure 5.1), that is cross validation process to fix number of Gaussian processes m and temporal step size d (refer to subsection 5.3.2), then, we show weak pose estimation error of Gaussian process regression compared with relevance vector machine, at the end, we explain our modified algorithm of Gaussian process to improve efficiency for learning; in subsection 5.4.3 we show action recognition accuracy of our method on Humaneva data set and on IXMAS data set. We take only the frontal view from the two dataset. Note that positions of vision cameras in these two dataset of frontal view are not set exactly the same.

5.4.1 Model training

In our experiments, we take the first half of each performance for training $< \mathbf{S}, \mathbf{\Psi} >$ and the second half for validation $< \mathbf{S}_{Val}, \mathbf{\Psi}_{Val} >$ and use cross validations to fix model parameters like number of Gaussian processes, vocabulary size, temporal step sizes and so on.

[1] http://vision.cs.brown.edu/humaneva/
[2] http://4drepository.inrialpes.fr/public/viewgroup/6

Action	Performer	Trial	Training	Validation	Total frames
Walking	1, 2, 3	1	586, 436, 445	585, 435, 445	2932
Jog	1, 2, 3	1	217, 395, 413	217, 395, 413	2050
Gesture	1, 2, 3	1	398, 341, 105	398, 340, 104	1686
Throw/Catch	1, 2	1	109, 402	108, 402	1021
Box	1, 2, 3	1	249, 232, 464	248, 231, 464	1888

Table 5.1: The composition of training data from Humaneva data set. Training data are composed of the first trial from the three performers performing five different actions. We list frames numbers for all training and validation sequences. Each number in "Training" and "Validation" columns denote frame number of the corresponding motion sequence. Total frames is the sum of all frames for one action including training and validation data.

The composition of training performances is shown in table 5.1.

5.4.2 Energy-k-means method for vocabulary computation

In this section, we compare the proposed energy-k-means method with the traditional k-means and energy-based method proposed in [28]. Table 5.3 shows that the

Methods	Voc size	Number of Gaussian processes			
		3	6	10	20
Energy-k-means	5	73.9	86.8	86.3	86.1
	10	67.7	83.6	82.9	84.4
	15	64.1	83.9	82.6	85.4
	20	64.7	79.0	77.5	78.4

Table 5.2: Comparisons of classification accuracy (%) among different vocabulary calculation methods: energy-k-means, k-means and energy-based method in [28].

proposed energy-k-means method outperforms k-means and energy-based method in all experiment configurations. While for k-means and energy-based method, proper parameter settings are needed for better results. For example, with *10* Gaussian processes, k-means outperforms energy-based method when the vocabulary size equals *10*, while energy-based method performs better when the vocabulary size equals *5*, *15* and *20*. The reason that the energy-based method does not give promising results is the big number of vocabulary size 5.4. Although we synthesize training data, still the number of training sequences (714) are not enough.

Number of Gaussian processes

x We train a set of Gaussian processes to learn mappings between shape context descriptors and *weak poses* in *UaSpace* with the training data $< \mathbf{S}, \mathbf{\Psi} >$. We calculate

Methods			Number of GPs			
			3	*6*	*10*	*20*
Energy-k-means	**Voc size**	*5*	73.9	86.8	86.3	86.1
		10	67.7	83.6	82.9	84.4
		15	64.1	83.9	82.6	85.4
		20	64.7	79.0	77.5	78.4
K-means	**Voc size**	*5*	67.2	65.7	58.6	57.9
		10	52.9	68.6	67.9	66.4
		15	60.7	51.4	62.9	67.9
		20	52.2	48.6	55	64.3
Energy-based			35.7	39.3	64.3	64.3

Table 5.3: Comparisons of classification accuracy (%) among different vocabulary calculation methods: energy-k-means, k-means and energy-based method in [28].

	Number of GPs			
	3	*6*	*10*	*20*
Voc size	608	602	639	641

Table 5.4: Vocabulary size calculated with energy-based method with different numbers of Gaussian processes.

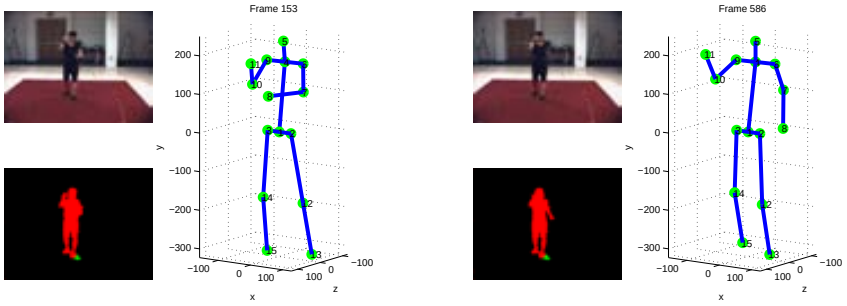
pose estimation errors between estimated *weak poses* $\hat{\mathbf{\Psi}}$ and the ground truth *weak poses* $\mathbf{\Psi}'$ as:

$$\varepsilon = \frac{1}{N} \sum_{p=1}^{P} \sum_{f=1}^{F_p} \|\hat{\psi} - \psi'\|^2, \tag{5.17}$$

where N is the total number of frames used for training, P is the total number of training performances and F_p is frame numbers of the p-th training performance. To discard missing human detection, we first calculate the energy of shape context descriptor for each training frame and filter the training sequences based on calculated energies by keeping 90% of the energies over all frames. This effectively eliminates frames containing catastrophic silhouette extraction failures. In our experiments, we evaluate different numbers of Gaussian processes (recall that we use one Gaussian process for each dimension in our *weak pose* space). From table 5.5, we observe that with fewer than 20 Gaussian processes, increasing the number of Gaussian processes results in noticeable increases in classification accuracy and also decreases in pose estimation error. Our explanation for this is: a small numbers of Gaussian processes are not able to capture or describe all the motion possibilities for actions, which results in predictions that are not accurate. After 20 Gaussian processes, increasing number of Gaussian processes does not result in notable increases in classification accuracy or decreases in pose estimation error. So the best trade-off between accuracy and model complexity is found with 20 Gaussian processes with a vocabulary size of 10. The

		Number of GPs						
		3	*6*	*10*	*15*	*20*	*25*	*30*
Voc size	*5*	73.9	86.8	86.3	86.0	86.1	86.1	85.6
	10	67.7	83.6	82.9	83.0	84.4	84.2	84.2
	15	64.1	83.9	82.6	80.8	85.4	83.9	83.7
	20	64.7	79.0	77.5	79.7	78.4	84.2	82.2
Mean Error		0.399	0.304	0.241	0.200	0.169	0.146	0.127

Table 5.5: Comparison of classification accuracy (%) and *weak pose* reconstruction error with different numbers of Gaussian processes and different vocabulary size. Reconstruction error is the difference between predicted *weak poses* and ground truth *weak poses*.



(a) (b)

Figure 5.7: Two example frames of good estimation of *weak poses* in HumanEva dataset. *Weak poses* are back-projected from *UaSpace* to the original parameter space and visualized as human poses.

subsequent experiments are computed with these optimal settings.

Weak pose reconstruction results

To visualize results of *weak pose* reconstruction, we project weak poses from *UaSpace* back to the original parameter space. Figs 5.7 and 5.8 show some examples of estimated *weak poses*. We can see that in fig. 5.7, pose estimation results are satisfactory. In fig. 5.8, there is a big difference between the estimation and the ground truth. But since our ultimate goal is action recognition and not pose estimation, we will not concentrate on further improvements on pose estimation. We show in following sections, that this pose estimation precision give a promising action recognition rate.

Authors in [3] also use a regression method to learn mappings between feature descriptors and 3D human poses. They use histogram of shape contexts as feature descriptors and estimate human poses by training relevance vector machines from histogram of shape contexts to each joint position of their human model. To address the

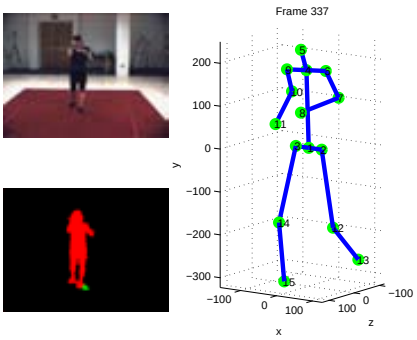


Figure 5.8: An example of bad estimation of a *weak pose* in HumanEva dataset.

Regression method	GP	RVM
Error	0.241	0.385

Table 5.6: The comparison of weak pose reconstruction errors between Gaussian process regression and relevance vector machine regression. Reconstruction error is the difference between predicted *weak poses* and ground truth *weak poses*. *Weak poses* is represented with direction cosine. The dimension of weak poses is 10.

problem of mapping ambiguities due to loss of depth information in video sequences or images, they embed pose estimation into a tracking framework. To compare with our method, we implemented a relevance vector machine for regression. We repeat the same procedure for weak pose estimation, but instead of Gaussian process regression, we use relevance vector machines in learning regressions between shape context descriptors and weak poses in *UaSpace*. Reconstruction errors for weak poses from shape context descriptors using relevance vector machine is shown in table 5.6.

Relevance vector machine regression, also known as a sparse Bayesian model, performs faster than Gaussian process regression in our experiment. Since the main idea of relevance vector machine is to select the most representative training data as relevant vectors. But the estimation error shows that relevance vector machine is not a suitable solution for our problem. Since reconstructing weak poses with dimensions of 3 using Gaussian process regressions has a similar error as reconstructing weak poses with dimensions of 10 using relevance vector machine, this indicates that estimated weak poses from relevance vector machines will not be able to provide priors good enough for further classification with the same dimensions.

Temporal step size

We also use cross validation to get optimal temporal step size d. We add Gaussian noise of different scales to the original 3D marker positions to test the robustness of the prosed method. We run each noise scale 5 times and calculate average accu-

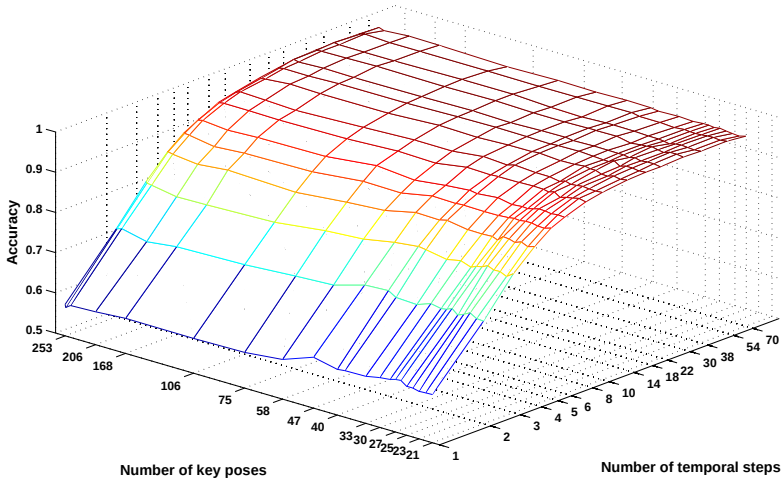


Figure 5.9: The relations between number of temporal steps, number of key poses and action recognition accuracy.

racy for all noise scales. Experiment results are shown in fig. 5.9. This figure shows relations between numbers of temporal steps, numbers of key poses and action recognition accuracies. From the figure, we can see that the size of temporal steps has more influences than the number of key poses (vocabulary size). And after the size of temporal steps reaches 13, classification accuracy becomes rather stable. This implies that the decisive factor in action recognition comes from the continuous motion. Motion elements of short duration is more representative for an action than the overall distribution of important poses. Later on, we fix temporal step size as 13 for the rest of our experiments.

Comparison with BoW

To verify the effect of the incorporation of *weak poses*. We repeat the experiment with the optimum parameter settings for traditional bag of words pipeline. We use energy-k-means for vocabulary selection and set vocabulary size of 10. Cost of support vector machine is as 5 and temporal step size is as 13. But instead of in *UaSpace*, vocabularies and histograms are calculated in 2D image feature space. Action recognition accuracy is from the traditional bag of word pipeline is 80.0%, while the action recognition accuracy for the proposed method is 84.4%.

5.4.3 Action recognition accuracy

We utilize a BoP model in classifying actions, as described in section 5.3. A set of Gaussian processes and a BoP model are trained on all training data including

Action	Performer	Trial	Test	Total frames
Walking	1, 2, 3	2	1104, 1256, 952	3312
Jog	1, 2, 3	2	944, 956, 936	2836
Gesture	1, 2, 3	2	1172, 1159, 686	3017
Throw/Catch	1, 2, 3	2	1012, 1475, 1072	3559
Box	1, 2, 3	2	739, 1104, 812	2655

Table 5.7: The composition of test data from Humaneva dataset. Test data are composed of the second trial from the three performers performing five different actions. We list frames numbers for all test sequences. Each number in "Test" column corresponds to one motion sequence. Total frames is the sum of all frames for one action.

Action	Performer	Trial	Test	Total frames
Walk	*Alba*, *Andreas*	1, 2, 3	234, 236, 116, 126, 136, 88	936
Wave	*Alba*, *Andreas*	1, 2, 3	68, 88, 51, 64, 54, 45	370
Throw a ball	*Alba*, *Andreas*	1, 2, 3	34, 40, 15, 10, 22, 28	149
Punch	*Alba*, *Andreas*	1, 2, 3	47, 59, 40, 47, 46, 33	272

Table 5.8: The composition of test data from IXMAS dataset. Test data are composed of two performers performing four different actions. We list frames numbers for all test sequences. Each number in "Test" column corresponds to one motion sequence. Total frames is the sum of all frames for one action.

training and validation data. With the trained models, we evaluate our method on the test data from both HumanEva and IXMAS datasets.

As we take the whole performance as one training example, we have an acute lack of training data. We address this problem by synthesizing training data like [14]. We first split training performances into sub-performances. Then, we translate sub-performances with $trans$ times the maximum difference of the training data, where

$$trans = \{-0.20, -0.15, -0.10, -0.05, 0.05, 0.10, 0.15, 0.20\}, \tag{5.18}$$

and scale sub-performances by

$$scale = \{0.80, 0.85, 0.90, 0.95, 1.05, 1.10, 1.15, 1.20\}. \tag{5.19}$$

We also split and translate test performances into sub-performances. The procedure is the same as for training date. Experimental results for HumanEva dataset are shown in table 5.9. The method from [51] shows upper bound accuracy for initialized latent pose conditional random field model ($LPCRF_{init}$ in [51]) with the same training and test data.

In our experiments, normalization of input data is a very important step for Gaussian process regression to make good predictions. So we experimented with two different ways of normalizing data: standard-deviation based and individual normalizations. Our method with individual normalization has better average classification accuracy than the approach presented in [51].

Acc.	Box	Jog	Gest	Walk	T/C	All - T/C	All + T/C
[51]	98.9	99.0	63.7	99.6	*no*	90.3	*no*
Std-norm	88.4	75.1	87.6	91.0	80.0	85.5	84.4
Ind-norm	97.1	91.8	91.9	94.6	80.0	**93.9**	**91.1**

Table 5.9: Comparison of action recognition accuracy (%) in HumanEva between our methods and the method presented in [51]. Classification accuracy is defined as correctly labeled samples over total number of samples(refer to equation 5.16). "Std-norm" and "Ind-norm" refer to standard deviation normalizing method and individually normalizing method (refer to section 5.2). The column "$All - T/C$" shows the average classification accuracy for all actions excluding "Throw/Catch" and and the column "$All + T/C$" including "Throw/Catch".

Due to illumination changes and errors from background subtraction, human silhouettes from every image frame have variant qualities. As a result, the total pixel numbers vary from one frame to another. Individually normalizing method eliminates these differences. So that, later histograms are computed on the same basis. On the contrary, standard deviation based normalization are more suitable to cases while different dimensions from image features have different range of variations. In this case, different dimensions are separately normalized. In later experiments, we fix our normalization as individual normalization.

From experimental results, we observe that for "Throw/Catch" action, in both normalization strategies, classification accuracy are not as satisfactory as other actions. One possible reason for this is the limited number of training samples for this action. We are using PCA in reducing representation dimensionality. In this case, if training examples for an action are too few, the variations of this action would not be able to be captured by the main eigenvectors. As a result, action recognition accuracy is not as good as other classes. Another observation is, for "Jog" and "Box", individual normalization has a much better performance than the standard-deviation based one. Our explanation for this is, "Jog" and "Box" have more variate poses compared with "Gesture" (the lower body parts of the performer are relatively stable), "Throw/Catch" (the lower body parts are also relatively stable) and "Walking" (the movements of body parts are not as fierce as in "Jog" and "Box"). As a result, when we normalize all training data together, these action classes are more likely to be influenced. While individual normalization keeps variate information of the SCD from each image frame.

In certain cases, the action recognition results are not comparable to state of art results. Our analysis is this is due to different qualities of subtracted silhouettes. Accordingly, we modify the way of normalizing shape context data. For each shape context descriptor, we normalize it with the total number of this shape context descriptor. In this way, we can get rid of quality difference between training and test data. Experimental results show that this normalization method is better than the first method and can compare with state of art method.

From the table, we can observe with the first performer, in which silhouette extraction has the best quality, classification accuracy is the highest. With the quality of

extracted silhouettes decreases, classification accuracy also decreases sharply. Testing videos might not be captured in the same condition as training video sequences, and also parameter settings in background extraction of testing videos might be different from those of training. These two factors results in differences of silhouettes extracted from testing videos with silhouettes extracted from training videos. Even these two factors are the same, in certain illumination condition or due to camouflage, the quality of extracted silhouettes are not good. Thus we conclude that accuracy of action recognition strongly depends on robustness of extracted silhouettes. So in order to guarantee our method works robustly, we need to guarantee high quality of extracted silhouettes from videos. Secondly, we need to guarantee the quality of silhouettes from the test video sequences should be compatible with the quality of silhouettes extracted from the training video sequences.

We run the experiments on a personal computer with a $3.19Hz$ processor, and $12GB$ memory. The time cost for training one Gaussian process is *hours*, and predicting one dimension is *minutes*. And the time cost for calculating the vocabulary, calculating histograms and classification is *minutes*.

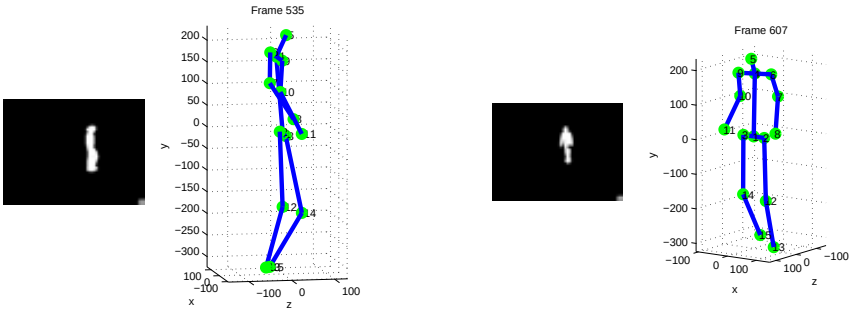
Accuracy	Punch	Wave	Throw a ball	Walk	All actions
Ind-normal	75.0	79.2	75	87.5	79.2
[26]	86.8	79.9	82.4	79.7	82.2

Table 5.10: Action recognition accuracy (%) of our individually normalizing method for IXMAS dataset using the models learnt from HumanEva dataset compared with the method prosed in [26].

We further test our action model (trained using HumanEva data) on IXMAS dataset and experimental results is shown in table 5.10. We compare our results with method in [26]. Note that camera settings in HumanEva dataset and IXMAS dataset are slightly different. This results in slight difference between human silhouettes from these two dataset. Also although we have four corresponding actions, they are not exactly the same action. But all corresponding actions in IXMAS dataset are subsets from HumanEva dataset. For example, "Gesture" action in HumanEva dataset semantically contains "Wave" and "Come".

Despite the differences between these two datasets, our models trained on HumanEva dataset obtain a relatively close result as method in [26]. We even achieve better results with action "Walk". One explanation is that test data in "Walk" have more frames than other actions in IXMAS dataset, and our holistic method performs better with more frames. Another reason might be, "Walk" is a comparatively repetitive action that does not have as much variance as other actions when performed by a different human. While for other action, this is not the case. For example, for "Box" in HumanEva dataset, performer "$S1$" does not move his legs while performer "$S2$" jumps forward and backwards during the performances.

In figure 5.10 and figure 5.11, we show sampled reconstruction of *weak poses*. We can see that in the condition of similar camera viewpoint and similar silhouette shapes, like in figure 5.10, reconstructed poses can be very precise. While the differences between HumanEva dataset and IXMAS dataset, for example, different ways of actors



(a) (b)

Figure 5.10: Two exampled frames of good estimations of *weak poses* in IXMAS dataset. *Weak poses* are back-projected from *UaSpace* to the original parameter space and visualized as human poses.

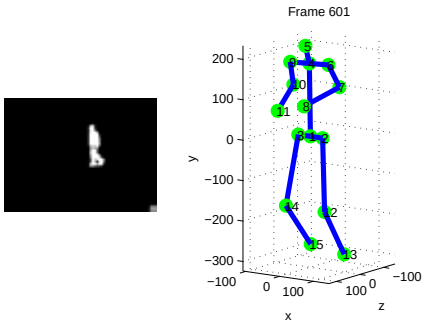


Figure 5.11: An example frame of bad estimation of a *weak pose* in IXMAS dataset.

performing the same actions, might cause some false prediction, like in figure 5.11.

5.5 Conclusions and future work

In this chapter we have proposed a novel approach to action recognition using a BOP model with *weak poses* estimated from silhouettes. We have applied GPR to model the mapping from silhouettes to *weak poses*. We modify the classic BOW pipeline by incorporating temporal information. We train our models with the HumanEva dataset and test it with test data from HumanEva and IXMAS datasets. Experimental results show that our method performs effectively for the estimation of *weak poses* and action recognition. Even though different datasets have different camera setting and

different perception about performing actions, our method is robust enough to obtain satisfactory results. Note that although the proposed method is not view-invariant, it is straightforward to extend to multiple view solution by including training data from all viewpoints. In prediction phase, viewpoint will be naturally selected in the regression procedure.

In future work, it would be interesting to explore how 3D motion data would benefit local image features which incorporate temporal information, like dense trajectories in [81]. The state-of-art work on action recognition from videos usually incorporate temporal information in feature descriptor or in motion models, although we enhance our method by considering feature of a temporal window, it would also be interesting to explore the effect of 3D motion data on local image descriptor which incorporate temporal information and compare their performances.

Chapter 6

Conclusion and future work

In this book, we explore the effect of 3D motion data in 3D pose estimation and action recognition problems. We investigate several important factors in pose estimation and action recognition. First, we compare two main school of approaches of estimating 3D poses from 2D body part positions: learning method and modeling method. Based on validations on public data set including HumanEva data set and TUM-kitchen data set, we conclude that when training set is variant and close to test set, learning method tends to outperform modeling method while modeling method provides a mathematical formulation for the reconstruction problem and is more resistant to differences between training and test set. Considering that our problems are most in confined environment, we choose the learning method as the mapping method for all later experiments. Later on, we extend the method by adding a module of 2D body part detector and propose a solution to estimate 3D poses from still images with cluttered backgrounds. We validate this method on public data set and compare with a baseline method with shape context as input feature. Experiment results show that our proposed solution is outperforming the baseline method.

Second, we resort to input features to accurately estimate 3D poses taking into consideration that most feature descriptors and distance measures take input features as a while, while certain channels of input features might contain noise or inaccurate information. In this solution, we propose a feature based on iterative close point algorithm which adapts to noise and discard unwanted noise according to overall evaluation of the quality of the input features, silhouettes in our case. We compare the proposed feature with standard PHOG feature and shows that the proposed feature outperforms the other in many cases.

Third, we extend the application of 3D motion data in action recognition. By explicitly encoding a module of pose estimation, we take advantage of 3D motion data in action recognition. Considering the unambiguous representation of 3D poses compared with 2D poses, we hypothesize that incorporating 3D motion data helps to enhance action recognition accuracies. We utilize a dimension-decreased representation of human poses, *weak poses* as the target space of mapping model and also the space where we recognize actions with bag of poses model. By validating the pipeline on HumanEva data set and IXMAS data set, our method outperform a related work

in average recognition accuracy. Whats more, compared with the framework where no motion data is incorporated, the recognition accuracy is 4.4% higher.

We conclude that 3D motion data provide important information for unambiguous pose estimation and comparatively accurate action recognition. It is useful in its original representation or in dimension decrease representation. As we can see, for action recognition, compared with the method that don't incorporate 3D motion data, there is a improvement in the performance. What's more, as the target space for pose estimation, it provide more accurate representation compared with 2D pose representation. And promising method is available for 3D pose estimation from 2D image features, even with cluttered backgrounds.

There are several possible ways to continue and extend the current work.

1. First, aiming at solving double counting problem in 2D body part detection with mixture of parts model (proposed in [89]), we can enhance 2D body part localization with optimization. With more accurate 2D body part localization, 3D pose estimation accuracies would also be able to improve. To deal with this problems, authors in [83] propose multiple tree models. The models contain a tree structure to account for kinematic constraints between connected body parts, tree structures for spatial constraints among body parts without direct connections, and tree structures for occluded body parts. Different tree structures are combined with a boosting procedure. Other research also explore the possibility of imposing constraints in the optimization target. For example, authors in [88] modify the optimization target and incorporate spatial constraints to deal with double counting problem. In referencing, those poses who violate the spatial constraints would end up with a lower score. In our future work, we are interested in exploring combining multiple feature cues for enhancing 2D body part detection accuracies.

2. Second, we could enhance 3D pose estimation accuracies from detected 2D body parts by composing physical constraints to the human model or by incorporating temporal cues. Due to lack of depth information in 2D images, 3D pose estimation from 2D still images is a n to n mapping problems, while we can boost the accuracies of mapping models with mixture models and enriching training samples, it is also interesting to know the effect of imposing physical constraints into human models. 2D body part detections are mostly considered effective when different body parts are localized, but by boosting to 3D poses, we need more information other than the precious 2D body part position. For example, it is important that the left side and the right side of the body are recognized correctly, which an essential factor to boost 2D body part positions to 3D poses.

3. Third, it would be interesting to explore how 3d motion data works for local features in action recognition problem instead of global features as in our work. As most of local image features exploited for state-of-art action recognition, for example dense trajectories in [81] incorporate temporal information in local feature, which is effective in capturing the attributions for actions.

4. Fourth, we would like to explore the application of 3D motion data in human

tracking problem. With the addition of 3D motion data, we would like to explore the possible enhancement in tracking, which would help us in automatic surveillance or house care for aged people.

Appendix A

Publications

Refereed journals

- Wenjuan Gong, F. Xavier Roca, Jordi Gonzàlez. 3D Motion Priors for Action Recognition from Video Sequences *EURASIP-Signal Processing*, In press.
- Wenjuan Gong, Preben Fihl, Xu Hu, Jordi Gonzàlez, Thomas B. Moeslund, Robustness of Input Features from Noisy Silhouettes in Human Pose Estimation. *IET-Computer Vision*. Submitted.

Refereed major conferences

- Wenjuan Gong, Jordi Gonzàlez, João Manuel R. S. Tavares, and F. Xavier Roca. A New Dataset on Human Action Interaction. AMDO 2012.
- Adela Bărbulescu, Wenjuan Gong, Jordi Gonzàlez, Thomas B. Moeslund. 3D Human Pose Estimation using 2D Body Part Detectors, ICPR 2012.
- Wenjuan Gong, Jürgen Brauer, Michael Arens, Jordi Gonzàlez. Modeling vs. Learning Approaches for Monocular 3D Human Pose Estimation. In 1st IEEE International Workshop on Performance Evaluation on Recognition of Human Actions and Pose Estimation Methods, in conjunction with ICCV, 2011.
- Jürgen Brauer, Wenjuan Gong, Jordi Gonzàlez, Michael Arens. On the Effect of Temporal Information on Monocular 3D Human Pose Estimation. In 2nd IEEE International Workshop on Analysis and Retrieval of Tracked Events and Motion in Imagery Streams (ARTEMIS2011), in conjunction with ICCV, 2011.
- Nataliya Shapovalova, Wenjuan Gong, Marco Pedersoli, F. Xavier Roca and Jordi Gonzàlez, On Importance of Interactions and Context in Human Action Recognition. In ibPRIA 2011.

- Wenjuan Gong, Andrew D. Bagdanov, F. Xavier Roca, Jordi Gonzàlez. Automatic Key Pose Selection for 3D Human Action Recognition. AMDO 2010.

Bibliography

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

[2] Ankur Agarwal and Bill Triggs. A local basis representation for estimating human pose from cluttered images. In *Proc. of Asian Conf. on Computer Vision*, pages 50–59, 2006.

[3] Ankur Agarwal and Bill Triggs. Recovering 3d human pose from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:44–58, 2006.

[4] M. Ahmad and S.W. Lee. Hmm-based human action recognition using multiview image sequences. In *ICPR*, pages 263–266, 2006.

[5] S. Ali and M. Shah. Human action recognition in videos using kinematic features and multiple instance learning. *PAMI*, 32:288–303, 2010.

[6] A. Amato, M. Mozerov, A.D. Bagdanov, and J. Gonzàlez. Accurate moving cast shadow suppression based on local color constancy detection. *TIP*, 20:2954–2966, 2011.

[7] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Monocular 3d pose estimation and tracking by detection. In *Proc. of CVPR 2010*, USA, 2010.

[8] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):509–522, 2002.

[9] Serge Belongie and Jitendra Malik. Matching with shape contexts. *Content-Based Access of Image and Video Libraries, IEEE Workshop on*, 0:20, 2000.

[10] M. Bertozzi, A. Broggi, M. Del Rose, M. Felisa, A. Rakotomamonjy, and F. Suard. A pedestrian detector using histograms of oriented gradients and a support vector machine classifier. In *Intelligent Transportation Systems Conference*, pages 143–148, 2007.

[11] Liefeng Bo and C. Sminchisescu. Structured output-associative regression. In *CVPR*, pages 2403–2410, june 2009.

[12] Liefeng Bo and C. Sminchisescu. Modeling and forecasting stock market volatility by gaussian processes based on garch, egarch and gjr models. In *WCE*, july 2011.

[13] Edwin Bonilla, Kian Ming Chai, and Chris Williams. Multi-task gaussian process prediction. In *NIPS*, pages 153–160. 2008.

[14] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. In *ICCV*, pages 1–8, 2007.

[15] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *CIVR*, pages 401–408, 2007.

[16] Anna Bosch, Andrew Zisserman, and Xavier Munoz. Representing shape with a spatial pyramid kernel. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, CIVR '07, pages 401–408, New York, NY, USA, 2007. ACM.

[17] M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. In *CVPR*, pages 994–999, 1997.

[18] Jürgen Brauer and Michael Arens. Reconstructing the missing dimension: From 2d to 3d human pose estimation. In *Proc. of REACTS workshop, in conj. with Int. Conf. of Computer Analysis of Images and Patterns (CAIP2011)*, Spain, Málaga, 2011. to appear in.

[19] Jürgen Brauer, Wenjuan Gong, Jordi Gonzàlez, and Michael Arens. On the effect of temporal information on monocular 3d human pose estimation. In *ICCV Workshops*, pages 906–913, 2011.

[20] C.K.I. Williams C.E. Rasmussen. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[21] K.M. Chai, C. Williams, S. Klanke, and S. Vijayakumar. Multi-task gaussian process learning of robot inverse dynamics. In *NIPS*, 2008.

[22] C.C. Chen and J.K. Aggarwal. Recognizing human action from a far field of view. In *IEEE Workshop on Motion and Video Computing*, 2009.

[23] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR (1)*, pages 886–893, 2005.

[24] J.W. Davis and A.F. Bobick. The representation and recognition of human movement using temporal templates. In *CVPR*, pages 928–934, 1997.

[25] Carl Henrik Ek, Philip H. S. Torr, and Neil D. Lawrence. Gaussian process latent variable models for human pose estimation. In *MLMI*, pages 132–143, 2007.

[26] R. Nevatia F. Lv. Single view human action recognition using key pose matching and viterbi path searching. In *CVPR*, pages 1–8, 2007.

[27] X. Feng and P. Perona. Human action recognition by sequence of movelet codeword. In *International Symposium on 3D Data Processing Visualization and Transmission*, pages 717–721, 2002.

[28] W. Gong, A.D. Bagdanov, J. Gonzàlez, and F.X. Roca. Automatic key pose selection for 3d human action recognition. In *AMDO*, 2010.

[29] Wenjuan Gong, Jordi Gonzàlez, João Manuel R. S. Tavares, and F. Xavier Roca. A new image dataset on human interactions. In *AMDO*, pages 204–209, 2012.

[30] J. Gonzàlez, D. Rowe, J. Varona, and F.X. Roca. Understanding dynamic scenes based on human sequence evaluation. *Image and Vision Computing*, 27:1433–1444, 2009.

[31] K. Grauman and T. Darrell. The pyramid match kernel: discriminative classification with sets of image features. In *ICCV*, pages 1458–1465, 2005.

[32] Gregor Gregorčič and Gordon Lightbody. Gaussian process approach for modelling of nonlinear systems. *Engineering Applications of Artificial Intelligence*, 22(4-5):522–533, 2009.

[33] J. Gu, X. Ding, S. Wang, and Y. Wu. Action and gait recognition from recovered 3-d human joints. *IEEE Transactions on Systems, Man, and Cybernetics Part B*, 40:1021–1033, 2010.

[34] Hao Jiang. 3d human pose reconstruction using millions of exemplars. In *Proc. of 20th ICPR*, pages 1674–1677, 2010.

[35] Ashish Kapoor, Kristen Grauman, Raquel Urtasun, and Trevor Darrell. Gaussian processes for object categorization. *International Journal of Computer Vision*, 88(2):169–188, 2010.

[36] K. Kulkarni, E. Boyer, R. Horaud, and A. Kale. An unsupervised framework for action recognition using actemes. In *ACCV*, pages 592–605, 2011.

[37] I. Laptev and T. Lindeberg. Space-time interest points. In *ICCV*, pages 432–439, 2003.

[38] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, pages 2169–2178, 2006.

[39] F.F. Li and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, pages 524–531, 2005.

[40] David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV '99, pages 1150–, Washington, DC, USA, 1999. IEEE Computer Society.

[41] F. Lv and R. Nevatia. Recognition and segmentation of 3d human action using hmm and multi-class adaboost. In *ECCV*, pages 359–372, 2006.

[42] F. Lv, R. Nevatia, and M.W. Lee. 3d human action recognition using spatio-temporal motion templates. In *ICCV Workshop on Human-Computer Interaction*, pages 120–130, 2005.

[43] Margaret A. McDowell, Cheryl D. Fryar, Cynthia L. Ogden, and Katherine M. Flegal. Anthropometric reference data for children and adults: United states, 2003?-2006. *National Health Statistics Reports*, (10):1–45, 2008.

[44] A. Melkumyan and F. Ramos. Multi-kernel gaussian processes. In *IJCAI*, pages 1408–1413, 2011.

[45] Arman Melkumyan and Eric Nettleton. An observation angle dependent nonstationary covariance function for gaussian process regression. In *ICONIP*, pages 331–339, 2009.

[46] G. Mori and J. Malik. Recovering 3d human body configurations using shape contexts. *PAMI*, 28:1052–1062, 2006.

[47] Greg Mori and Jitendra Malik. Estimating human body configurations using shape context matching. In *Proceedings of the 7th European Conference on Computer Vision-Part III*, ECCV '02, pages 666–680, London, UK, UK, 2002. Springer-Verlag.

[48] Greg Mori and Jitendra Malik. Recovering 3d human body configurations using shape contexts. *PAMI*, 28(7):1052–1062, 2006.

[49] Jürgen Müller and Michael Arens. Human pose estimation with implicit shape models. In *Proc. of the first ACM international workshop on Analysis and Retrieval of Tracked Events and Motion in Imagery Streams*, ARTEMIS '10, pages 9–14, New York, NY, USA, 2010. ACM.

[50] I. Nazli, G. C. Ramazan, P. Selen, and D. Pinar. Recognizing actions from still images. In *ICPR*, pages 1–4, 2008.

[51] H Ning, W Xu, Y Gong, and T Huang. Latent pose estimator for continuous action recognitionmodeling 3d human poses from uncalibrated monocular images. In *ECCV*, pages 419–433, 2008.

[52] V. Parameswaran and R. Chellappa. View independent human body pose estimation from a single perspective image. In *In Proc. of CVPR 2004*, volume 2, pages II–16 – II–22 Vol.2, june-2 july 2004.

[53] Marco Pedersoli, Andrea Vedaldi, and Jordi Gonzàlez. A coarse-to-fine approach for fast deformable object detection. In *CVPR*, pages 1353–1360, 2011.

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

[55] R. Poppe and M. Poel. Comparison of silhouette shape descriptors for example-based human pose recovery. In *Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition*, pages 541–546, 2006.

[56] M. Raptis, K. Wnuk, and S. Soatto. Flexible dictionaries for action classification. In *Proceedings of the Workshop on Machine Learning for Visual Motion Analysis*, 2008.

[57] Carl Edward Rasmussen. Gaussian processes for machine learning. MIT Press, 2006.

[58] Bernhard Riemann. On the hypotheses on which geometry is based (ueber die hypothesen, welche der geometrie zu grunde liegen), 1867.

[59] Ignasi Rius, Jordi Gonzàlez, Javier Varona, and F. Xavier Roca. Action-specific motion prior for efficient bayesian 3d human body tracking. *Pattern Recognition*, 42(11):2907–2921, 2009.

[60] O. Rudovic and M. Pantic. Shape-constrained gaussian process regression for facial-point-based head-pose normalization. In *ICCV*, pages 1495–1502, November 2011.

[61] M.S. Ryoo and J.K. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *ICCV*, 2009.

[62] Andres Sanin, Conrad Sanderson, Mehrtash T. Harandi, and Brian C. Lovell. K-tangent spaces on riemannian manifolds for improved pedestrian detection. In *ICIP*, 2012.

[63] C. Schüldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *ICPR*, pages 32–36, 2004.

[64] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th international conference on Multimedia*, pages 357–360, 2007.

[65] Leonid Sigal, Alexandru O. Balan, and Michael J. Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *IJCV*, 87:1–24, 2010.

[66] Leonid Sigal and Michael J. Black. Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In *CVPR*, volume 2, pages 2041–2048, 2006.

[67] P. Siva and T. Xiang. Action detection in crowd. In *BMVC*, pages 9.1–9.11, 2010.

[68] Brahim-Belhouari Sofiane and Amine Bermak. Gaussian process for nonstationary time series prediction. *Computational Statistics and Data Analysis*, 47(4):705–712, 2004.

[69] R. Souvenir and J. Babbs. Viewpoint manifolds for action recognition. In *CVPR*, pages 1–7, 2008.

[70] F. Suard, A. Rakotomamonjy, A. Bensrhair, and A. Broggi. Pedestrian detection using infrared images and histograms of oriented gradients. In *Intelligent Vehicles Symposium, IEEE*, pages 206–212, 2006.

[71] Camillo J. Taylor. Reconstruction of articulated objects from point correspondences in a single uncalibrated image. *Computer Vision and Image Understanding*, 80:349–363, 2000.

[72] Moritz Tenorth, Jan Bandouch, and Michael Beetz. The TUM Kitchen Data Set of Everyday Manipulation Activities for Motion Tracking and Action Recognition. In *THEMIS workshop. In conj. with ICCV 2009*.

[73] Michalis K. Titsias and Neil D. Lawrence. Bayesian gaussian process latent variable model. *Journal of Machine Learning Research - Proceedings Track*, 9:844–851, 2010.

[74] Oncel Tuzel, Fatih Porikli, and Peter Meer. Pedestrian detection via classification on riemannian manifolds. *PAMI*, 30:1–15, 2008.

[75] R. Urtasun and T. Darrell. Sparse probabilistic regression for activityindependent human pose inference. In *CVPR*, pages 1–8, 2008.

[76] Raquel Urtasun, David J. Fleet, and Pascal Fua. 3D people tracking with gaussian process dynamical models. In *CVPR*, pages 238–245, 2006.

[77] Raquel Urtasun, David J. Fleet, Andreas Geiger, Jovan Popovic, Trevor Darrell, and Neil D. Lawrence. Topologically-constrained latent variable models. In *ICML*, pages 1080–1087, 2008.

[78] Raquel Urtasun, David J. Fleet, Aaron Hertzmann, and Pascal Fua. Priors for people tracking from small training sets. *IEEE Int. Conf. on Computer Vision*, 1:403–410, 2005.

[79] S.N. Vitaladevuni, V. Kellokumpu, and L.S. Davis. Action recognition using ballistic dynamics. In *CVPR*, pages 1–8, 2008.

[80] C. Wallraven, B. Caputo, and A.B.A. Graf. Recognition with local features: the kernel recipe. In *ICCV*, pages 257–264, 2003.

[81] Heng Wang, Alexander Kläser, Cordelia Schmid, and Liu Cheng-Lin. Action Recognition by Dense Trajectories. In *CVPR*, pages 3169–3176, Colorado Springs, United States, Jun 2011.

[82] Jack M. Wang, David J. Fleet, and Aaron Hertzmann. Gaussian process dynamical models for human motion. *PAMI*, 30(2):283–298, 2008.

[83] Yang Wang and Greg Mori. Multiple tree models for occlusion and spatial constraints in human pose estimation. In *Proceedings of the 10th European Conference on Computer Vision: Part III*, pages 710–724, Berlin, Heidelberg, 2008. Springer-Verlag.

[84] Xiaolin K. Wei and Jinxiang Chai. Modeling 3d human poses from uncalibrated monocular images. In *IEEE 12th Intern. Conf. on Computer Vision*, pages 1873–1880, October 2009.

[85] D. Weinland, E. Boyer, and R. Ronfard. Action recognition from arbitrary views using 3d exemplars. In *ICCV*, pages 1–7, 2007.

[86] D. Weinland, R. Ronfard, and E. Boyer. Motion history volumes for free viewpoint action recognition. In *ICCV PHI workshop*, 2005.

[87] D. Weinland, R. Ronfard, and E. Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 115:224–241, 2011.

[88] Yi Xiao, Huchuan Lu, and Shifeng Li. Posterior constraints for double-counting problem in clustered pose estimation. In *IEEE International Conference on Image Processing*, 2012.

[89] Yi Yang and Deva Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, pages 1385–1392, 2011.

[90] Yi Yang and Deva Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, pages 1385–1392, 2011.

[91] V.M. Zatsiorsky. *Kinematics of Human Motion*. Human Kinetics Publishers, 1998.

[92] V.M. Zatsiorsky. *Kinetics of Human Motion*. Human Kinetics Publishers, 2002.

[93] Zhengyou Zhang. Iterative point matching for registration of free-form curves and surfaces. *Int. J. Comput. Vision*, 13(2):119–152, 1994.

[94] Xu Zhao, Yun Fu, and Yuncai Liu. Temporal-spatial local gaussian process experts for human pose estimation. In *ACCV*, pages 364–373, 2009.

[95] J. Zhu, S. Hoi, and M. Lyu. Nonrigid shape recovery by gaussian process regression. In *CVPR*, pages 1319–1326, 2009.

[96] M. Zobl, F. Wallhoff, and G. Rigoll. Action recognition in meeting scenarios using global motion features. In *Proceedings Fourth IEEE International Workshop on Preformance Evaluation of Tracking and Surveillance*, pages 32–36, 2003.