TECHNISCHE UNIVERSITÄT
BERGAKADEMIE FREIBERG
Die Ressourcenuniversität. Seit 1765.

Learning Continuous Human-Robot Interactions from Human-Human Demonstrations

Von der Fakultät für Mathematik und Informatik
der Technischen Universität Bergakademie Freiberg

genehmigte

DISSERTATION

zur Erlangung des akademischen Grades

Doktor-Ingenieur

(Dr.-Ing.)

vorgelegt von **M.Sc. David Dominic Vogt**

geboren am 10. August 1986 in Plauen

Gutachter: Prof. Dr.-Ing. habil. Bernhard Jung, Freiberg, Deutschland
Prof. Dr.-Ing. Heni Ben Amor, Tempe, USA

Tag der Verleihung: 6. Februar 2018

Versicherung

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Bei der Auswahl und Auswertung des Materials sowie bei der Herstellung des Manuskripts habe ich Unterstützungsleistungen von folgenden Personen erhalten:

Die Betreuer der vorliegenden Dissertation, Bernhard Jung und Heni Ben Amor, gaben Hinweise für die Überarbeitung früher Versionen des Manuskripts. Teile des dargestellten Materials entstanden im Rahmen von mir betreuter studentischer Qualifizierungsarbeiten von Steve Grehl, Ben Lorenz und Simon Stepputtis.

Weitere Personen waren an der Abfassung der vorliegenden Arbeit nicht beteiligt. Die Hilfe eines Promotionsberaters habe ich nicht in Anspruch genommen. Keine Personen haben von mir geldwertige Leistungen für Arbeiten erhalten. Die Arbeit wurde bisher weder im Inland noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt.

Acknowledgement

I would like to express my sincere gratitude to my advisers Prof. Jung and Prof. Ben Amor. Their extraordinary support, expert advice and continuous encouragement, guided me all the time and I could not have imaged having better mentors. In the end, their maddening attention to detail drove me to finally appreciate composition in graphics and appreciate color schemes.

I would also like to thank my colleagues and friends Erik Berger, Steve Grehl, Ben Lorenz and Simon Stepputtis for their wonderful collaboration. They supported me greatly and were always willing to try out new ideas in the lab. I am very thankful for all the stimulating discussions and our sleepless but productive nights before deadlines.

Thanks to the financial support of the German Academic Exchange Service, I had the pleasure to work with colleagues and labmates from CIDSE at ASU. These times are very much appreciated.

This thesis would have been impossible without the support of my wonderful wife Doreen who had to renounce much for my research. Last but not least, I would like to thank my family: my parents, grand-parents, parents-in-law and kids for supporting me spiritually throughout writing this thesis and my life in general.

Contents

Previous Publications

This thesis interpolates material from the authors own publications:

[1] David Vogt, Erik Berger, Heni Ben Amor, and Bernhard Jung. Learning Two-Person Interaction Models for Responsive Virtual Characters and Humanoid Robots. In *Fachgruppe Virtuelle und Augmentierte Realität der Gesellschaft für Informatik*, 2012

[2] David Vogt, Erik Berger, Heni Ben Amor, and Bernhard Jung. A Task-Space Two-Person Interaction Model for Human-Robot Interaction. In *Fachgruppe Virtuelle und Augmentierte Realität der Gesellschaft für Informatik*, 2013

[3] David Vogt, Heni Ben Amor, Erik Berger, and Bernhard Jung. Learning Two-Person Interaction Models for Responsive Synthetic Humanoids. *Journal of Virtual Reality and Broadcasting*, 11, 2014

[4] David Vogt, Steve Grehl, Erik Berger, Heni Ben Amor, and Bernhard Jung. A Data-Driven Method for Real-Time Character Animation in Human-Agent Interaction. In Timothy Bickmore, Stacy Marsella, and Candace Sidner, editors, *Intelligent Virtual Agents SE - 57*, volume 8637 of *Lecture Notes in Computer Science*, pages 463–476. Springer International Publishing, 2014

[5] David Vogt, Ben Lorenz, Steve Grehl, and Bernhard Jung. Behavior generation for interactive virtual humans using context-dependent interaction meshes and automated constraint extraction. *Computer Animation and Virtual Worlds*, 26(3-4):227–235, may 2015

[6] David Vogt, Simon Stepputtis, Richard Weinhold, Bernhard Jung, and Heni Ben Amor. Learning human-robot interactions from human-human demonstrations (with applications in Lego rocket assembly). In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 142–143. IEEE, nov 2016

[7] David Vogt, Simon Stepputtis, Steve Grehl, Bernhard Jung, and Ben Amor Heni. A System for Learning Continuous Human-Robot Interactions from Human-Human Demonstrations. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2882–2889, Singapore, 2017. IEEE

[8] David Vogt, Simon Stepputtis, Bernhard Jung, and Ben Amor Heni. One-Shot Learning of Human-Robot Handovers with Triadic Interaction Meshes. *Springer Special Issue: Learning for Human-Robot Collaboration*, 2018

At the core of this thesis, chapter 4 proposes a new imitation learning framework for collaborative robots. It is based on material from the authors own peer-reviewed articles [3, 5, 6]. Chapter 5 meanwhile rests on references [3] and [5]. The applications presented in chapter 6 previously appeared in the following publications of the author [6, 7]. At the time of writing, the results in chapter 7 are submitted for peer-review in [8].

Nomenclature

Boldface $\mathbf{x}$ represents a vector, the scalar x_i is the ith element of $\mathbf{x}$. $\mathbf{X}$ denotes a matrix and $\mathbf{X}^{i,j}$ indicates ith row and the jth column. Capital calligraphic letters such as $\mathcal{X}$ and $\mathcal{Y}$ represent sets. Unbolded capital letters X are scalars.

$(\cdot)^{oa/ca/po}$ denotes matrices, indices or scalars specific to the *observed* or *controlled* agent and the *passive object* respectively. $\mathbf{p}^{oa,i}$ for example means the i-th joint of the observed agent. $(\cdot)_{i:j}$ represents a range from i to j of the elements in $(\cdot)$.

$(\cdot)_t$ refers to the specific instance of $(\cdot)$ at time step t.

Symbol	Description
T	Amount of motion capture frames
K	Amount of segments
S	Amount of previous poses for temporal alignment
H	Index to latest user pose
I	Number of interactions/subtasks
M, N, U	Number of joints of the observed, controlled agent and the object
Z	Dimensionality of the pose optimization formulation
R	Number of kernels of the Gaussian Mixture Model
P	Number of tetrahedrons in an Interaction Mesh
$\mathbf{p}$	Point/Posture/Set of motion capture markers
$\mathbf{p}'$	Posture in Laplacian coordinates, i.e. $L(\mathbf{p})$
$\hat{\mathbf{p}}$	Live posture Point/Posture/Set of motion capture markers
$\mathcal{R}$	Cartesian space of the motion capture system
$\mathcal{G}$	Global posture space
$\mathcal{L}_m$	mth local posture space
$\mathcal{Q}_m$	mth segment, i.e. a set of points/poses $\mathbf{p}_{r:v}$), $\mathcal{Q} \in \mathcal{L}$
d_m	Euclidean distance of a pose $\mathbf{p}^o$ to segment $\mathcal{Q}_m$
$\mathbf{E}$	Distance matrix of past user poses to segments
$\mathbf{M}$	Mesh Laplacian
$\mathbf{C}$	Hard constraint matrix
$\mathcal{C}$	Index set of hard constraints
$\mathbf{F}$	Soft constraint matrix
$\mathbf{W}$	Soft constraint weight matrix
$\mathbf{D}$	Dynamic Time Warp cost matrix
Θ	A Hidden Markov Model
$\mathcal{S}$	Set of internal states
P_i	Start probabilities
$P_{i \to j}$	Transition probability matrix
$p_i(o)$	Probability of observing symbol o in state s_i, $s_i \in \mathcal{S}$

Acronyms

ANN Artificial Neural Net

c-HMM continuous Hidden Markov Model

DMP Dynamic Movement Primitive

DOF Degree of Freedom

DR Dimensionality Reduction

DTW Dynamic Time Warp

EM Expectation Maximisation

g-HMM Gaussian Hidden Markov Model

GMM Gaussian Mixture Model

GMR Gaussian Mixture Regression

GPS Global Posture Space

HAI Human-Agent Interaction

HMD Head Mounted Display

HMM Hidden Markov Model

d-HMM discrete Hidden Markov Model

HRI Human-Robot Interaction

IK Inverse Kinematics

IM Interaction Mesh

KDE Kernel Density Estimation

k-**d tree** k-dimensional tree

LME Laplacian Mesh Editing

LPS Local Posture Space

MDP Markov Decision Process

PC Principal Component

PCA Principal Component Analysis

pm-HMM Path Map Hidden Markov Model

PPCA Probabilistic Principal Component Analysis

sc-HMM semi-continuous Hidden Markov Model

SLE System of Linear Equations

TCP Tool Center Point

TPN Time Petri Net

TRL Technology Readiness Level

VR Virtual Reality

List of Figures

List of Tables

1. Introduction to Interaction Learning

The trick to forgetting the
big picture is to look at
everything close up
Palahniuk [9]

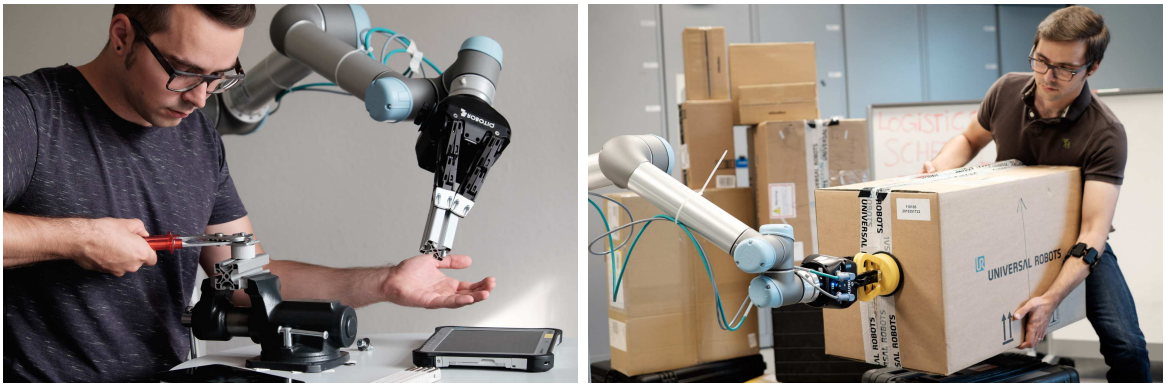


Fig. 1.1.: Robotic co-workers that work along-side humans are a great vision of robotics. Robots that are able to anticipate human motions could potentially change the environment we work in. They could for example release the physical load burdened on a human worker and in doing so increase their production efficiency. For that however human motions need to be constantly monitored and robot responses need to be seamlessly blended with that of the human co-worker. The figure shows typical interaction scenarios that are pursued in this thesis.

1.1. Robotic Co-Workers and Workplaces of the Future

Manufacturing companies have long sought for robots to aid manual production processes and lower fabrication costs. Over the last decade a number of applications emerged where robots are embedded in assembly processes, successfully transforming traditional workplaces into robot-aided assembly lines. Most facilities, however, rely almost exclusively on human operators to trigger robot motions and collaborations between humans and robots often proceed sequentially. As it stands, most human-robot collaborations are structured in a stop-and-go fashion, inducing delays, and following a rigid command-and-response pattern [10]. Despite notable benefits, there is a variety of

tasks where these widely used turn-taking implementations are infeasible. Hand-overs, joint assemblies and product transports are prominent examples where it is insufficient to play back prerecorded robot motions. Instead, they require continuous adaptation of the robot's motion to the human co-worker in order to be executed successfully.

Giving a robot this ability and allowing it to work side-by-side with humans would increase the number possible application scenarios and, more importantly, reduce the effort burdened on the human co-worker. In a joint transportation task for example, the physical load is typically shared among a group of workers and it could be significantly reduced by introducing a robotic assistant to carry most of the weight. Robots able to engage in such collaborations, have consequently the potential of improving workplace ergonomics and offer substantial reductions in injury and serious health hazards.

However, these seemingly simple and repetitive behaviors appear automatable at first, yet they are often done manually and robots are caged wherever people work among them. The reason for that results from the extensive set of skills, required to participate in interactions safely. Among others, *human-aware responses* and *spatiotemporal awareness* of the robot tend to be the most important aspects of a human-robot interaction. Traditionally, a skilled programmer is required to implement all possible interaction parameters with control routines. Unfortunately, this approach is intractable even for a small number of interactions as they are inherently hard to foresee beforehand. This disadvantage becomes considerably important in low-volume production processes. Often, assembly lines are altered and changed for new products in short intervals, requiring the robot to be re-configured to the new situation. As a result, interaction parameters need to be re-implemented regularly which clearly increases maintenance and production costs. Hence, efficient programming techniques are called for that allow fast and reliable skill acquisition, so that even untrained users are able to make efficient use of their new robotic co-worker.

Newly programmed robot behaviors need to be blended seamlessly with the user's motion during a collaborative task while at the same time being demonstrably safe. This requires a high level of coordination of the robot, so that it is able to produce well-synchronized actions with human co-workers. The human-centered environment in which the robot operates further depends upon dynamic timing of behaviors and appropriate motion plans.

A work place of the future, where robots work jointly with humans displaying the kind of fluency that humans are accustomed to, is envisioned in this thesis (see Fig. 1.1). For this vision to become reality, however, an algorithmic foundation is needed that allows efficient training of robots by non-professionals while providing flexible and adaptive motion generation mechanisms.

1.2. From Imitation Learning to Interaction Learning

Leveraging human knowledge to train robots is a fundamental problem in robotics. Inspired by nature *imitation learning* has emerged as a valued tool for bootstrapping

motor skills in various anthropomorphic structures [11]. It is generally referred to as the concept of learning the actions and postures required to do a task by observing a human teacher.

So far however, imitation learning is dominated by studies of individuals acting alone and it is mostly regarded as single agent approach. Yet, many day-to-day behaviors require the ability of two or more individuals to coordinate their motions concurrently to interactions such as in dances, collaborative transportations or hand-overs. These cooperative interactions are constantly influenced by contextual cues, e.g. postures of interaction partners or object positions. Still, humans are able to engage in them remarkably fluent. They tend to anticipate future actions of their interaction partners and adapt their behavior accordingly. In doing so they continuously reduce the discrepancy between the intended goal and the actual situation by changing temporal and positional courses [12]. In a high-five interaction for example, the individual movement of both interactants naturally changes in each repetition due to varying muscular activations. However, the interaction can be easily carried out, retaining the intent of the joint behavior, i.e. to clap the other's hand. The reason for that is the underlying mutual understanding of both partners, which often results in visible body synchrony and entrainment [13]. This leads to the hypothesis that the imitation of a single behavior by robot is not sufficient enough in order to engage in human-robot interactions. Instead, it stands to reason if imitation can be used to *imitate the interaction* by learning from two demonstrators instead of focusing on a single trainer.

Mimicking the behavior of one human during a two-person interaction is challenging problem for a robot. It requires continuous prediction of the other's behavior and interdependent motion planning as each action influences the outcome of the respective other. Giving a robot this ability would change how it is able to interact with people, allowing a wide range of application scenarios. Imitation learning seems to be a promising methodology to achieve that goal, as it offers a number of valuable properties that render it suitable for learning joint actions[1].

First, it allows the design of robot actions based on a trainers understanding of how the interaction should look like [14]. The underlying hypothesis is that humans would prefer to interact with robots the same way as they do with other people [15]. By imitating human motions in a human-robot interaction, no additional instructions would be required to train users as the robot's behavior would appear intuitive and inherently natural.

Second, it enables robots to mimic human behavioral patterns during a joint task. Psychological studies suggest that for coordinated actions between robots and humans to emerge, human-like trajectories are essential [16]. By adopting the role of one of the demonstrators, the robot would able to imitate postures and temporal properties to coordinate its action with the user. This would benefit its acceptance, seeing that its actions appear not only functional but also predictable [17, 18].

[1] In this context a joint action is considered a goal-directed action of two interactants that requires precise manipulation towards a Cartesian coordinate, such as hand-overs or collaborative transportations.

From a demonstrator's point of view, imitation learning provides a powerful knowledge-transfer tool. In contrast to low-level control techniques, the concept of teaching someone by demonstrating a task is deeply rooted in the human learning process. The resulting efficiency with which knowledge is transferred is the main reason for its growth of popularity. Skills that can take weeks to master by oneself are often learned by observing a teacher, reducing the cumbersome trail-and-error process to a mere set of failed attempts.

From a computational point of view there is a general consensus that pre-imposing a programmer's design to a robot's control structure is counterproductive as researchers strive towards adaptive robots [19]. Within this context imitation is an intriguing concept as it can be seen as learning tool on a behavioral level. Instead of explicitly defining motor commands abstract motion templates that allow for adaptation in unseen situations can be trained [11].

Recent advances in sensing technologies also encourage the use imitation learning for human-robot interactions. The latest developments of motion capture hardware has impacted professional systems that target film industries as well as consumer-grade devices. New capturing systems give users the ability of capturing gestures and movements with high quality and low latency. Some of the system even allow the recording of several people at the same time [20–22]. This offers additional insight into behavior patterns and correlation structures of several interactants. One key aspect of these parallel recordings is that they provide information about body synchrony and spatiotemporal relationships, features that are not available in single actor recordings. It is hypothesized that, instead of explicitly programming the robot for each foreseeable situation, these two-person recordings can be used to learn interactive behaviors.

Despite notable benefits, imitation also presents a number of difficulties that need to be addressed. Consider the reproduction of a single human movement on a robot for example. It can be easily assumed that the human's skeletal structure and the robot's kinematic chain differ from each other. As a result, motions cannot be transferred without further optimization, a problem that is generally referred to as the *correspondence problem* [23].

In addition to that, imitation by itself requires a significant amount of perceptual and cognitive abilities [19]. A skill that is to be imitated needs to be observed, recognized and internally represented. The optimal implementation however is still an open research question and it is actively discussed in the community. Yet, one common understanding it that it requires efficient integration of visual, memory and motor systems so that a robot is able to infer actions during runtime [24]. In a hand-over task for example, a robot needs to detect objects, anticipate hand-over locations and adapt the learned behavior to new positions. For that the skill of spatial reasoning about the user's intent, i.e. the hand-over location, as well as the prediction of temporal properties, i.e. when the hand-over will take place is required. In other words, the robot needs to generalize demonstrated behaviors spatially and temporally so that motions can be inferred for varying environmental conditions.

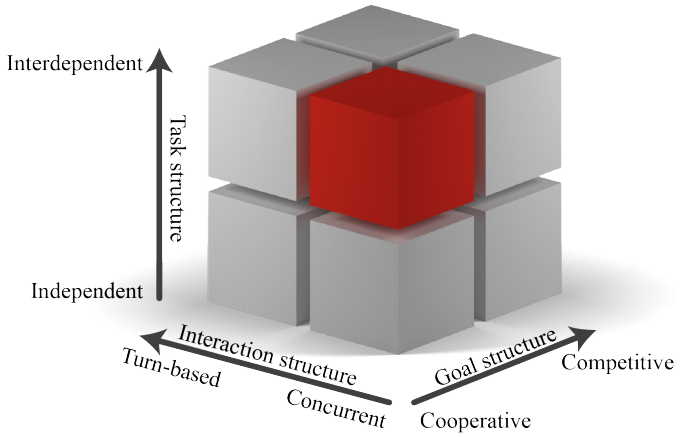


Fig. 1.2.: Overview of interaction structures. Day-to-day tasks such as hand-overs and object transportations fall in the category of interdependent, concurrent, cooperative tasks. In contrast to turn-based interactions, interdependent behavior planning requires continuous adaptation of one's body movement and its implementation in robotic assistants is the primary goal of this thesis. The figure is adapted from [28, p.2]

1.3. Interdependent, Concurrent and Cooperative Human-Robot Interaction

Similar to imitation as being seen as an approach focused on a single agent, dyadic interactions among humans have also been examined mostly from a single perception point of view in neuroscience [25, 26]. Here, interactions are regarded as the individual's motion simultaneously or sequentially executed that affect the immediate and future outcomes of the other individuals involved in the situation [25]. In these, stand-alone roles dominate the field and surprisingly few physical human-human interaction analyses exist. More recently, however, a shift in perspective can be observed and neuroscientists are also beginning to asses neuro-cognitive mechanisms, investigating what enables humans to coordinate actions. Instead of seeing interactions as individual motions with interaction-relevant properties, joint actions are given priority, focusing on both behaviors at the same time. Interestingly, many properties of the joint action rest on the team members' cognitive processes, such as motor-based simulation, prediction and behavior anticipation [27]. Resulting from this, it is argued that interactions should be differentiated into several distinguishable groups, moving towards a two-person neuroscience [28]. Following this argument and thereby emphasizing the importance of the second-person, it is proposed that interactions between humans can be differentiated by three properties: (1) interaction structure, (2) goal structure and, (3) task structure. In this taxonomy each property assumes one of two values as illustrated in Fig. 1.2.

The *interaction structure* is divided into *concurrent* and *turn-based* interactions. Here, the agents' motions are executed either simultaneously or as sequential actions that influence the immediate and current outcome of both individuals. In concurrent interactions both agents strive towards a common goal simultaneously, such as in races or martial arts whereas turn-based interactions follow a sequential order similar to a game of chess.

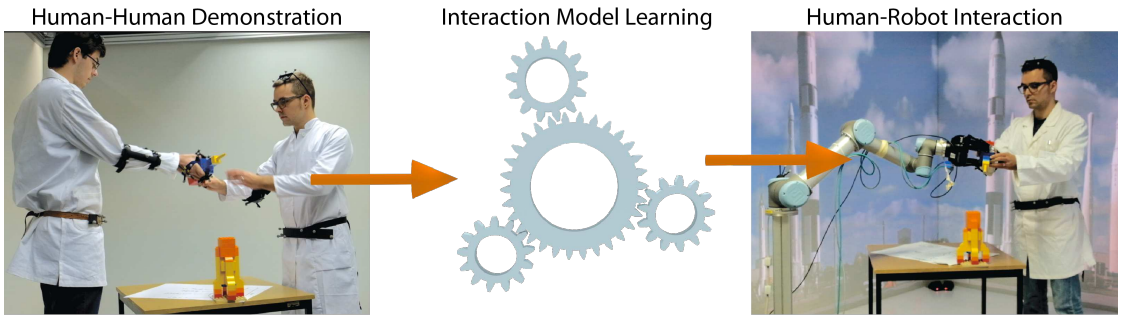


Fig. 1.3.: Overview of the interaction learning methodology. Based on human-human task demonstrations, an interaction model is learned that captures how the interactants moved during task demonstration. The model allows a robot to continuously adapt its own motion during a human-robot interaction spatially as well as temporally to that of its huat of its human interaction partnerr.

The *goal structure* of a two-person interaction is separated into *cooperative* and *competitive* scenarios. Here, agents may expedite the goal achievement of others, resembling cooperation or impede it competitively.

The *task structure* of an interaction defines whether motions are *independent* or *interdependent*. The first denotes interactions where each individual's behavior influences the task independently, such as in a race or a bowling game. Only the outcome of each behavior impacts the interactive task. Interdependent interactions on the other hand require adaptation of each individual's behavior in order to affect the outcome of the task, such as in carrying an object, dancing or martial arts. The most visible property of interdependent interactions is body synchrony.

The interaction learning approach presented in this thesis directly addresses concurrent interactions, that exhibit interdependent tasks structures and cooperative goals (see red highlight in Fig. 1.2). As outlined earlier, these tasks include but are not limited to hand-overs, collaborative transportations and assembly tasks, i.e. joint behaviors that emerge often in day-to-day life. By their nature these tasks require, in contrast to turn-based actions, continuous spatiotemporal coordination of the individuals' behavior and its outcome since both agents act simultaneously. On top of that both interactants operate on the same task, e.g. they carry an object jointly or hand-over items. This renders the task structure interdependent and each movement has to be planned with respect to the current situation.

For a robot to collaborate in these tasks with a human interaction partner, it is necessary to anticipate the human's behavior and adapt motor commands continuously. At the same time, the amount of possible interaction scenarios should not be limited to single behavior but rather feature a broad repertoire.

1.4. Problem Statement

There is a clear need for interactive robots that are able to interact with humans seamlessly and imitation learning might be the concept for achieving that goal. Despite its success, not much research has been conducted to extend it from single-actor imitation

to two-person interactions. In this thesis, it is proposed that by building upon task demonstrations of two human users, interaction skills of robots can be bootstrapped. Within this context, the challenge is how interaction dynamics, such as spatial relationships and body synchrony among the interactants should be gathered, represented internally and adapted during runtime.

The questions addressed in this thesis can further be formulated as follows:

- What steps are required to extend imitation learning to two-person interactions?
- Using motion capture, what additional knowledge about a joint task can be gathered and used by a robot during runtime to optimize human-robot collaboration?
- How can spatiotemporal generalization of demonstrated behaviors be achieved?

To address the above research questions, a methodology is presented in this thesis that utilizes motion capture recordings of two users (see Fig. 1.4). Based on these two-person task demonstrations, interaction dynamics are represented in several low-dimensional spaces, efficiently compressing motions while preserving intrinsic details. Spatial relationships of both interactants are captured in a topological space that allows for efficient optimization of robot responses during runtime. A novel data structure is developed that utilizes data of two interaction partners in single unified *interaction model*. Using the model during runtime a robot's response is inferred by aligning observed user motions spatially and temporally, before a robot's response is adapted spatially at each frame. The robot's motion is thereby seamlessly blended into the joint task with the human user and provides instant and safe responses. Using the proposed methodology a robot is able to participate and engage in interdependent cooperations, such as hand-overs or assembly tasks.

1.5. Contributions

In the following methodical and applicationr-related contributions made by this thesis are summarized.

Methodical Contributions

- An efficient approach for extracting human-robot interaction from human-human demonstration
- Data-driven fusion of methods for human behavior recognition and robot response generation
- Definition of a joint interaction model that generalizes the dynamics of trained behaviors spatially and temporally
- Introduction of triadic human-robot interactions based on intrinsic information in motion capture recordings

Application-related Contributions

- General framework for robot *and* virtual character in human-agent interactions
- Extensive behavior repertoire for interactive virtual characters
- Complex behaviors skills for collaborative robots in human-robot assembly tasks

1.6. Overview Over the Main Chapters

In chapter 2 current interaction learning approaches are reviewed. The survey starts by structuring and classifying each based on interaction types, real-time capabilities and action recognition features. In chapter 3 important mathematical concepts that are essential for the development of this thesis are introduced. Chapter 4 is the core of this thesis and presents a framework for learning human-robot interactions from human-human demonstrations (see Fig. 1.4). In subsequent chapters, the proposed methodology is evaluated in different scenarios. Beginning with chapter 5 interaction models are applied in a virtual agent setting to demonstrate how previously recorded motion capture templates are adapted to new situations. Then, the approach is utilized in a human-robot interaction setting in chapter 6. Here, a robotic arm learns to jointly assemble objects and coordinate its own motions with that of a human interaction partner. It will be demonstrated that, despite the different kinematic chains, a robot is still able to adapt human motion capture recordings to new situations. In chapter 7 the methodology is employed in a triadic human-robot handover setting. The triadic scenario allows, in addition to action recognition and body synchrony between both interactants, object rotations and positions to be represented during task demonstration. As a result, the robot is able to account for variations of such during runtime which results in more natural and intuitive human-robot handovers. The performance of the approach is evaluated in a within-subject study and compared to state-of-the-art handover methodologies.

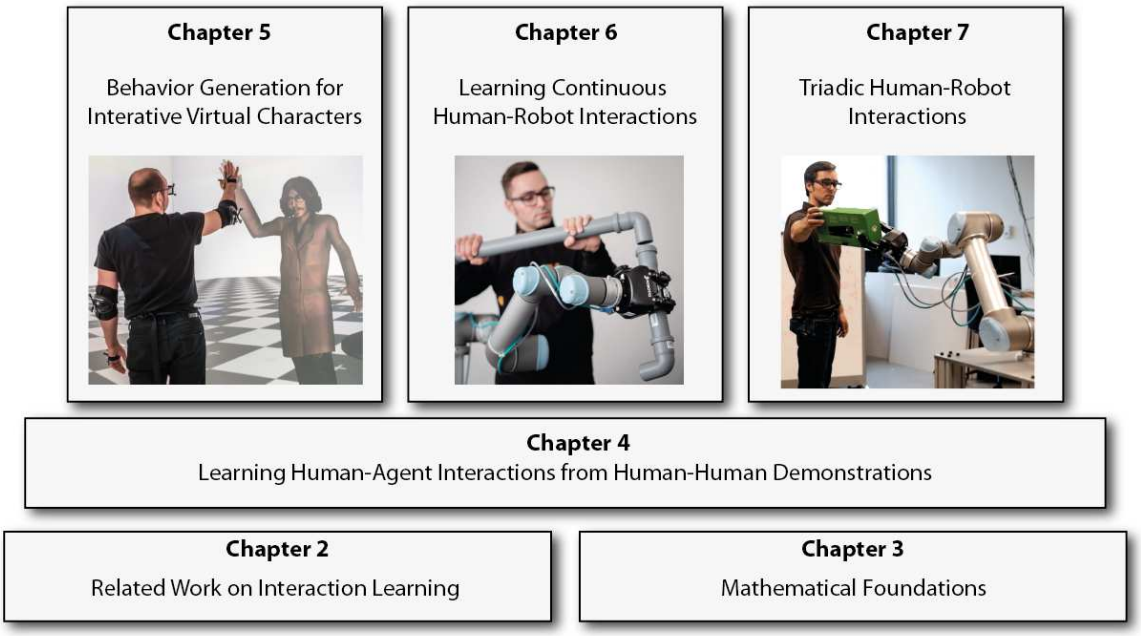


Fig. 1.4.: Overview of the main chapters. Based on related work on imitation learning in chapter 2 and the mathematical concepts in chapter 3 a novel interaction learning framework is developed (see chapter 4). The methodology is evaluated with virtual characters in a controlled virtual environemnt in chapter 5. Chapter 6 adopts the approach in complex human-robot interactions that involve body synchrony and spatiotemporal adaptation. In chapter 7 the framework is employed in a triadic human-robot handover setting.

2. Related Work

The extensive body of research in the field of Human-Robot Interaction (HRI) started to evolve around turn-based scenarios featuring cooperative goals. More recently this focus shifted to continuously planning robots that collaborate with people and approaches emerged that directly target concurrent behavior execution. The amount of open research questions and the increasing number of publications suggest that concurrent and interdependent interactions are the most difficult ones to implement, as they require continuous situation monitoring and motion adaptation in realtime. A similar shift can be observed in Virtual Reality (VR) research even though it is dedicated to character animation. Both fields seek control algorithms that enable agents to engage in seamless interactions with people and in doing so they focus on the same research topics such as motion recognition and response generation. Since the aim of this thesis is the development of an interaction learning methodology that is applicable in both areas, the following literature review introduces existing approaches from both fields. Their applicability to interaction learning, i.e. the learning from two-person task-demonstrations is discussed in detail and key benefits as well as limitations are highlighted.

2.1. Learning Human-Character Interactions

In the field of computer animation various methods have been proposed for animating virtual characters in interactions with users [29–33]. Despite these results most existing techniques require a skilled animator to control the animation and they rely on conventional input devices such as mouse or keyboard to trigger pre-defined motions interactively. More recently however, the advent of cost-effective depth sensors such as the Microsoft Kinect camera, enabled researchers to capture live user motions. As a result motion capture became accessible to a broad audience. With the rise of these systems traditional speech and gesture input paradigms gave way to adaptive animation techniques showing the growing interest in seamless human-agent interactions.

When developing such methods, however, one import necessity is the classification of human behavior during an ongoing interaction with the character. A successful classification allows an agent to react in a believable manner and, thus, provides a more intuitive and natural interaction experience to the user. At the same time the requirement for online motion adaption emerges. In order to react to human motions appropriately, the agent not only needs to classify the observed behavior but also adapt its own motion to best fit the current situation. Towards that end different

example-based methods have been introduced. In [34] for example the authors propose a framework for online action recognition using histograms. Here, live user motions are mapped on motion capture samples using Dynamic Time Warp (DTW). Additional temporal properties are preserved using dynamic programming.

Also based on motion capture recordings, Camporesi and colleagues [35] describe a character animation framework for real-time motion blending. For that, several example motions are captured using a reduced marker set. During runtime agent motions are generated by blending example motions to new situations. The underlying minimization problem, i.e. motion blending, is efficiently solved which allows for real-time motion generation.

Xiao and colleagues on the other hand propose a human-agent interaction system that relies on trained user gestures [36]. During training motion capture in conjunction with a data glove is used to extract user poses. Then a k-dimensional tree (k-d tree) is constructed based on the resulting 32-dimensional feature vector. During runtime a nearest neighbor search is employed to classify live user postures which in turn trigger agent responses. Since the classification accuracy highly depends on the employed distance metric, the authors argue for a general Mahalanobis-metric [37] in contrast traditional Eulclidean distances. In doing so higher classification accuracies are achieved for large data sets. However, as the method proposed by Camporesi [35] as well as Xiao [36] rely on exhaustive training data it is the user's responsibility to forsee all possible interaction scenarios. At the same time Xiao et al. do not optimize agent responses which increases the amount of required training data further. These limitation render both approaches unfeasible in real-world settings, i.e. when the amount of interactions increases.

To study the influence of interaction on human perception Taubert and colleagues [38] present an approach based on a hierarchical probabilistic model. Each layer is composed of a Gaussian process latent variable model (GP-LVM) with a Gaussian process dynamical model (GPDM) on top. At first, each kinematic marker of both interactants is mapped onto a latent variable in a bottom layer. The interaction between both actors is then computed in the second layer, called *interaction layer* and a mapping between three latent variables and the bottom layer is constructed. In a final layer, named *dynamic layer*, a GPDM encodes dynamic dependencies such as velocities and accelerations. During runtime interactive motions are inferred by back projecting the user's current hand position into latent space, yielding an interaction variable in the interaction layer which is in turn projected down to extract joint angles for character animation. However, due to the computational expensive nature of the underlying probabilistic algorithm and the requirement for exhaustive training data, model learning is a time consuming process. At same time only a single interaction can be synthesized during runtime limiting the system to one scenario.

2.1.1. Approaches based on Two-Person Demonstrations

Towards the goal of motion recognition and response generation, Vogt et al. learn a continuous mapping function between user poses and agent responses [3]. In contrast to the above techniques, the authors use motion capture data of two interactants. After recoding two-person interactions, a low-dimensional space is computed for each interactants' joint angle data using PCA. The problem of inferring an agent's response during runtime is interpreted as a mapping problem. It is argued that for each point of user's posture space a corresponding point the agent's posture space can be extracted. The required mapping function is approximated with an Artificial Neural Net (ANN). Querying the net during runtime using the user's motion as input, seamless agent responses are computed.

Due to the generalizing nature of the network, poses that are not part of the initial recording are still mapped to character responses. Also, the authors argue that by using a recurrent neural network instead of a traditional feedforward net, past user poses are incorporated allowing for various interactions to take place. One drawback is however that joint angle data is used for model learning which naturally yields a mapping function in joint angle space. The problem that arises is that a robot's posture might be optimal with respect to joint angle values but suboptimal in task space, i.e. tool center point coordinates. The lack of goal-driven optimization in task space renders the approach impractical in scenarios where precise manipulation is required.

The need for distance preservation has also been recognized by the computer graphics community, which eventually lead to Interaction Meshes (IMs). First proposed in non-realtime environments, IMs [39] are an adoption of Laplacian Mesh Editing (LME) [40] to virtual character animation. For that a mesh is constructed between the joints of both interactants as well as points sampled on their surface. Using LME the net is deformed to fit the animators' requirements, such as motion styles, character heights or varying skeleton proportions. It is thereby important to note that the net's topology has a strong influence on its overall generalization capability. A thorough analysis of this is presented in chapter 5.

The method has later been adopted to human-agent interactions [41]. For that additional constraints such as foot contact constraints and velocity constraints have been included into the optimization procedure. However, in order to react to ongoing human motions an appropriate reference IM has to be selected at each time step. For this, Ho and colleagues use a k-d tree classification to store all postures of the active interactant, i.e. the user. The k-d tree is queried during runtime with the user's current posture as input, yielding a motion capture frame from the initial recording as well as the corresponding IM.

One problem that arises is that similar user postures should trigger different responses of the virtual character depending on the current situation. The beginning of a high-five interaction for example might very well suggest a hand-shake interaction. However, using the k-d tree a character's response alternates between the behaviors depending on the closest match of the user's pose. This leads to visually unpleasing

results. At the same time Ho et al. synthesize a character's motion solely based on a single user posture and the last neighbor search result. This limits the temporal context of the response generation to two poses and long-range temporal dependencies are not considered. In order to distinguish between interactions that share similar segments, larger memories have to be taken into account to animate the character in a believable/contextual manner. Since the required contextual information is not incorporated, the method thus fails to compute reliable results for varying interactions.

The above algorithms indicate an interest in adaptive motor control schemes that allow for seamless human-agent interactions. A common goal among researchers is an increase in user immersion which is achieved by generating agent responses tailored to the current situation. In doing so, motion capture is a key technology as it is used to track user motions in the real world while at the same time enhancing the user's experience as no traditional input devices such as key boards are required [38]. An important insight is that by utilizing one's own motion as the main way of interacting with the agent more intuitive and natural interactions emerge. Also, adaptive responses that take the specifics of the current situation in to account are in general perceived more natural in contrast to predefined animations. This increases the user's experience further.

Focusing on the same naturalness and intuition, roboticists strive for similar goals in human-robot collaboration research. However, whereas virtual characters operate in simulated worlds, robots need to consider physical limitations and constraints of the real world. This increases the complexity of response generation algorithms by a magnitude. Nevertheless a number of approaches have been proposed and the following section presents an excerpt of them.

2.2. Learning Human-Robot Interactions

The previous section introduced character animation techniques that allow virtual agents to respond to human motions in a natural and intuitive way. A key aspect was that by using human motion capture data intuitive and adaptive agent responses can be triggered. Pursuing the same responsiveness in robots, roboticists proposed various control approaches which are reviewed in the following. The survey is structured by the underlying learning algorithm and each method is discussed regarding its benefits and limitations in interdependent human-robot interactions.

2.2.1. Approaches based on DMPs

Generalizing robot motions from task demonstrations is a long sought goal of roboticists. With the advent of Dynamic Movement Primitive (DMP) a novel learning approach has been proposed that allows movement planning under uncertainty and allows inference in unseen situations [42]. The concept follows the assumption that complex behavior skills are composed of several building blocks and DMPs are considered the

mathematical representation of such. They are fundamentally based on dynamical systems, i.e. point attractors. A combination such can be used to model movements over time. Several lines of work have spun from this approach.

In [43] for example, the concept of DMPs is used and extended to human-robot handovers. Here, Prada and colleagues update target positions of a attractor continuously during runtime to generalize to unseen positions. In a similar fashion, Ben Amor and colleagues propose a mixture of interaction primitives, a variation of DMPs for robot arm control in human-robot interactions [44, 45]. By maintaining a distribution over DMP parameters, inherent correlations of the joint task are encoded. In doing so, a robotic arm learns to react to human motions such that behaviors can be executed jointly. Unfortunately, interaction primitives require several past user poses as observations before reliable inferences about the robot's motion can be generated [46]. This increases the delay and the risk of generating responses that are unsuitable in the current situation. So far, interaction primitives have only be generated based on kinesthetic teaching. This can be cumbersome and not very efficient for complex interactions.

2.2.2. Approaches based on GMMs

Other approaches for learning human-robot collaboration use force measurements to anticipate human intentions. Rozo and colleagues for example propose a time-dependent method to mimic forces and positions in the context of transportation tasks [47]. Here a robotic arm is kinesthetically trained to match a user's hand while carrying an object. A GMM is computed to approximate task-specific parameters. The model is later employed during runtime to infer velocities, positions and forces in ongoing interactions.

GMMs are also used in dressing scenarios [48] where a robot learns to assist a human to put on a jacket. In this context the GMM models user joints captured from several depth images. During runtime the learned model is employed to recognize user motions and adapt pre-defined robot gripper poses to new positions.

Tanaka et al. [49] estimate user positions in an assembly line setup and implement a control scheme so that a robot passes objects and tools to the user whenever they are needed. Using a laser scanner the area of interest, i.e. the region in which the user operates, is first sampled from task demonstrations and then discretized equidistantly. To obtain binned data for training a Markov chain the positions are approximated by a set of predefined Gaussian distributions yielding a GMM. During runtime the user's position is captured by a laser scanner and the most likely distribution of the GMM is inferred. A robot trajectory is computed that specifically targets the mean position of the inferred Gaussian with an additional user-set offset. The underlying timing, i.e. when to pass the next object or tool, only depends on the previous state of the trained Markov chain. Whenever an object is picked up by the user, a timer is started and the next object is fetched and delivered.

While the approach shows interesting results in the considered assembly line setup, the static order of steps encoded in the Markov chain is restrictive in dynamic envi-

ronments. Situations might arise in which the user approaches its work place from a different angle and, thus, a different distribution in the GMM is inferred. Since the robot's motion and the corresponding timing are planned independently, the robot's behavior might be unsuitable or even obstructive in a given context.

Another force based approach is proposed by Evrard and colleagues [50, 51]. They present a dyadic haptic collaboration scenario in which a bipedal humanoid robot carries an object with a human user. These transportation tasks typically require an action strategy on how to perform the task collaboratively, i.e. a plan where motion directions and operation roles (leader/follower) are negotiated. In order to create such a plan, Evrard et al. compute a GMM over correlations of joint velocities, positions and forces of both the human and the robot. Unfortunately, the robot is controlled remotely during training, which is unintuitive especially for high degree-of-freedom robots such as robotic arms.

During runtime the robot interprets forces as guidance for a one dimensional lifting motion by utilizing the learned GMM. Doing so, Gaussian Mixture Regression (GMR) is employed to analyze perceived joint forces to compute suitable motor commands that best fit the current situation. Interestingly, the robot is able to seamlessly switch between leader and follower roles. However, only a single scenario can be modeled, rendering the approach unfeasible for settings with varying interactions.

2.2.3. Approaches based on HMMs

Similar to [50], Kulvicius et al. [52] address physical transportation tasks. Instead of modeling the forces directly, both interactants are treated as two point particles coupled by a virtual spring. The forces applied by the interaction partner and, thus, to the particles, tell the robot how to adapt its own trajectory. In the early learning phase, the robot uses the measured force values to follow the human guidance during task execution. Perceived force and motion patterns are then used to incrementally learn a HMM that predicts the human's next action. Over time the robot learns to take over a more active role in the interaction.

Instead of using forces to differentiate between phases of an interaction, HMMs are also used to classify tracking data. In [53] for example human motions are mapped to high-level actions. Each of them corresponds to an HMM that encodes the robot's response. In order to generate a suitable robot behavior that take the current situation into account, direct marker control is used. For that pre-recorded motion capture samples are adapted to the robot's kinematic chain by attaching virtual springs to the robot's extremities which in turn pull joints towards recorded marker positions. During physical human-robot interaction an additional virtual spring is placed between the leading hand of the user as well as the robot and in doing so the robot is able to spatially adapt its behavior. Due to the temporal features of HMMs, interactions such as high-five or waist-turns can be created seamlessly at every frame. However, based on the single connection between both interactants only a limited amount spatial generalization is incorporated. This contrasts to the approach developed in the presented

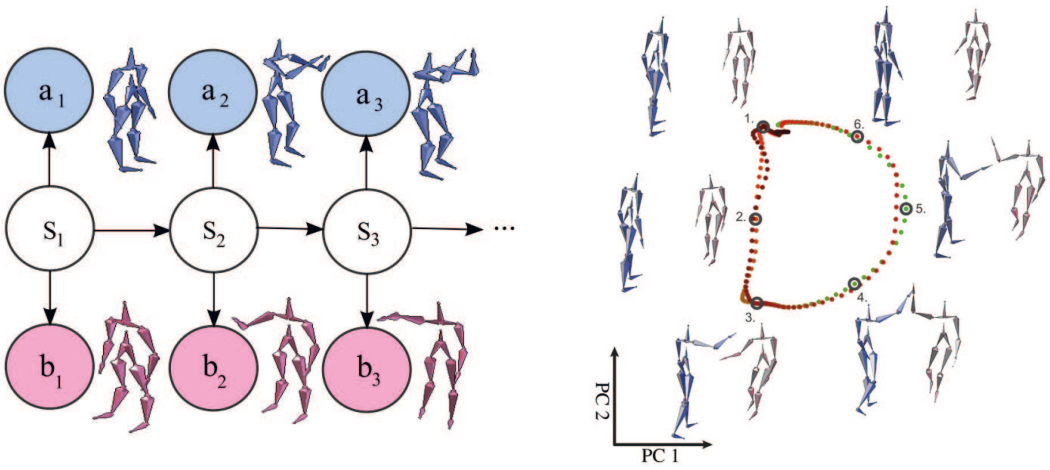


Fig. 2.1.: Left: A graphical model of a Path Map HMM. It relates time-series behavior of a cue-system, to an internal node system (depicted white). In contrast to traditional HMMs two observable nodes are connected (depicted red and blue). Here, joint angle data is observed from the human interaction partner (red) and mapped to corresponding joint angle data of the robot (blue). Right: A latent space of a high-five motion reduced in dimensionality using PPCA. Each point resembles a pair of poses reduced to 2 dimensions. Even if only one posture is observed during the interaction with a robot, response poses can still be inferred. The figures are adapted from [54, p. 3259]

thesis where additional relationships and, thus, more complex multi-joint interactions that involve body synchrony of both agents are trained.

To model the mutual dependency between two interactants Path Map Hidden Markov Models (pm-HMMs) are proposed by [54]. Typically, an HMM is defined in such a way that each hidden node is connected to one observable node only. In [54] however a different graph structure is assumed. Here, a path map relates the time-series behavior of a cue system to that of a target system. This is achieved by connecting each hidden node to two observables nodes: one observable for the cue system and one observable for the target system. A path map for the task of interaction modeling can be seen in Fig. 2.1.

As stated by Ben Amor et al. the HMM-based learning algorithm provides a rich set of tools for recognizing and estimating the current state of the interaction [54]. Yet, it comes at the price of high computational demands as well as limited generalization abilities. Using the HMM, no control over the robot's response posture is provided and only the most similar pose from the recording is inferred. Given an optimal output of the HMM, further optimization might be required in order to adapt the response to the current situation. For example, a high-five motion might be optimal in joint angle space, given the inference of the HMM, but it might also be suboptimal in tool center coordinates. What this results to is that optimization and response inference should be carried out in task space rather then joint angle space. This is in generally not the case in [54] which renders the approach unsuitable for precise and goal directed motions.

Another HMM-based approach is proposed by Medina et al. [55]. In contrast to the above methods, the model is utilized in a hierarchically clustered fashion to incrementally learn form human-robot interactions. It utilizes several channels of communication to gain experience in a unsupervised manner. In the beginning, the robot passively

follows user forces using a first order admittance controller while continuously storing forces in a database. If an input force occurs recurrently, the user is tasked to attribute that segment with a semantic description. This knowledge is used during runtime to ask about desired motion trajectories. Whenever a recognition is successful, either by querying it from the database or with additional certainty from human feedback, a response motion is generated by controlling gains of the admittance controller.

Targeting human-robot handovers, HMMs have also been used in [56]. Here, a two-stage process is implemented that, first handles the physical aspect of the hand-over tasks using the HMM, and, second, reasons about the user's commitment to the given task using a higher-level cognitive layer. The latter evolves from the behaviour expected from the human user. In contrast to other methods the HMM learned in [56] models states of the robot on a higher-level, e.g. *robot is picking up object* and *user grabs object*. During a live interaction, the user's overall commitment to the task is inferred by evaluating the current state of the HMM in addition to gaze directions and head orientations. In doing so, the robot is able to evaluate if the user is ready to receive the object and, if so, executes pre-defined motions accordingly.

However all actions of the robot follow the same motion providing no adaptations in joint angle space. For example, the pick-up location and the hand-over position of the object are assumed static, i.e. they remain unchanged throughout all interactions. Relying on the user's ability to accommodate for that limitation, e.g. by moving towards the robot, can be counterintuitive. While the above approaches present valuable insight for effective motion recognition, they still lack adaptation of such to account for spatial variations. As a result, additional optimization techniques are called for that allow online alternation of robot motions so that the current user pose can be taken into account.

2.2.4. Approaches based on IMs

In order to adapt robot responses to new situations while preserving spatial relationships between the interactants, IMs have been proposed [39]. To compute optimal paths for a robotic arm in semi-dynamic environments Ivan et al. [57] propose a dynamic variant of IMs. Here, proximities between the robot and its environment are modeled with IMs at frame level and their spatial relationships are maintained during motion execution so that the robot is able to adapt to changes in its surroundings. However, they are only incorporated offline, rendering the approach unsuitable for real-time human-robot interactions. At the same time, human motions are not taken into account and the interaction with them is not supported.

In a similar fashion, Ho and Shum [58] utilize IMs and create various net topologies between a humanoid robot and its constrained environment. In doing so they are able to compute a robot behavior that successfully avoids contact with obstacles while at the same time retaining the original shape of the motion. The control program, however, is computed offline and experiments are only shown in simulated worlds. This is due to the fact that the method requires vertices sampled on the robots surface and its

surrounding environment, which is unfeasible in real-world settings.

In contrast to the above Yang et al. [59] present an HRI scenario, where a lightweight robotic arm is avoiding human contact while executing a behavior. During task execution the robot's motion is encoded in IMs and net topologies are dynamically computed by evaluating distances between the robot's joints, its motion targets and detected obstacles. By planning the motion incrementally the robot's behavior is adapted during runtime and obstacles, such as users entering the workspace, are safely avoided. The task structure of the presented experiments can be considered independent and the robot essentially follows its own goal, i.e. continuing its motion. As a result the applicability to interdependent human-robot interactions is limited since these dynamic scenarios do no provide long-term targets for the robot to reach for. Similar observations hold true for the IM variants in [57] and [58].

The above approaches show the feasibility of IMs in the field of robotics. They do however lack user detection and state estimation which prohibits the robot to differentiate or react to user poses. This renders current IM methods inapplicable in interdependent HRI settings.

2.2.5. Hybrid Approaches for Modeling Human-Robot Interaction

In [60] Donner et al. propose a force-based approach for cooperative swinging of objects. Cooperative swinging is, in contrast to the applications scenarios pursued in this thesis, a repetitive and cyclic interaction. It features a very visible amount of body synchrony and requires precise timing as well as motion control to maintain certain energy levels, i.e. swinging heights. To manipulate this energy during an interaction with a human partner, the methodology shown in [60] continuously extracts pulse forces from a simplified pendulum model. This approach is, due to its low computational complexity and load during runtime, very suitable for cyclic time-dependent interactions. It scales, however, unfavorably to interaction structures that do not feature high degrees of periodicity. The application scenarios pursued in this thesis not only depend upon body synchrony but also task-space behavior coordination. This is not provided in the approach of Donner et al. which renders it unsuitable in the considered human-robot collaboration tasks.

From a more cognitive point of view Nikolaidis and Shah [61] show the advantages of cross-training for HRI. It is essentially a technique often used in human team training where all team members assume the others' role in a joint task iteratively. They thus, experience the collaborative plan hands-on from different perspectives. The authors argue that this is beneficial for a robot as it is able to better understand its human interaction partner. In order to do so, however, a shared model of the joint task is required. As the computational teaming model of the robot and the mental model of the human interaction partner converge, it is expected that the same pattern of internal states are visited. The similarity of both models is computed by evaluating the overlap of executed actions and the more actions are taken by both interaction partners the

better the computational model performs. In an extensive survey the authors highlight the increased interaction performance in terms of fluency. Here, a human user and a robot are tasked to drill a hole and insert a screw jointly . In the end an increase in concurrent motions by over 70 percent and a reduction of human idle times of 40 percent clearly shows the benefits of the shared model.

Addressing human-robot interaction timing, [62] present a robotic assistant scenario where a robot delivers parts and tools to a human worker. In contrast to the above approaches the robot's behavior is planned independent from that of the human interaction partner and objects are handed over passively with a tray. In a survey the authors compare interaction times of human-robot with human-human trials. They conclude that human-robot interactions last significantly longer, compared to human standards, despite the robot being noticed much earlier during an interaction. This is arguably not due to the robot's saliency, but rather the human's comfort which affects its perception. It stands to reason if the observed degradation of fluency during the joint task can be decreased by planning the robot's behavior interdependently from the human's motion. The authors state that this could be achieved with imitation learning, limiting the effects of discomfort through intuitive and legible motions.

In an earlier work, Sakita et al. [63] also address timing and interruption during human-robot collaboration. The scenario is developed around three ways of engaging in a Lego assembly task: (1) executing the action simultaneously with a user, (2) disambiguating a situation or (3) taking over for the human. The robot's action during runtime is inferred by estimating the user's intent using gaze directions and fixation times. While the interaction relies on nonverbal communication symbols, the approach is still strictly turn-based and it does not offer anticipatory motor control for the robot. More recently, Awais and Henrich [64] propose an intention estimation approach based on finite state machines for more ambiguous situations in which a human's behavior is not obvious. Instead of waiting for the situation to become clear, a proactive action is triggered and adapted to the most likely intention. Both methodologies, [63] and [64] do not include interdependent motor movement coordination or timing of the joint task.

Also focusing on fluency in human-robot collaboration, Cakmak et al. [65] directly address coordination timing in hand-over tasks. To suggest that a robot is ready to hand-over an object, several poses are defined. These are distinct from other things that the robot might be perceived to be doing when it has an object in its hand. This postural difference is called spatial contrast. In a similar vein the underlying timing during the hand-over is named temporal contrast. Cakmak and colleagues found that this positively effects the human's perception of the task and improves the readability of the robot's behavior. However, despite that finding, no elaboration on the robot's motion is given, focusing on its final posture instead. It is reasonably save to assume that motion has a similar effect on the robot's readability but further investigation is required to validate that hypothesis.

Timing in human-robot interactions has also been addressed in [66]. Here a Time Petri Net (TPN) is employed to model actions for a tower of Hanoi game, which is well

known for its ability to structure knowledge from various input channels, such as in multi-modal communications. Within this context, interaction fluency is interpreted as the robot's ability to shift from different actions and, thus, offering smooth recoveries after task interruptions.

In similar vein [67] and colleagues focus on timing in human-robot collaboration. They propose a probabilistic approach for user action recognition and time-sensitive planning of robot behaviors in an assembly task. Here, a robot is programmed to place and replace several boxes depending on the current progress of an assembly task. For that user actions as well as the state of the task have to be estimated to minimize the time spent on waiting for the next container. They propose a sensor model to heuristically infer the time when a robot has to switch boxes and they conclude that correctly estimating a sensors reliability is key to fluent human-robot collaboration.

Despite the results regarding motion and interaction timing the above approaches suffer from the lack of motion adaptation mechanisms. This limits the applicability of the methods but also indicates the need for optimizations techniques.

Similar to [59], Ceriani et al. [68] implicitly address timing in human-robot collaboration by proposing a collision avoidance methodology based on automatic task constraint extraction and temporal adaptation at frame level. In their approach the authors harness joint redundancies of a robotic arm and optimize a given trajectory to avoid human contact using additional distance sensors attached to the robot. The underlying velocity constraints of each actuator belong to one of the following types: (1) hard constraints, i.e. instantaneous constraints that possibly over-constrain the system, (2) skill constraints, i.e. constraints that can be relaxed without causing the system to fail (such as temporary suspension) and (3) soft constraints, i.e. constraints concerning the position and velocity of redundant actuators. Using a state machine template and the extracted constraints, a safety strategy is then automatically generated. During runtime a collision avoidance trajectory is computed for the robot based on the interaction danger assessment and Cartesian control [69]. In contrast to [59] no active tracking of human motions is incorporated and the system purely reacts at frame level by evaluating distance sensors. While this approach is suitable in the considered industrial applications, i.e. for avoiding human contact while continuing a given task, it is nevertheless unfeasible in physical human-robot collaborations where adaptive robot responses need to be triggered in order to retain a joint interaction goal.

2.2.6. Comparison of Methods

Many of the above approaches program robot behaviors explicitly (see Tab. 2.1 *Learning-based*). Despite the great progress in recent years, this paradigm still remains unintuitive and unfeasible especially for non-experts. The reason for that is the complexity and the dynamic nature of human-robot interactions. Many parameters cannot be foreseen beforehand which renders the problem inherently hard to program. As a result most approaches focus on a specific interaction, such as handovers, and do no scale to different scenarios (see Tab. 2.1 *Multiple Scenarios*).

Similar observations can be made with respect to learning-based approaches. In contrast to explicit programming however the generation of adaptable robot motions is not the limiting factor. Instead, recognizing user actions is the most difficult feature to incorporate. Besides the author's own results in the field, i.e. [3] and [54], only [47] and [45] allow varying interactions with the robot. Other approaches feature only a single interaction skill for the robot, such as handovers or reach motions. The reason for that is the difficulty in estimating the phase of an interaction when several scenarios are considered. Quite often similar user postures are obtained in several interactions, rendering contextual analysis a necessity. On the other hand, it is interesting to note that most learning methods feature interdependent motion generation and concurrent task execution, a benefit not present in explicitly programmed interactions. This is of course due to the inherently complex nature of two-person interactions.

While the presented methods produce impressive results they do not utilize temporal and spatial relationships of two interactants and they focus on a single agent instead, viz. [45, 47, 52]. In all reviewed approaches, the robot is trained kinesthetically in such way as to mimic human motion. While manually operating the robot circumvents the correspondence problem, it can be tedious and error prone task especially for high degree of freedom robots. Motion capture on the other hand offers the benefit of tracking two interacting partners simultaneously and consequently provides additional insight into body synchrony and dynamics. Also, users are captured unobtrusively and they are able to interact naturally. This in turn comes at the price of an additional optimization step required to project the human motions onto the robot's kinematic chain.

Adapting robot motions to new interaction scenarios with users can be achieved with various methods. Currently, there are two main research direction opposing each other. On one side DMPs have been proposed to model motor control using probabilistic models [70] and on the other side IMs that explicitly model proximities between the interactants. Both methods are able to adapt previously recorded behaviors to unseen situations. The latter however has the benefit that it can be applied instantly whereas DMPs require several past user postures to before robot responses can be inferred. This increases latencies of DMPs as pointed out by [46]. Also, DMPs extract GMMs from several task demonstrations in order to be able to generalize to new situations which in turn increases the amount of training data required to learn the model. IMs are generated from a single motion capture recording while still offering a reasonable degree of spatial generalization. In the end IMs seem to be superior in terms of training efficiency but their ability to generate seamless robot motions in varying interactions has yet to be confirmed.

2.3. Conclusion

The above literature review and amount of articles published in recent years clearly indicate an interest in interactive learning schemes. This research activity also suggests

Authors	Learning-based	Multiple Scenarios	Interdep. Motions	Concurr. Actions	Real-Robot
Maeda et al. [70]	●	●	●	●	●
Ewerton et al. [45]	●	●	●	●	●
Rozo et al. [47]	●	●	●	●	●
Vogt et al. [3]	●	●	●	●	●
Ben Amor et al. [54]	●	●	●	●	●
Kulvicius et al. [52]	●	○	●	●	●
Grigore et al. [56]	●	○	●	○	●
Nikolaidis and Shah [61]	●	○	●	●	●
Gribovskaya et al. [71]	●	○	●	●	○
Medina et al. [55]	●	○	●	●	●
Evrard et al. [50]	●	○	●	●	●
Yang et al. [59]	○	●	●	●	●
Gao et al. [48]	○	○	●	●	●
Chao and Thomaz [66]	○	○	●	●	●
Donner et al. [60]	○	○	●	●	○
Goto et al. [72]	○	○	●	●	●
Cakmak et al. [65]	○	○	●	○	●
Awais and Henrich [64]	○	○	○	●	●
Unhelkar et al. [62]	○	○	○	●	●
Ceriani et al. [68]	○	○	○	●	●
Tanaka et al. [49]	○	○	○	●	●
Sakita et al. [63]	○	○	○	○	●

Tab. 2.1.: A comparing overview of recent human-robot interaction methodologies which are closely related to the approach developed in this thesis. The table differentiates learning and model–based approaches that feature single or multi-task scenarios with interdependent/independent actions that are executed either concurrently or turn-based. The last column indicates if real-world experiments have been conducted or if results have been achieved only in simulation.

that no ideal approach has been found so far and that the community still strives for algorithms that allow intuitive skill transferal to robots. Reflecting that interest, several approaches have been put forward and learning from demonstration emerged as the most promising methodology. Imitation learning in particular is of significant importance as it utilizes the effectiveness of human teaching. The ease with which knowledge can be transferred to agents is the main reason for its popularity.

To evaluate the performance of these approaches several scenarios are envisioned and the field of robotics centers around industrial applications. Current settings range from collaborative assembly to transportation tasks.

Computer animation research on the other hand targets wider application areas. Whereas roboticists focus on programming for cooperating robots, computer animation scenarios are also diversified to competitive gaming setups through to disaster training systems. Nevertheless, the main vision - and to some extent the requirements - of the two research fields are similar. Both seek seamless human-agent interactions regardless of their individual task structure.

While recent results are encouraging, there are still various methodological shortcomings and limitations that need to be addressed. Most notably, recent literature lacks insight into problems that arise from the imitation of two-person interactions. Whereas current methods almost exclusively focus on a single agent, not much research has been presented that concentrates on two-person interaction skills. This thesis aims at improving on these limitations by generalizing the concept of imitation learning to *interaction learning*, so that synthetic artifacts can learn how to engage in seamless human-agent interactions. Special interest is put on interdependent interactions where both actors pursue a common goal and their behavior is influenced by the respective other. These interactions tend to be the most difficult ones to anticipate as their complex spatial and temporal relationships prohibit playback of recorded motions. Existing methods do not capture interaction dynamics or body relationships of the interactants, with the result that an agent's response does not take the ongoing situation into account. This increases the cognitive load burdened on the user and yields unintuitive interactions as he has to adopt to the robots behavior instead. Using human-human demonstrations it is hypothesized that physical interactions can be imitated by a robot, providing more intuitive and human-readable behaviors during runtime. Addressing a fundamental drawback of current methods, where only a single agent is observed, special emphasis is put on smooth and instant responses of the agent.

Also, existing methods for interaction learning are bound to a single kinematic structure which is the robot or virtual character at hand. They are, as a result, not applicable in human-character *and* human-robot interaction settings. In the presented work, virtual characters as well as anthropomorphic robots are controlled in various scenarios using the same learning framework. One key aspect is thereby that the method is purely data-driven and combines discrete action recognition with continuous movement control in a one-shot learning approach. Whereas existing methods require extensive training phases the presented methodology demands only a single task demonstration.

3. Mathematical Foundations

For a robot to learn from human-human demonstration it is essential that the complex relationships between interactants and their corresponding dynamics effects are captured. In this chapter core data representations that allow modeling of the various parameters are introduced. They are essential for the development of the interaction learning framework of this thesis.

Before learning an interaction, the behavior of two people demonstrating a collaborative task is captured in Cartesian coordinates. The representation of poses and kinematic chains is discussed in section 3.1. The resulting motion capture recordings are inherently high-dimensional and, thus, require Dimensionality Reduction (DR). A technique often employed in literature is PCA. Its mathematics and implications to tracking data is elaborated in section 3.2. Given a set of motion capture recordings, a robot needs to extract relevant information about joint relationships between the interactants in order to be able to generalize a demonstration to new situations. These relationships have been successfully extracted for character animation settings using differential coordinates. Their underlying mathematical concept is introduced in section 3.3.

Trajectories in Cartesian and low-dimensional space are essentially low-level representations of motions. However, some of the considered human-robot interaction scenarios are composed of several subtasks which require high-level recognition capabilities. In order to recognize human motions in these situations, HMMs are used.

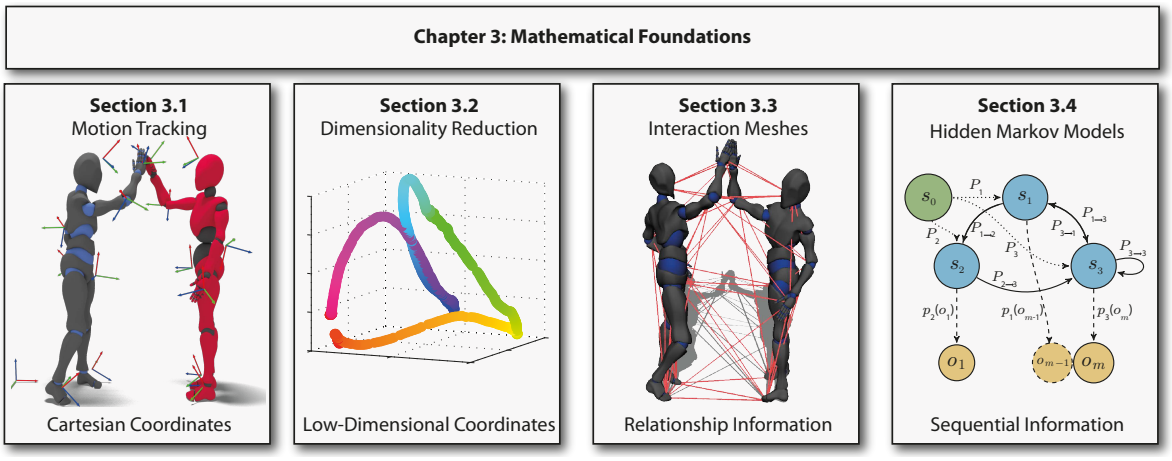


Fig. 3.1.: The figure visualizes coordinate representations that are introduced in this chapter. Each has its own advantages and limitations which are discussed in the corresponding section.

Overall, the chapter is structured in such a way as to give the reader a general overview of the mathematical concepts and representations involved (see Fig. 3.1). Each section features a broad introduction to the concept at hand. It provides by no means an exhaustive view on the matter and it is instead meant to highlight key aspects. The interested reader is referred to the respective publications, which are listed at the end of each section.

3.1. Motion Tracking and Representation of two-person Interactions

The human body is a highly articulated structure composed of a hierarchical system of joints. A posture is defined as a vector of joint rotations and positions with respect to a reference coordinate system (see Fig. 3.1 left). Motions are time series of postures and they can be different with various motion capture technologies.

Optical tracking systems, such as *A.R.T. ARTRACK5* [73] and *Vicon Bonita* [22] for example are based on markers attached to the users' bodies to record the movement of joints. Others reconstruct poses based on depth images, e.g. the *Kinect depth sensor* [74] or compute joint positions based on infrared video streams, e.g. *organic motion BioStage* [75]. In all of these systems a user pose or posture $\mathbf{p}$ is expressed in Cartesian coordinates as a vector of N joint positions. A motion $\mathcal{R}$ is then naturally defined as a time series of poses $\mathbf{p}_{1:T}$ over T timesteps:

$$\mathbf{p} = (x_1, \ldots, x_N, y_1, \ldots, y_N, z_1, \ldots, z_N) \quad x, y, z \in \mathbb{R} \tag{3.1}$$

$$\mathcal{R} = \mathbf{p}_{1:T} = [\mathbf{p}_1, \ldots, \mathbf{p}_t, \ldots, \mathbf{p}_T] \quad t \in \{1, \ldots, T\} \tag{3.2}$$

An interaction between two users is defined as two motions $\mathcal{R}^{oa}$ and $\mathcal{R}^{co}$ for the first and second interactant respectively.

When using marker-based tracking systems for motion capture, it is noted that marker positions do not correspond to joints directly. During calibration however, relative translations and rotations with respect to a human limb can be extracted and stored in a transformation matrix. The resulting affine transformation is applied at each motion capture frame, yielding recovered joint locations.

In order to record two-person interactions, two interactants are tracked simultaneously and their poses are extracted at each frame. When doing so, twice the amount of markers is required even though single person tracking already suffers from the their limited number. As a result, the human's highly articulated joint structure is captured only partially[1]. Nonetheless, when sparse maker set-ups are used, joint locations can be recovered using Inverse Kinematics (IK). Literature distinguishes between analytical and numerical approaches. The former are inherently hard to derive since high Degree of Freedom (DOF) kinematic chains are underconstrained. In general, there is an

[1] A.R.T. ARTRACK3 systems for example are able to track 18 makers at 100 Hz.

infinite number of joint angle configurations that match a tracked marker set. As a result, the numerical reconstruction methods are used in real-world situations where a analytical closed form solutions are infeasible [76, 77].

In the proposed two-person interaction model framework, motions can be represented either based on joint or marker locations. As marker set-ups are typically sparse, the latter often yield more compact representations. Using IK, it is however always possible to convert a motion defined via marker positions into a motion based on joint positions, if considered beneficial in the respective application.

3.2. Dimensionality Reduction of Motion Capture Data

Motion capture often yields very large datasets which is due to the large number of markers required to track the human body. Quite often 16 markers or more are tracked, resulting in 48 individual measurements per motion capture frame (3 Cartesian coordinates for each marker). However, it is known that human motion intrinsically lies on low-dimensional manifolds and Dimensionality Reduction (DR) should be applied to strip off redundant information [11]. The goal of DR is to reduce the number of required components to represent data, while retaining the information within it. Minimizing the amount of memory needed to store motion capture data is just one benefit of DR. More importantly, it can be seen as a tool for limiting a learning algorithm's search space. As these algorithms tend to be computationally expensive, reducing the search space improves runtimes, which in turn is of particular importance in real-time applications.

3.2.1. Principal Component Analysis

Over the years, different DR techniques have been experimented with and PCA[2] is a widely used linear DR variant. The main reason for its popularity results from the well founded mathematical concept as well as its low computational complexity. As compression of data is achieved with a single matrix operation, computational efficient solutions can be implemented, rendering the approach suitable in real-time applications.

Ben Amor and colleagues showed that the number of dimensions required to represent tracking data can be reduced to 2 to 3, while retaining up to 98 percent of the information [11]. This indicates that for most motion capture datasets two to four dimensions retain enough information to reconstruct motions without visible degradation. In addition, no hyper-parameters often found in other DR techniques are required.

[2]PCA is also known as *Karhunen-Loève transform* with the difference that it is also applied on non-centered data. In literature PCA and Karhunen-Loève transform are considered equivalent when applied to centered data sets.

As PCA balances compression and computational load favorably in most applications, it became one of the most commonly used DR methods for motion capture data.

From a mathematical point of view PCA reduces the dimensionality of a dataset $\mathcal{R}$ based on the covariance matrix Σ of modeled variables. DR is achieved by finding a small set of orthogonal linear combinations (the Principal Components (PCs)) of the original variables depending on the largest variance. For that the mean μ_k of each dimension k and the corresponding covariance is computed

$$\mu_k = \frac{1}{T}\sum_{i=1}^{T}\mathcal{R}^{k,i} \tag{3.3}$$

$$\Sigma_{i,j} = \frac{1}{T-1}\sum_{k=1}^{T}(\mathcal{R}^{k,i}-\mu_i)(\mathcal{R}_{k,j}-\mu_j) \quad . \tag{3.4}$$

The diagonal terms of Σ capture the variance of the individual features, whereas off-diagonal terms store the covariance between corresponding features. The key idea of PCA is to transform the data so that the covariance terms are zero.

Using eigenvalue decomposition Σ can be written as

$$\Sigma = \mathbf{U}\Lambda\mathbf{U}^T, \tag{3.5}$$

where $\mathbf{U}$ is an orthogonal matrix containing the eigenvectors of Σ and Λ is diagonal matrix with ordered eigenvalues λ_i of Σ. The eigenvectors hold the PCs, whereas the eigenvalues store the variance of $\mathcal{R}$. Using the eigenvalues the contribution of each dimension to the overall information is computed and evaluated. Essentially, the largest eigenvalue corresponds to the PC with the most information. The cumulative proportion of the variance that is captured by the first L dimensions is computed as follows

$$\frac{\sum_{i=1}^{L}\lambda_i}{\sum_{j=1}^{N}\lambda_j} \quad . \tag{3.6}$$

In order to compress the data while retaining the information within it, different amounts of eigenvalues have to be evaluated. In most applications the amount of original dimensions N is significantly higher than the number of dimensions L of the reduced space. The relation $L \ll N$ thus holds true and only a few eigenvalues are required in order to describe the variance of the data sufficiently.

PCA is a linear DR technique and the PCs describe an orthogonal sub-system. As a result, points in $\mathcal{R}$ are transformed into lower dimensions with a single matrix operation

$$\hat{\mathbf{p}} = \mathbf{W}^T(\mathbf{p}-\boldsymbol{\mu}) \quad . \tag{3.7}$$

Here, $\mathbf{W}$ denotes a matrix containing the first L eigenvectors column-wise of the data $\mathcal{R}$ with the mean vector $\boldsymbol{\mu}$. $\mathbf{p}$ is a point in $\mathcal{R}$ which is subject to DR. Reprojecting $\hat{\mathbf{p}}$ from

low-dimensional space into its original dimension is achieved by rewriting equation 3.7 into

$$\tilde{\mathbf{p}} = \boldsymbol{\mu} + \mathbf{W}\hat{\mathbf{p}} \quad . \tag{3.8}$$

It should be noted that $\tilde{\mathbf{p}}$ holds an error that results from the loss of information during DR. Overall, the square reconstruction error ϵ of all points is the sum of eigenvalues of those eigenvectors that were not considered for the construction of $\mathbf{W}$

$$\epsilon = \mathcal{E}\{\|\mathbf{p} - \tilde{\mathbf{p}}\|^2\} = \sum_i \lambda_i \ , \ \ i \in \{i \in N : i \notin L\} \quad . \tag{3.9}$$

Thus the largest eigenvectors not only account for the largest variance of the data but also minimize its reconstruction error.

3.2.2. PCA and Motion Capture Data

When applied to motion capture data, PCA creates a so called *posture space* [11] where each point corresponds to a posture and a trajectory to a motion when projected back to its original dimension. They are used in this thesis to compress motion capture data of two person interactions. In Fig. 3.2 for example a posture space of the giver's motion during a handover gesture is illustrated. The underlying motion capture data contains 6 optical motion capture markers and has been reduced to a 3, 2 and 1 dimensional posture space retaining 98%, 97% and 71% of the information respectively. Color is used to illustrate how points correlate in each low-dimensional space. Even though the space has been drastically reduced in dimensionality, a large portion of the initial information is retained. This is of particular importance for learning algorithms (see chapter 4).

The increased information density however is not the only benefit of PCA. Another advantage arises from the orthogonal linear sub-space that is created. It allows traditional distances metrics such as Euclidean distances to remain applicable. A typical application scenario where this feature is harnessed are motion capture databases where a vast number of postures are stored and queried during runtime. Here, a pivotal question is what motion capture frame resembles a given query pose most. For this a comparison to existing postures has to be computed. But instead of calculating differences in high-dimensional Cartesian space, low-dimensional representations can be used, reducing the computational load significantly [78].

3.2.3. Further Reading

PCA is an often used DR technique that is applicable in various applications and it is particularly well suited for preprocessing motion capture data for learning algorithms [11]. Yet, other non-linear DR techniques have also been experimented with and alternate approaches have been put forward. The interested reader is advised to review [11, 79, 80] for recent developments.

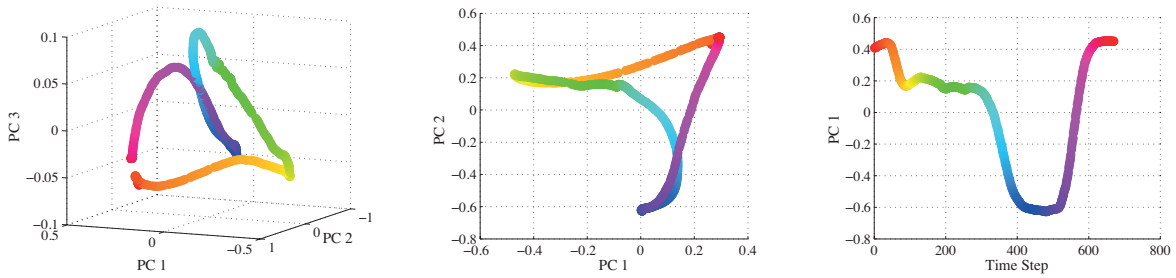


Fig. 3.2.: When applied to motion capture data PCA yields a posture space. The figure illustrates low-dimensional embeddings of an assembly subtask (see chapter 6) with 3 PCs (left), 2 PCs (middle) and a single PC (right). Color is used to indicate the consecutive order and correlation in low-dimensional space.

3.3. Laplacian Mesh Editing and Interaction Meshes

Laplacian Mesh Editing is a widely used free form and mesh deformation technique developed by the computer graphics community. First proposed by Sorkine and colleagues [40] its focal point is the conservation of geometric surface properties during modeling tasks. An important characteristic is that local details are preserved while still following a modeler's guideline on a global level. At its core, vertices are encoded in differential or Laplacian coordinates rather than their Cartesian counterparts. Since this graph-theoretical point of view is uncommon in the robotics community, its benefit and potential for interaction learning might not be obvious. As a mathematical understanding is essential for the development of this thesis, the characteristics of Laplacian coordinates are summarized in the following section. Subsequently the concept of LME is applied in a character animation setting. The mathematical additions required for that are formally known as Interaction Mesh (IM) [39] and they build one corner stone of the presented interaction learning framework. In chapter 4 a significantly extended version of IMs will developed to spatially adapt two-person interactions to new situations in real-time. It is thus important to highlight their inner workings in order to be able to improve on current limitations.

3.3.1. Laplacian Coordinates

In computer graphics meshes are vertices $\mathbf{V} = (\mathbf{p}_1, .., \mathbf{p}_N)$ with Cartesian coordinates $\mathbf{p}_i = (p_x, p_y, p_z)$ and a corresponding connectivity. For modeling operations however they are interpreted from a graph-theoretical point of view. Here, it is considered that a net is composed of N vertices that are sparsely connected to their topological neighbor by edges and thereby forming a bi-directional graph. This naturally allows formal definitions of neighborhood rings. In this sense, the neighborhood of vertex $\mathbf{p}_i$ is denoted by $supp(i)$ whereas the one-ring neighborhood, i.e. the vertices directly connected to $\mathbf{p}_i$, is denoted by $N(\mathbf{p}_i)$.

Given the neighborhoods of vertices in a mesh, differential coordinates can be derived. Laplacian coordinates are the simplest form of such and their coordinate representation is obtained by applying the Laplace operator $L(\mathbf{p}_i)$. Such an operator encodes the

variation of a function around the neighborhood of a vertex using a specific weighting scheme.

Let $S(\mathbf{p}_i)$ be a scheme of approximating $\mathbf{p}_i$ by a linear combination with other vertices

$$\mathbf{p}_i \approx S(\mathbf{p}_i) = \sum_{j \in supp(i), i \neq j} w_{ij}\mathbf{p}_j \quad . \tag{3.10}$$

Essentially, the transformation $\mathbf{p}_i - S(\mathbf{p}_i)$ is the linear differential mesh operator of scheme $S(\mathbf{p}_i)$. There are various weighting schemes used in literature but uniform weighting and cotangent weights are the most common forms [40]. Despite the fact that the cotangent Laplacian best fits the continuous Laplace-Beltrami operator, the former is often used in real-time applications due to its numerical stability [81]. Using uniform weighting a Laplacian coordinate $\mathbf{p}'_i$ of the Cartesian equivalent $\mathbf{p}_i$ is defined as the difference between $\mathbf{p}_i$ and the average of its neighbors:

$$\mathbf{p}'_i = L(\mathbf{p}_i) = \mathbf{p}_i - \frac{1}{|N(\mathbf{p}_i)|} \sum_{j \in N(\mathbf{p}_i)} \mathbf{p}_j \tag{3.11}$$

Here, $|N(\mathbf{p}_i)|$ denotes the number of neighbors in the one-ring neighborhood of $\mathbf{p}_i$, i.e. the immediate neighborhood of $\mathbf{p}_i$. At its core Equ. 3.11 essentially describes $\mathbf{p}_i$ as a linear combination of its topological neighbors and thus encodes local details of the mesh rather than global metrics.

In order to account for all vertices Equ. 3.11 is reformulated, yielding the mesh Laplacian $\mathbf{L}$

$$\begin{pmatrix} \mathbf{p}'_1 \\ \vdots \\ \mathbf{p}'_{|V|} \end{pmatrix} = \mathbf{L} \begin{pmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_{|V|} \end{pmatrix} \qquad \mathbf{L} = \begin{cases} -1 & \mathbf{p}_j \in N(\mathbf{p}_j) \\ \frac{1}{|N(\mathbf{p}_i)|} & i = j \\ 0 & \text{otherwise} \end{cases} \tag{3.12}$$

It is noted that this operation accounts for all Cartesian components of $\mathbf{p}_i$ and $(\mathbf{p}'_{i,x}, \mathbf{p}'_{i,y}, \mathbf{p}'_{i,z}) = L(\mathbf{p}_{i,x}, \mathbf{p}_{i,y}, \mathbf{p}_{i,z})$ holds true. This in turn implies that $\mathbf{L}$ is a block diagonal matrix.

In the end, solving Equ. 3.12 yields Laplacian coordinates for all vertices. Reversing that operation however is computationally expensive (see Equ. 3.13). Since $\mathbf{L}$ can be singular with rank $N-1$, it is not invertible and the uniqueness of the solution is not guaranteed.

$$\begin{pmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_{|V|} \end{pmatrix} = \mathbf{L}^{-1} \begin{pmatrix} \mathbf{p}'_1 \\ \vdots \\ \mathbf{p}'_{|V|} \end{pmatrix} \tag{3.13}$$

To nevertheless recover Cartesian coordinates assumptions on the initial transformation have to be made. A technique often employed in literature is the fixation of vertices. E.g. for computer animation tasks, a skilled modeler is required to in-

troduce a set of positional constraints $\mathcal{C}$ which force selected vertices to user-defined positions [81, 82]. Given a set of those, vertices are anchored, i.e constrained as follows

$$\mathbf{p}_i = \mathbf{c}_i \quad i \in \mathcal{C}, \mathbf{c}_i \in \mathbb{R} \quad . \tag{3.14}$$

Each constraint is added as an additional row to Equ. 3.13 rendering the system overdetermined. It is essential to know that at least one constraint has to be added ($|\mathcal{C}| \geq 1$) in order to ensure full rank and, thus, the uniqueness of the solution. A constraint is also attributed a weighting term w_i to model its importance. Reformulating the system in Equ. 3.13 to account for all constraints yields the following System of Linear Equations (SLE)

$$\begin{pmatrix} & \mathbf{L} & \\ & & \\ \cdots & w_1 & \cdots \\ & \vdots & \\ \cdots & w_{|\mathcal{C}|} & \cdots \end{pmatrix} \begin{pmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_N \end{pmatrix} = \begin{pmatrix} \mathbf{p}'_1 \\ \vdots \\ \mathbf{p}'_{|N|} \\ w_1 c_1 \\ \vdots \\ w_{|\mathcal{C}|} c_{|\mathcal{C}|} \end{pmatrix} \tag{3.15}$$

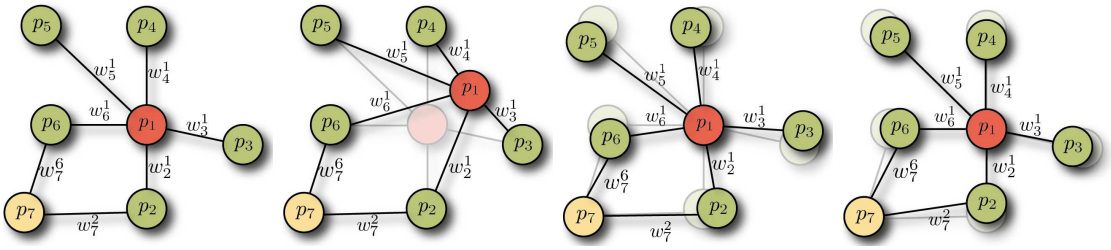
Solving the Equ. 3.15 in a least squares sense is equivalent to minimizing the error functional

$$E_L(\mathbf{p}) = \sum_{i=1}^{|V|} \frac{1}{2} \|L(\mathbf{p}_i) - \mathbf{p'}_i)\|^2 + \sum_{j \in \mathcal{C}} w_j \|\mathbf{p}_i - c_j\|^2 \tag{3.16}$$

Using least squares to solve the SLE is often favored in real-world applications due to its numerical stability [40]. However, the approach holds an error and is generally ill-conditioned. As a result, if one is to convert the mesh back and forth between the coordinate representations, low-frequency errors can be visible [83]. To limit these effects, additional constraints can be added [82, 83]. A typical application scenario where this is required is Laplacian Mesh Editing (LME). Its operating principle is introduced in the following.

3.3.2. Laplacian Mesh Editing

The process of modeling is often expressed explicitly in Cartesian coordinates although it is more desirable to model shapes intrinsically, i.e. as a set of functions, to preserve local detail during editing. Among others LME [40] established itself as a valuable framework which features such operations. It does so by interpreting modeling as a local neighborhood manipulation rather than an explicit global transformation. The key idea of LME is that translations in Cartesian space are reflected as positional changes in Laplacian coordinates and, consequently, expressed by weight changes to



(a) Undeformed mesh **(b)** Translation **(c)** Rotation **(d)** Scaling

Fig. 3.3.: Using Laplacian coordinates the process of modeling is expressed in local coordinates (see (a)). (b): vertex p_1 is translated and its transformation is represented in weight changes to its topological neighbors. Since p_7 is not part of the direct or one-ring neighborhood of p_1 its transformation is not reflected in w_7^6 and w_7^2. (c) and (d): rotational changes and scaling operations are not reflected in local coordinates since Laplacian coordinates are only sensitive to linear translations.

topological neighbors. Fig. 3.3 illustrates this and shows how distances change in the neighborhood of vertex $\mathbf{p}_i$ during global modeling operations.

During modeling, user-defined translations of vertices are essentially expressed as constraints and the Laplacian differences to topological neighbors is to be minimized in order to apply the manipulation to the geometry. In doing so, modifications are distributed throughout the entire mesh while still preserving local surface details.

The initial error functional that governs the modeling process is at its core the Laplacian mesh operator

$$E_L(\hat{\mathbf{p}}) = \sum_{i=1}^{|V|} \frac{1}{2} \|L(\hat{\mathbf{p}}_i) - L(\mathbf{p}_i)\|^2 + \sum_{i \in \mathcal{C}} w_i \|\hat{\mathbf{p}}_i - c_i\|^2 \tag{3.17}$$

The current locations of vertices are denoted by $\mathbf{p}$ and their new user-intended positions are indicated by $\hat{\mathbf{p}}$. In essence, Equ. 3.17 captures the difference in Laplacian coordinates with the first term whereas the second accounts for the validity of added constraints. As illustrated in Fig. 3.3 Laplacian coordinates are only sensitive to translations with the result that the local structure of the mesh can not be scaled or rotated. One way to solve this limitation is to introduce a local transformation $\mathbf{T}_i$ for each vertex $\mathbf{p}_i$ based on the assumed positions of $\hat{\mathbf{p}}_i$. The interested reader is referred to [40] and [81] for a formal derivation of $\mathbf{T}_i$. The key insight is however, that the following form of $\mathbf{T}$ can be used to account for isotropic translations, rotations and scalings can be used[3]

$$\mathbf{T}_i = \begin{pmatrix} s_i & -h_{i,z} & h_{i,y} & t_{i,x} \\ h_{i,z} & s_i & -h_{i,x} & t_{i,y} \\ -h_{i,y} & h_{i,x} & s_i & t_{i,z} \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.18}$$

The elements of $\mathbf{T}_i$ are derived from $\hat{\mathbf{p}}$ so that they are a linear function of $\mathbf{p}$. In this sense s_i, h_i, t_i denote the unknowns and a valid parametrization of them is obtained

[3] A formal proof of Equ. 3.18 can be found in [84].

by solving a SLE, cf. [40]. The final modeling operation is formulated with $\mathbf{T}_i$ and the constraint matrix $\mathbf{C}$, which holds the weights w_i in column i (see Equ. 3.19). Solving the system for $\hat{\mathbf{p}}$ yields new Cartesian coordinates for the vertices.

$$\begin{pmatrix} \mathbf{L}-\mathbf{T}_{x,x} & -\mathbf{T}_{x,y} & -\mathbf{T}_{x,z} \\ -\mathbf{T}_{y,x} & \mathbf{L}-\mathbf{T}_{y,y} & -\mathbf{T}_{y,z} \\ -\mathbf{T}_{z,x} & -\mathbf{T}_{z,y} & \mathbf{L}-\mathbf{T}_{z,z} \\ \mathbf{C}_i & 0 & 0 \\ 0 & \mathbf{C}_i & 0 \\ 0 & 0 & \mathbf{C}_i \end{pmatrix} \begin{pmatrix} \hat{\mathbf{p}}_x \\ \hat{\mathbf{p}}_y \\ \hat{\mathbf{p}}_z \end{pmatrix} = \begin{pmatrix} \mathbf{p}_x \\ \mathbf{p}_y \\ \mathbf{p}_z \\ w_1 c_1 \\ \vdots \\ w_{|\mathcal{C}|} c_{|\mathcal{C}|} \end{pmatrix} \tag{3.19}$$

Note that the structure in Equ. 3.19 is slightly different compared to Equ. 3.15 to account for all three Cartesian components. In the following, rotation and scaling operations are not considered. Character animation problems are typically overdetermined by the number of positional constraints and feature densely connected meshes. As a result, rotation and scaling operations affect the one-ring neighborhood of anchored vertices and their preservation during optimization is already accounted for.

3.3.3. Interaction Meshes

LME is also used in the character animation community to retarget motions. Consider a kick interaction and the adaptation of such to new heights for example. Since the interaction naturally involves a large portion of the kinematic chain of both interactants, several joints have to be considered during retargeting. This includes the position of the kicking foot and the arms of the defending character. However, it also requires changes to knees, elbows and pelvises. Using traditional methods such as inverse kinematics to adapt the animation might results in visually unappealing results. Potential side effects include misalignment of joints, e.g. the defending hand not meeting the foot at the correct height, foot skating and collisions with the surrounding environment [39]. The reason for that result from the amount of joints and their complex spatial relationships that need to be adapted to retain the visual appeal of the animation. This in turn requires a skilled animator that fine-tunes each joint individually at each frame.

This a time consuming and tedious task that eventually led to the development of Interaction Meshes [39]. Proposed by Ho end colleagues, IMs are an interpretation of LME for close virtual character interactions. Instead of treating each joint and its relationship to others individually, IMs connect them to a mesh which is deformed to match an animator's requirement. Joints, points on body parts and environment contacts are thereby seen as vertices. The connectivity of them is computed by applying Delaunay tetrahedralization. Since it favors short distances over long ones, vertices in close proximity are connected by an edge. In doing so, a mesh is created that inherently focuses on close spatial relationships.

During adaptation the net is then deformed using LME. The spatial relationships of joints are encoded in differential coordinates, and thus, preserved during editing. Fig. 3.4 illustrates this for the previously mentioned kick example. Here, different kick

Fig. 3.4.: An Interaction Mesh is a topological representation of a character animation in differential coordinates. It has been proposed by Ho et. al [39] for motion retargeting in character interactions (left) or body scaling operations (right). The Fig. is adapted from [39].

heights have been defined by an animator. As depicted, the overall pair of postures remains similar, yet, adapted to the new situation. A second example is shown on the right hand side of Fig. 3.4. Here, body proportions are scaled and the skeletal structure of the interactants is adapted to different heights. Yet, the overall style of the interaction is preserved.

Positional Constraints In addition to the general requirement of at least one constraint to recover Cartesian coordinates (see section 3.3), IMs feature additional positional constraints to anchor body parts or joints to specific locations. Here, an animator defines constraints for vertices so that they remain at a given location during optimization and in consequence do not move during the animation. In the above kick interaction for example (see Fig. 3.4) the left foot of the yellow character is constrained to different heights, prompting the IM to adapt the remaining vertices. Doing so, vertices retain their relative distances to neighboring/connected vertices and the overall shape of the pair of poses is preserved. In a similar vein, feet are positional constraints to limit the effect of foot skating [39]. Contacts with surrounding objects are also modeled using positional constraints [41, 57, 59].

They are modeled similarly to LME constraints (see Equ. 3.14) using the following form:

$$C_P(\hat{\mathbf{p}}) = \mathbf{W}\hat{\mathbf{p}} - \mathbf{C} \quad . \tag{3.20}$$

In contrast to the definition above however, IM positional constraints are written in matrix form to account for larger constraint numbers. Here, $\mathbf{W}$ is a weight matrix that stores the weight w_i of each vertex $\hat{\mathbf{p}}_i$. $\mathbf{C}$ denotes the matrix for the value of the constraints, i.e. their Cartesian position. Similar to the right hand term of the LME error functional, an energy interpretation is derived

$$E_p(\hat{\mathbf{p}}) = \sum_{i \in \mathcal{C}} w_i \|\hat{\mathbf{p}}_i - c_i\|^2 \quad c_i \in \mathbb{R} \tag{3.21}$$

In essence the functional denotes the cumulative difference between a manipulated vertex position $\hat{\mathbf{p}}_i$ and its anchor c_i.

Bone Length Constraints Satisfying positional constraints during adaptation might not account for bone lengths and the skeletal proportions of the characters might vary during the animation. This can cause visual artifacts such as unnatural proportions or stretching of limbs. These can be limited using bone length constraints [39]. Here, the

preservation of a distance between two joints, i.e. the bone length, is defined by the following energy functional

$$E_b(\hat{\mathbf{p}}) = \sum_{e(i,j)\in\mathcal{E}} \left(\|\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j\| - l_{e(i,j)}\right)^2 \tag{3.22}$$

$\mathcal{E}$ denotes the set of edges/bones and $e(i,j)$ is the edge/bone connecting vertex i and j. $l_{e(i,j)}$ is the distance or length of a bone connecting $\mathbf{p}_i$ and $\mathbf{p}_j$. The error functional essentially forces the distance between of two joints to a desired length $l_{e(i,j)}$.

However, the distance term exhibits a non-linear component, rendering traditional least squares solvers are inapplicable. To circumvent this limitation a linearization can be introduced with the result that iterative Gauss-Newton methods can be employed to solve the system, cf. [39, 85].

Constraint Energy When animating virtual characters it is reasonable to assume varying constraint importances. The supporting foot for a kick motion for example is typically more important for a human-like motion than the precise preservation of a head position. Towards that end, constraints are separated into soft and hard constraints.

$$\mathbf{F}_i\hat{\mathbf{p}}_i = \mathbf{f}_i \quad \text{soft constraint} \quad \mathbf{C}_i\hat{\mathbf{p}}_i = \mathbf{h}_i \quad \text{hard constraint} \tag{3.23}$$

The preservation of soft constraints during optimization is modeled with a weighting term w_i for each joint. The overall constraint energy which accounts for the amount of violation is defined as follows

$$E_c(\hat{\mathbf{p}}) = \frac{1}{2}\hat{\mathbf{p}}^T\mathbf{F}^T\mathbf{W}\hat{\mathbf{p}} - \mathbf{f}^T\mathbf{W}F\hat{\mathbf{p}} + \frac{1}{2}\mathbf{f}^T\mathbf{W}\mathbf{f} \tag{3.24}$$

$\mathbf{F}$ denotes the constraint matrix which captures the joints that are attributed a soft constraint and $\mathbf{f}$ stores the corresponding constraint value. To weight the importance of each a weighting matrix $\mathbf{W}$ is introduced. It contains the weighting terms w_i for each constraint at column i.

Hard constraints on the other hand are equivalent to LME positional constraints (see Equ. 3.14). In typical application scenarios, bone lengths and one positional constraint (the supporting foot for example) are attributed a hard constraint whereas all remaining constraints are treated as soft constraint depending on their weight w_i.

Given the above constraints the problem of deforming an IM from its current Cartesian coordinates $\mathbf{p}$ to $\hat{\mathbf{p}}$ is equivalent to the following minimization problem

$$\underset{\hat{\mathbf{p}},1\leq i\leq|V|}{\arg\min} = \sum_{i=1}^{|V|} E_L(\hat{\mathbf{p}}_i) + E_b(\hat{\mathbf{p}}_i) + E_c(\hat{\mathbf{p}}_i) \tag{3.25}$$

Essentially, the above definition accounts for the deformation of the mesh in Laplacian coordinates (E_L), the validity of hard constraints (E_L), the preservation of bone

lengths (E_b) and the additional soft constraints (E_c). Similar to LME the problem is finally transformed into an SLE

$$\begin{pmatrix} \mathbf{M}^T\mathbf{M} + \mathbf{F}^T\mathbf{W}\mathbf{F} & \mathbf{C}^T \\ \mathbf{C} & 0 \end{pmatrix} \begin{pmatrix} \hat{\mathbf{p}} \end{pmatrix} = \begin{pmatrix} \mathbf{M}^T\mathbf{b} + \mathbf{F}^T\mathbf{W}\mathbf{f} \\ \mathbf{h} \end{pmatrix} \tag{3.26}$$

and solving it yields a set of vertex positions of the adapted IM.

In the resulting optimization problem the transformation $\mathbf{T}$ which initially accounted for isotropic translations, rotations and scalings, is omitted. This simplification is introduced since the amount of constraints that are typically added are sufficient enough to render the system well-defined. Also, when deformation operations do not feature scaling or rotation operations to a large extent, positional constraints added by an animator already preserve local details to a reasonable degree [83].

3.3.4. Further Reading

The concept of IMs is used in various applications ranging from character animation [39, 86] to robot motion planning [57–59]. The constraint definitions that are presented above are found in most of these implementations. It is noted however that other constraint types are also proposed. In [39] for example velocity constraints for limiting joint movement between frames or collision constraints that guard environment contact are implemented. Since the formal definition of IMs does not account for rotations explicitly, Maciel and De et al. [86] add rotational constraints.

The applicability in real-time applications, such as in human-character or human-robot interaction, has also been addressed. In [41], for example a online IM retrieval system is presented that optimizes a character response given a human pose. Since this system is closely related to the methodology presented in this thesis, a thorough comparison is presented in chapter 5.

3.4. Hidden Markov Models

Hidden Markov Models were originally developed in the context of natural language processing and speech recognition [87] but they are now a recurring theme in many pattern recognition and classification applications [88]. The thematic context in which they are employed focuses primarily on measurements and interpretations of sensor readings in terms of patterns that evolve over time. HMMs are trained with sequences of data and they model sequential information using probabilities. They are thus particular well suited for time series such as motion capture recordings and allow extraction of temporal contexts that are not easily accessible with other data representations.

In this thesis an HMM will be used for classifying human motion during human-robot interaction tasks. Towards that end an introduction to their general concept and working principle is given in the following.

3.4.1. Introduction

The key idea of an HMM is to model the statistical regularities that govern a process as exactly as possible. Its purpose is to infer information about the process it models and the future development of such. This applies to questions whether a data sequence is produced by the process or which state sequence has the highest probability. Consider the following human motion classification problem. For each observed pose sequence there is an internal state that corresponds to the behavior type such as a handshake gesture or high five. This type is not directly visible. What is observable are joint positions provided by a motion capture system. Given training data, that is an association of recorded poses to behavior types, an HMM is able to extract the probabilities of observing a behavior for a given sequence of poses. During runtime the user's current behavior is unknown but observed joint position and the internal model, i.e. the HMM reveal a belief what it could be. This allows the computation of the probability of transitioning between behaviors and the sequence of poses required to do so.

From a more mathematical point of view an HMM is essentially a stochastic process with an underlying second stochastic process that is not visible. The first process describes that for every point in time an observation is made. The probability of the observation occurring with it only depends on the current state. The second process is a Markov chain model which describes the random transition between model states using a transition probability. Each state depends only on its predecessor, a property that is known as the Markov assumption [87]. The state space of the model is finite and stationary. Also, the sequence that generated the observations is not visible from the outside view.

Formally, an HMM Θ is described as

$$\Theta = (\mathcal{S}, P_i, P_{j \to i}, p_i(o)) \tag{3.27}$$

where $\mathcal{S}$ is a set of finite states and $P_{j \to i}$ is a matrix containing transition probabilities for transitioning from state j to i. P_i is a vector of start probabilities and $p_i(o)$ is a matrix containing the probabilities of observing o in state i. o is typically modeled as a finite set of discrete observations.

As mentioned above, HMMs are, similar to their origins in Markov chains, based on two simplifying assumptions. First, the probability of a particular state depends only on its predecessor (Markov assumption)

$$p(s_i|s_1, \ldots, s_{i-1}) = p(s_i|s_{i-1}) \quad s_i \in \mathcal{S} \tag{3.28}$$

And, second, the probability of observing an output o only depends on the current state.

$$p(o_i|s_1, \ldots, s_i, o_1, \ldots, o_i) = p(o_i|s_i) \tag{3.29}$$

Fig. 3.5 illustrates the structure of a three state HMM graphically.

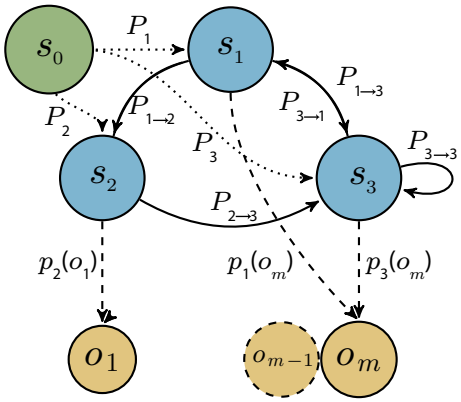


Fig. 3.5.: The figure illustrates a discrete HMM with three internal states $s_{\{1,2,3\}}$, an additional start state s_0 and several outputs $o_i, \ldots, o_m$.

3.4.2. Three Problems Related to HMMs

Given the above definition of an HMM three computational problems can be formalized and solved. The interested reader is encouraged to review Rabiner [89] for a more thorough and comprehensive definition.

- *Likelihood estimation* Given an HMM Θ, the likelihood $P(\mathbf{o}|\Theta)$ of observing the sequence $\mathbf{o}$ is to be determined
- *Decoding* Given Θ and $\mathbf{o}$, the sequence of hidden states is to be computed
- *Learning* Estimation of improved HMM parameters $\hat{\Theta}$ given $\mathcal{S}$ and a sequence of observations $\mathbf{o}$

Likelihood estimation targets the problem of computing the probability $P(\mathbf{o}|\Theta)$ of a observation sequence $\mathbf{o}$. This means with respect to the above behavior classification example, that the question how likely a sequence of poses is, can be computed. The probability of a posture is thereby seen as the likelihood of the data given every possible series of states. This generally requires summation over all possible state sequences, where $P(\mathbf{o}, \mathcal{S}|\Theta)$ is the output probability of a single path

$$P(\mathbf{o}|\Theta) = \sum_{i=1}^{|\mathcal{S}|} P(\mathbf{o}, s_i|\Theta) \quad s_i \in \mathcal{S} \tag{3.30}$$

Since this requires marginalization over all state sequences, the computational costs grow exponentially ($O(|\mathcal{S}|^T)$). However, using dynamic programming Equ. 3.30 can be efficiently computed using the *forward algorithm* [89]. The algorithm harnesses the fact that all HMMs are strictly time-synchronous due to the Markov assumption. Given Θ and a state it is irrelevant for future states on which path this state has arrived from. It is therefore sufficient enough to consider only possible states at time step t for the computation of future states.

The algorithm starts by defining an auxiliary variable α_i for the total probability of the observations through t time steps in state s_i

$$\alpha_i(t) = P(o_1, \ldots, o_t, s_t = s_i|\Theta). \tag{3.31}$$

This yields the probability that the first part of $\mathbf{o}$ up to time step t is generated and that the model Θ is in state s_i. Then, given this definition the probability of $P(\mathbf{o}|\Theta)$ is represented as

$$P(\mathbf{o}|\Theta) = P(o_1, \ldots, o_T|\Theta) = \sum_{i=1}^{|\mathcal{S}|} P(o_1, \ldots, o_T, s_T = s_i|\Theta) = \sum_{i=1}^{|\mathcal{S}|} \alpha_i(T). \tag{3.32}$$

The algorithm starts at $\alpha_i(1) = P_i p_i(o_1)$, i.e. the probability of generating the first observation element o_1 at the initial point in time $t = 1$ and for reaching state s_i. Based on the induction principle $\alpha_i(t)$ can be computed at each recursion step by evaluating

$$\alpha_i(t) = \sum_{i=1}^{|\mathcal{S}|} \alpha_i(t-1) P_{i \to j} p_i(o_t) \ , \quad j = 1, \ldots, |\mathcal{S}|, t = 1, \ldots, T. \tag{3.33}$$

In the end, the probabilities $\alpha_i(t)$ are obtained for each time step and the total output probability $P(\mathbf{o}|\Theta)$ is computed by summarizing over them

$$P(\mathbf{o}|\Theta) = \sum_{i=1}^{|\mathcal{S}|} \alpha_i(T) \tag{3.34}$$

As a result of this, only $O(|\mathcal{S}|)$ operations have to be computed at each time step, resulting in a final complexity $O(|\mathcal{S}|T^2)$ to compute $P(\mathbf{o}|\Theta)$. This is significantly lower than initially assumed.

Decoding can be seen as estimating the *believe state*, i.e. the probability of being in a state at specific point in time given a sequence of observations $\mathbf{o}$. With respect to the above behavior classification problem, the believe state is the assumed behavior of the human given a set of observed postures. In chapter 4 for example decoding will be used to recognize human motions during a human-robot collaboration.

Mathematically speaking, the state sequence $\mathbf{s}^*$ with maximal posterior state probability is sought

$$\mathbf{s}^* = \arg\max_{\mathbf{s}} P(\mathbf{s}|\mathbf{o}, \Theta) \tag{3.35}$$

By applying Bayes' rule, the posterior state probability is written as

$$P(\mathbf{s}|\mathbf{o}, \Theta) = \frac{P(\mathbf{o}, \mathbf{s}|\Theta)}{P(\mathbf{o}|\Theta)} \tag{3.36}$$

Since the total output probability $P(\mathbf{o}|\Theta)$ of the model is constant the equation is

simplified to

$$\mathbf{s}^* = \arg\max_{\mathbf{s}} P(\mathbf{s}|\mathbf{o}, \Theta) = \arg\max_{\mathbf{s}} P(\mathbf{o}, \mathbf{s}|\Theta) \tag{3.37}$$

The optimal path $\mathbf{s}^*$ consequently corresponds to the path that maximizes the probability of the observation sequence $\mathbf{o}$. Computing $\mathbf{s}^*$ can be achieved using brute force but the Viterbi algorithm is used in real-world applications due to its computational advantages ($O(|\mathcal{S}|T^2)$). In contrast to the above forward algorithm, the Viterbi algorithm utilizes a maximization step instead of summation over the probabilities of predecessor steps. The initial step before starting the recursive procedure is to define probabilities for partially observed sequences up to $\mathbf{o}_t$ with maximal probability and end state i

$$\delta_i(t) = \max_{s_1,\ldots,s_{t-1}} P(o_1, \ldots, o_{t-1}, s_1, \ldots, s_{t-1}, s_t = s_i|\Theta) \tag{3.38}$$

Since the optimal path may change over the period of T time steps *backward pointers* $\psi_j(t)$ are defined along the partial paths. These store the optimal predecessor state for every corresponding $\delta_j t$

$$\psi_j(t) = \arg\max_{\forall i \in \{1,\ldots,|\mathcal{S}|\}} \delta_i(t-1) P_{i \to j} \tag{3.39}$$

Starting in reverse order, i.e. at time step T, the final path is then recursively recovered by evaluating

$$\mathbf{s}_T^* = \arg\max_i \delta_T(i) \quad \mathbf{s}_t^* = \psi_{t+1}(\mathbf{s}_{t+1}^*) \tag{3.40}$$

The globally optimal state path $\mathbf{s}^*$ is only known when the observation sequence is considered in its entire length. This is disadvantageous in real-time applications where only a subset can be evaluated at any point in time. As a result the optimal path might vary when new observations are made.

Learning or parameter estimation refers to the process of iteratively optimizing initial estimates of transition, emission and start probabilities following an optimality criterion. The probabilities that describe these are extracted from training samples where the association between an observations and its the corresponding state are known. With respect to the above motion classification problem, learning can be described as the process of refining an initially estimated HMM to best fit the statistic relationships between poses and behavior types. Such an HMM is for example trained in chapter 4 to recognize user motions during human-robot cooperation tasks. It allows the robot to infer the current user behavior and react accordingly.

Over the years different methods emerged to train HMMs in such a data-driven fashion. Most prominent examples are the *Baum-Welch* algorithm, *Viterbi training* or *segmental k-means* and to some extent the brute force method *forward-backward algorithm*. In literature the Baum-Welch algorithm is the most common form of optimizing HMMs. It uses the total production probability $P(\mathbf{o}|\Theta)$ as an optimality measurement and optimizes model parameters based on example data $\mathbf{o}$. In more general terms it

can be seen as a variant of *expectation maximization*, which computes optimal parameters of a multi-stage stochastic process based on the maximum likelihood of data. The core idea of learning is to optimize a given or assumed model Θ so that its optimized parameters $\hat{\Theta}$ yield a greater output probability.

$$P(\mathbf{o}|\Theta) \leq P(\mathbf{o}|\hat{\Theta}) \tag{3.41}$$

The Baum-Welch algorithm is based on the assumption that the forward and backward variables allow inference of internal states given a sequence of observations. It requires the posterior probability $P(s_t = s_i|\mathbf{o}, \Theta)$ for the occurrence of state s_i at time step t and the probability $P(s_t = s_i, s_{t+1} = s_j|\mathbf{o}, \Theta)$ for transitioning from state s_i to s_j at time step t to $t+1$. The first probability is denoted $\gamma_t(i)$ and the second is indicated by $\gamma_t(i, j)$ [88].

$$\gamma_t(i) = P(s_t = s_i|\mathbf{o}, \Theta) = \frac{\alpha_t(i)\beta_t(i)}{P(\mathbf{o}|\Theta)} \tag{3.42}$$

$$\gamma_t(i, j) = P(s_t = s_i, s_{t+1} = s_j|\mathbf{o}, \Theta) = \frac{\alpha_t(i)P_{i\to j}p_j(\mathbf{o}_{t+1})\beta_{t+1}(j)}{P(\mathbf{o}|\Theta)} \tag{3.43}$$

In Equ. 3.43 $\alpha_t(i)$ denotes the posteriori probability, i.e. the probability that the model is in state s_i given the observations. $\beta_t(j)$ on the other hand is the backward variable representing the probability for generating $o_{t+1}, \ldots, o_T$ from time step $t + 1$ onward starting at state j

$$\beta_t(j) = P(o_{t+1}, \ldots, o_T|s_t = s_j, \Theta) \tag{3.44}$$

The numerator in Equ. 3.43 refers to the probability of generating output $\mathbf{o}$ under the constraint that a transition from state s_i to state s_j occurs at time step t. The model Θ is learned by replacing parameters with their respective conditional expected values. Optimized transition probabilities for example are obtained by computing the statistical average over individual transition probabilities $\gamma_t(i, j)$ for all points in time $t = 1, \ldots, T$.

$$\hat{P}_{i\to j} = \frac{\sum_{t=1}^{T-1} P(s_t = s_i, s_{t+1} = s_j|\mathbf{o}, \Theta)}{\sum_{t=1}^{T-1} P(s_t = s_i|\mathbf{o}, \Theta)} = \frac{\sum_{t=1}^{T-1} \gamma_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{3.45}$$

Optimized start probabilities are interpreted as special cases of transition probabilities and they are collected by evaluating $\hat{P}_i = P(s_{t=1} = s_i|\mathbf{o}, \Theta)$. The individual probabilities can be obtained in a similar vein.

Improved output probabilities are obtained by evaluating the expected number of outputs for a given state $s_t = j$ and the occurrence of the symbol o_k at o_t. Naturally, this number has to be normalized with respect to the total amount of symbols.

$$\hat{p}_j(o_k) = \frac{\sum_{t=1}^{T} P(s_t = s_j, o_t = o_k|\mathbf{o}, \Theta)}{\sum_{t=1}^{T} P(s_j|\mathbf{o}, \Theta)} = \frac{\sum_{t:o_t=o_k} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)} \tag{3.46}$$

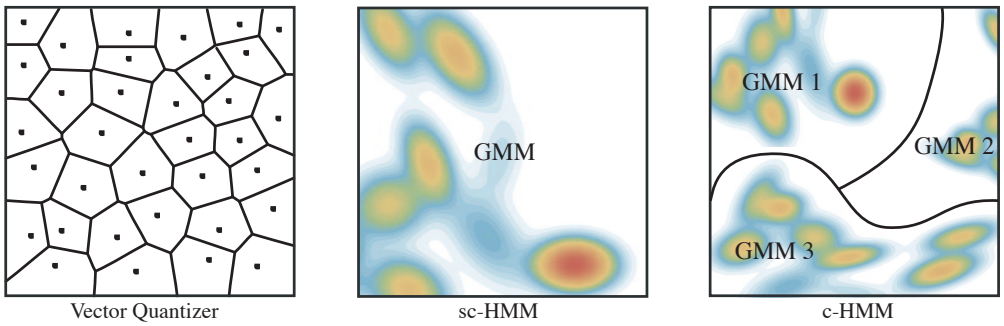


Fig. 3.6.: Different types of HMMs. Left: The parameter space is divided into partitions using a vector quantizer yielding discrete probabilities for each observation. Middle: Continuous densities model observation probabilities. Each state of the HMM has its own set of mixtures. Right: A shared GMM models the observation probability globally for all states of the HMM.

The resulting parameters of the HMM $\hat{\Theta} = (\mathcal{S}, \hat{P}_j, \hat{P}_{i \to j}, \hat{p}_i(o))$ exhibit at least the same the overall output probability $P(\mathbf{o}|\hat{\Theta})$ as the start solution Θ (see Equ. 3.41).

3.4.3. HMM Variants

Modeling emission probabilities in HMMs can be achieved using various distributions, depending on the problem at hand. Traditionally, discrete observation probabilities are assumed. Some applications however might require real valued observations in stead of symbolic inventories [87] due to complex relationships. Consider the aforementioned motion classification problem for example.

When the user's poses are tracked by the motion capture system it can be easily assumed that real-valued measurements are recorded.

There are two common alternatives to handle continuous observations. One option is to convert the multivariate observation vector into discrete univariate samples using clustering (see Fig. 3.6 left). In order to deal with arbitrary distributions of observations however *continuous* emission probabilities should be used, cf. see [88, chapter 5]. These model observation probabilities with continuous density functions.

Several realizations of such are reported in literature. The most simplistic continuous HMM variant is a Gaussian Hidden Markov Model (g-HMM). Here, the emission probability is modeled using a single Gaussian distribution at each state. Based on the assumption of a Gaussian distribution more complex probability density functions can be derived.

Continuous Hidden Markov Models (c-HMMs) for example employ a mixture of Gaussian in each state. They are able model complex observation patterns by varying mixture weights and component amounts (see Fig. 3.6 middle). Semi-continuous Hidden Markov Models (sc-HMMs) on the other hand use a single GMM for all states. For both variants the emission probability is inferred by accumulating over all components of the GMM. In order to train continuous HMMs the Gaussians are adapted to best fit the data. This naturally requires optimization of several parameters such as means, covariances and mixture weights. As a result, more training data is required in contrast to discrete HMMs.

In order to balance the amount of training data needed to learn a model with its modeling capabilities sc-HMM are often used literature, cf. see [88]. Tab. 3.1 compares the mentioned HMM variants with respect to their emission probabilities.

Tab. 3.1.: HMM variants ordered by their computational complexity (from least to most complex).

HMM Variant	Emission Probability	Observation Symbols
d-HMM	categorical	discrete
g-HMM	1 Gaussian per state	continuous
sc-HMM	1 GMM for all states	continuous
c-HMM	1 GMM per state	continuous

3.4.4. HMMs for Motion Capture Data

Training HMMs requires data sets that capture the statistical properties of the problem one seeks to model. Selecting appropriate features that resemble these in an optimal way is a crucial step towards reliable and robust models. Determining the most suitable features heuristically can be an error-prone and tedious task [88]. Adding additional features to an already existing model might seem suitable given a certain metric or characteristic, but it requires additional training data to learn the model. As this demand can often not be satisfied, modeling quality decreases. This situation is generally referred to as the *curse of dimensionality* [90]. In literature there is still an ongoing debate how to determine whether a feature is important and what insight it actually adds to the statistical properties of a given problem [88]. Nevertheless, there is a general consensus that improving data quality is an important step towards high-quality models. The key question is how data should be optimized, so that it is beneficial for model learning.

In the field of motion recognition, elbow markers often feature a strong correlation to hand makers, which is due to the human skeletal structure. Decorellating these by means of transformation into alternate representations, improves model quality as feature independence is a necessity [88, 91]. A common approach to do so is PCA [91–95]. As introduced in chapter 3.2.2, decorrelation of data is achieved by orienting coordinate axes so that they build a new orthogonal basis system maximizing the data's variance on each axis. In the context of real-time applications special restrictions are further burdened onto the computational demand of an HMM. During runtime the Viterbi algorithm correlates with the observation sequence length polynomially ($O(|S|T^2)$). As a result the required amount of computations increases with each additional observation and time step. As time progresses the amount of observation increases till a point where a Viterbi path can not be computed interactively anymore, i.e. the calculation of $\mathbf{s}^*$ requires more than 25 ms. This renders the algorithm unsuitable in real-time applications as the amount of observations increases continuously. It also considers the behavior of an HMM over a finite length of observations and their entirety. The

assertion of the final internal state can thus only be inferred once all observations are captured.

To improve on the aforementioned limitations different extensions have been proposed. In [96, 97] for example a sliding window approach is used to limit the Viterbi path length. Here, the key idea is keep a fixed amount of previous observations in memory and compute a Viterbi path accordingly. For the new path the start probability P_i of starting in state s_i is set to probability of reaching state i following the optimal path $\mathbf{s}^*$. However, the length of the window has to be set in an application-specific manner and requires manual fine tuning. A more general approach uses fusion points to merge Viterbi paths [96]. The algorithm increases the amount of observations constantly until all possible paths converge to a single point in the graph, the so called fusion point. The states prior to the fusion point are then considered decoded and a new path started at the fusion point.

3.4.5. Further Reading

There is an extensive line of research around the general concept of HMMs. The interested reader is encouraged to review Rabiners introduction [87]. Based on that [88] is a valuable starting point for a more recent view on pattern recognition.

3.5. Conclusion

In this chapter several representations of motion capture data have been introduced each offering benefits as well as limitations. Whereas Cartesian coordinates offer unfiltered and raw information about joint movements, their sheer dimensionality often prohibits direct utilization in learning algorithms. Representing motions in latent spaces on the other hand avoid that drawback by expressing information with fewer dimensions. This makes data more accessible for learning techniques since intrinsic details are uncovered. However, topological relationships such as neighborhoods or relational properties are not represented. This is where differential coordinates have proven advantageous. By expressing joint positions as a set of functions with respect to their neighbors, topological information is inherently captured. Harnessing this benefit IMs have been introduced to encode two-person motion capture data in differential coordinates. By using LME, an IM can be adapted to best fit an animators requirements while preserving spatial relationships, and thus, the shape of the motions.

HMMs are a well-established method for motion recognition. In the framework for two-person interaction models introduced in the next chapter, they can be used for classifying user motions in case of longer interaction scenarios consisting of several subtasks.

Combining the benefits of different data representations a learning from demonstration framework is developed in the following that enables virtual characters and robots to learn from human-human interactions. Spatial and temporal dynamics are thereby captured in a single interaction model.

4. Two-Person Interaction Models: *From Human-Human Demonstration to Human-Robot Interaction*

4.1. Introduction

Imitation learning has proven to be a valued robot programming paradigm and it is used in several scenarios (see chapter 2). In this chapter, a novel approach is introduced where this fundamental learning concept is applied to human-robot interactions where a great amount of additional difficulties need to be addressed. On top of learning motor skills required to execute an action, interactions depend upon continuous adaptation of one's behaviour during a joint task. They involve complex dynamics in each situation including temporal aspects of coordination and joint action understanding. Imitating these properties is unfeasible by focusing only on a single agent. Instead, they require consideration of both interactants to allow for mutual coordination.

Consider a high-five scenario for example. If the motions of both interactants were occurring independently, slight variations in the individual joint velocities and positions would instantly lead to changing hand positions and, most likely, to a failing interaction. Humans, however, naturally compensate for these muscular variations by adapting their individual behavior interdependently with the interaction's goal in mind, i.e. to clap the other's hand. Using perception, each interactant accommodates variations and adjusts to the respective interaction partner. Neuroscience studies of human-human interactions show that this is due to the shared mutual understanding of people [98, 99]. It is argued that humans tend to know how day-to-day interactions, such as high-fives, should take place and interactants become a coupled unit where moment-to-moment mutual adaptation takes place continuously [100]. This is one of the main reasons why humans are able to engage in a variety of situations so seamlessly and fluently. Removing this adaptation layer by blindfolding for example, condemns the interaction task to fail at once as the interactants have no means of synchronizing their behaviors. As a result, one can see the importance of continuous adaptation and joint temporal coordination.

Another key aspect that arises from the aforementioned mutual understanding is the ability to anticipate the behavior of an interaction partner. It enables humans

to recognize motions and engage in interactions early on without observing an action being fully carried out. In the high-five scenario for example, one interactant typically initiates the interaction by raising his hand, where as his partner recognizes the motion and reacts slightly delayed. Naturally, he will compensate for the delay by moving faster, so that both hands meet at the right time and position. Due to mutual understanding, high-five positions and relative timings are anticipated and the interaction can be carried out successfully.

Giving robots the ability to anticipate human behavior and adapt their own motion so they are able to collaborate with humans effortlessly, would clearly increase their interaction capabilities, smoothing the way for fluent and seamless human-robot teaming. Following that vision, **the aim of this chapter is to develop a methodology that enables robots to imitate coordinated actions by means of learning from demonstration.** For this however, two key challenges have to be addressed. First, the robot has to be programmed in such a way that it is able perceive and recognize user motions. Second, adequate responses need to be generated continuously to take the specifics of the current situation into account.

A prevalent approach to solving this task is to use machine learning algorithms, such as reinforcement learning or neural networks, to identify the intention of observed movements and then trigger a programmed behavior as a response, cf. chapter 2. This, unfortunately, does not scale when the number of interaction scenarios increases. The amount of training data that is required in order to learn tasks becomes increasingly prohibitive, since it is, for interactions in particular, often not available. In addition, continuous human-robot interactions require robot responses to be computed in a limited amount of time so that unnatural delays are avoided. This contrasts to single actor imitation learning that does not require online user perception and motion adaptation.

What all this amounts to, is a general need for efficient interaction learning algorithms satisfying the following requirements:

- An efficient training scheme, requiring only a few task demonstrations while allowing even unskilled users to train the robot
- Spatiotemporal adaptation of robot motions during the course of an human-robot collaboration
- Robot response generation at interactive rates to allow for instant responses

Addressing these requirements, the core of the proposed methodology is based on imitation learning of two-person interactions. One distinct feature is that the approach utilizes parallel behavior demonstrations of two human partners to learn a joint interaction model. This differs from other methods since correlations among low-level actions of *both* interaction partners are explicitly captured. The interaction model encodes how each interactant adapted his/her behavior during task demonstration (see Fig. 4.1). It encapsulates spatial relationships of joints and temporal properties of the collaborative

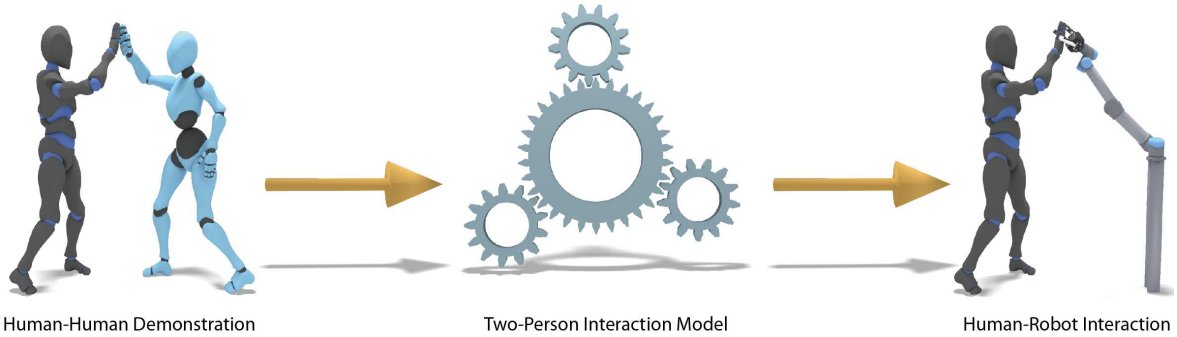


Fig. 4.1.: Based on parallel behavior demonstrations, a two-person interaction model is learned that captures how each interactant moved during task demonstration. Using the model at runtime a robot is able to adapt its behavior during a joint human-robot task while producing seamless and smooth responses.

behavior independently of the robot structure. In the end, inter-personal (between-person) dynamics of the interaction and intra-personal (within-person) motor skills are unified in a central model.

Utilizing the model during human-robot interaction, robot motions are produced efficiently while being spatiotemporally adapted to the current situation. In doing so, continuous human-aware responses are generated and complex sequences of joint actions are executed.

4.2. Methodology

Similar to the imitation of a motor skill, imitating human-human interactions roots upon observation, representation and reproduction. However, in contrast to the imitation of a single-person behavior, imitation of human-human interactions requires human-aware adaptation during runtime, in order to ensure the safety of the human collaboration partner as well as the success of the joint task. Towards that end, interaction learning as presented in this thesis, is composed of an *offline learning* and *online adaptation* phase as illustrated in Fig. 4.2.

During training, an interaction model is learned that describes how the two interactants synchronize their movements. First *human-human* demonstrations of two users performing cooperative tasks are recorded using motion capture. Here, several tasks can be demonstrated but only one example demonstration per task is necessary. Using the recorded demonstrations an interaction model is learned, which encodes spatial and temporal information of the parallel behavior demonstration.

At runtime, the model is used to continuously adapt the robot's movements to that of the human interaction partner. Generally, a leader-follower type scenario is assumed, where one person acts as an assistant. During human-robot interaction, the robot will assume the role of the assistant. For notational clarity, following [54] the first interaction partner, i.e. the human, is referred to as the *observed agent*, while the second interaction partner, i.e. the robot, will be called *controlled agent*.

The proposed methodology builds upon previous work from computer animation

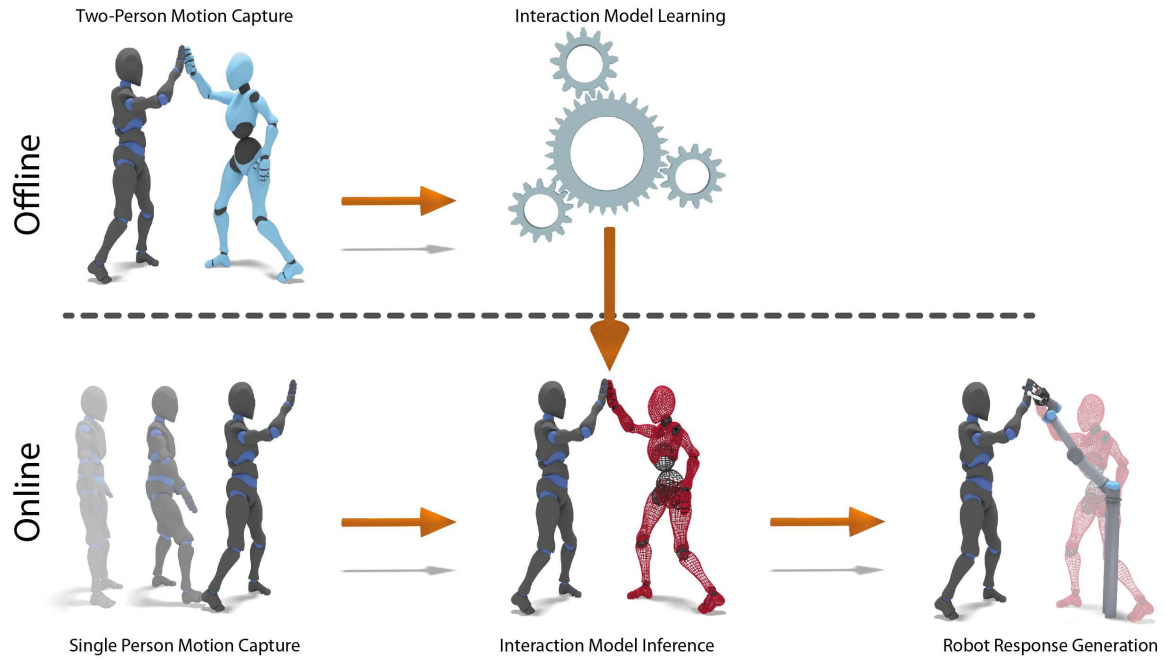


Fig. 4.2.: The proposed methodology is composed of an offline learning and an online adaptation phase. Using the learned interaction model robot poses are continuously inferred based on the current user motion and adapted at each frame to match the situation.

and provides a significantly extended version of the initial IM approach [39] for spatial adaptation of interactions. Here, instead of manually adjusting the topology and constraint settings of the IM at each frame, two-person task demonstrations are used to automatically extract relevant parameters.

The temporal properties of the demonstration are captured in low-dimensional spaces, varying IM constraints and an optional HMM. Previous work has shown that HMMs are well suited for modeling sequences of data [97, 101, 102]. However, recent results also indicate that models based on HMMs alone do not generalize sufficiently to postural changes in typical human-robot interaction tasks [54]. As a result, further optimizations to the robot's posture and movement are required to ensure efficient and safe physical interaction.

Using posture spaces for estimating the state of an interaction and IMs for spatial adaptation, an interaction model, learned from two-person task demonstrations is introduced in the following. Processes involved in the creation are discussed in detail and application examples in the field of character animation (see chapter 5), human-robot collaboration (see chapter 6) and triadic human-robot handovers (see chapter 7) are provided subsequently.

4.3. Learning Two-Person Interaction Models

The interaction model presented in this section serves to generate a controlled agent's response to the movement of the observed agent in cooperative tasks. From a methodological point of view, the model provides a means of estimating the state of an ongoing

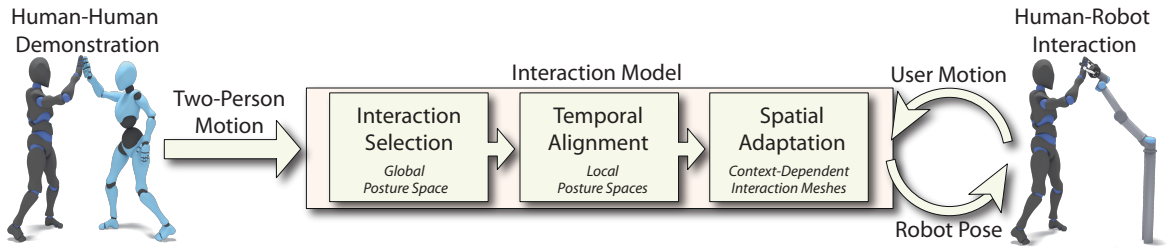


Fig. 4.3.: Based on human-human demonstrations an interaction model is learned that captures how each person moved during the joint execution of a physical collaboration. Using the learned model in a human-robot interaction the robot's role is inferred and controls are computed continuously. The interaction is first derived and then temporally aligned locally in low-dimensional space. Spatial adaptation based on matched interaction demonstrations is achieved using context dependent IMs.

interaction and spatiotemporal adaptation of behavior demonstrations.

Structurally, the interaction model consists of the following components (see Fig. 4.3):

- a *global posture space* for state estimation,
- a set of *local posture spaces* for temporal alignment
- and, a database of *context-dependent IMs* for spatial adaptation.

During model learning the global posture space is computed based on all demonstrations. Then, for each interaction a low-dimensional local posture space is defined and further segmented into smaller parts before finally an IM is constructed for each motion capture frame. The posture spaces allow the selection of the best matching IM during an ongoing human-robot interaction. Each IM represents a pair of postures of the human-human demonstration at a single time step and it captures the spatial relationships of the interaction. A context-based variant of IMs is proposed that focuses on the most relevant joints of the two human demonstrators in order to increase postural generalization.

At runtime, a suitable human-human interaction is first selected in global posture space and then temporally aligned in the matching local posture space using DTW. Then, a robot's pose is optimized to best-fit the current situation using the matching IM. Subsequently, details of the various components of the interaction model are explained in more detail.

4.3.1. Data Acquisition: Human-Human Demonstration

Imitating motor skills requires the perception of a demonstrator's motion. Interaction data for HRI is typically gathered using kinesthetic teaching and single-person motion capture, e.g. in [45, 53, 103]. In kinesthetic teaching the robot's joints are manually operated by a user and its joint angles are stored in intervals. At the same time, the human interaction partner is recorded using motion capture. In the end, the data of both interactants is represented independently in separate datasets. This hinders optimization of a robot's pose since motions have to be transformed into a joint task-space in order to be accessible for learning techniques [2]. Also, operating the robot's

joints manually has the potential of being physically demanding and, thus, unfeasible even for moderately sized robots.

Instead of manipulating the robot's joints manually interaction data can also be gathered using two-person motion capture. To differentiate between the interactants professional tracking systems often build upon markers attached to the interactants' bodies. They capture joints and their movements with very high accuracies (<1 mm) for up to 15 persons simultaneously [22]. While these systems found widespread acceptance in movie production environments, they are still expensive and cumbersome to setup.

Advances in sensing technologies however gave rise to consumer-grade capturing devices. Depth sensors such as the Kinect camera gained ubiquitous distribution in robotics research due to their ease of use and low acquisition costs. They track users based on a time-of-flight sensor and, thus, do not rely on tracking markers. A benefit of these devices is that people can be recorded unobtrusively in their natural unconstrained environment. This releases the burden of wearing markers and favors the ease with which new motions can be recorded. They are on the other not as accurate and often suffer from line of sight problems [104].

Nevertheless, consumer-grade as well as professional motion capture systems offer advantages over kinesthetic teaching and single actor tracking. One eminent feature is the smoothness of the recordings. Whereas kinesthetic teaching can lead to jerky data sets, motion capture creates inherently smooth recordings for both interactants. But more importantly, it captures users in a joint coordinate frame and, thus, circumvents the need for additional transformations. The resulting recordings are particularly suitable for optimization techniques since they offer a great amount of insight into the interperson dynamics of interactants. Among others, body synchrony, spatial constraints and temporal relationships are readily available for learning. Also, kinesthetically trained goal-directed behaviors, as mainly focused on in motion planning research, are often considered unpredictable and non-engaging for humans. Surveys indicate that this is mainly due to users' not being confident about the robot's intent [18]. Using two-person motion capture minimizes these effects by creating inherently natural and human-like robot motions, allowing collaborators to quickly infer a robot's goal. It also has the advantage of being physically undemanding whilst allowing both interactants to move freely. Two-person motion capture harnesses the human's natural ability of demonstrating tasks unobtrusively since no robot has to be manipulated during task demonstration. This results, from a psychological and neuroscientific point of view, in more efficient communication and knowledge transfer between the interactants. Studies suggest that, information of a specific task is preferably transferred by demonstration and, when imitated by a robot, more understandable robot motions are created [15]. This benefits human-robot collaborations as more fluent physical interactions emerge.

Mathematically, a motion capture recording is a time series of postures at T time steps, where each posture is a vector of joint positions or motion capture markers

$$\mathbf{p} = (x_1, \ldots, x_N, y_1, \ldots, y_N, z_1, \ldots, z_N) \quad x, y, z \in \mathbb{R} \tag{4.1}$$

$$\mathcal{R} = \mathbf{p}_{1:T} = [\mathbf{p}_1, \ldots, \mathbf{p}_t, \ldots, \mathbf{p}_T] \quad t \in \{1, \ldots, T\} \tag{4.2}$$

A two-person recording is defined as a pair of single actor recordings in a global coordinate system. N denotes the amount of joints that are tracked for each interactant. It is noted that the interaction learning approach presented in this thesis does not assume a specific tracking hardware and it has been tested successfully with optical tracking systems (A.R.T. DTrack), time of flight cameras (Kinect One) as well as structured light depth sensors (Kinect 360/Asus Xtion). Examples using different tracking system are presented in the subsequent chapters. However, due to the number of available systems and the resulting variety of skeletal configurations an intermediate kinematic structure is assumed. Joints that contribute most to a behavior including feet, hands, elbows, head and torso are attributed a motion capture marker. Using this mapping scheme benefits interoperability as it is also used in the majority of motion capture frameworks, e.g. *OpenNI*, *Kinect SDK* and *Motion Builder*. Joints such as knees, vertebras and wrists, i.e. links that are not captured, are reconstructed using IK. For that each marker is linked to a IK controller to drive the subsequent reconstruction. The interested reader is referred to [105] for a detailed introduction into IK.

Using the recording of two demonstrators a model of the joint task demonstration is learned as described in the following.

4.3.2. Global Posture Space

An integral part of an interaction model is the ability to select an interaction recording during HRI. It is based on the assumption that human motion intrinsically lies on low-dimensional manifolds and that it can be represented with fewer dimensions. To reduce the dimensionality of motion capture data PCA is employed (see chapter 3.2). It defines a set of principal components that maximize the variance over joint motions and consequently their importances along an axis. The linear subspace that is created allows traditional metrics such as Euclidean distances to remain applicable. Using these and the fact that a point in low-dimensional space corresponds to a pose, differences in the posture space resemble postural variations in Cartesian space. As a result, similar human poses lead to close points in the low-dimensional embedding. In the end, PCA not only reduces the search space's complexity but also decreases the required amount of memory to store motions.

A latent embedding for all motions of the observed agent $(\cdot)^{oa}$ is computed and denoted *global posture space* $\mathcal{G}$. It captures motions executed during all task demonstrations in a single posture space and it provides a compact search space for estimating

the state of the interaction during runtime.

$$\mathcal{R}^{oa} \xmapsto{PCA} \mathcal{G}, \quad |\mathcal{R}^{oa}| = |\mathcal{G}| \tag{4.3}$$

$$H(\mathcal{R}^{oa}) = H(\mathcal{G}) + \epsilon \tag{4.4}$$

$\mathcal{G}$ represents user motions with fewer dimensions while retaining the amount of information within (denoted by H). DR is achieved at the expense of information lost ϵ. The balance, i.e. the amount of latent dimensions, is to be set depending on the problem at hand. As general rule of thumb, motion capture recordings with 8 markers, yielding a 24-dimensional state space, require on average $3-6$ dimensions to provide reliable results ($\epsilon < 0.05$).

The global posture space might be composed of interaction demonstrations that share similar poses among them. These situations often occur when users retract to a rest pose during motion capture. As a result, the recordings feature similar start and end points in low-dimensional space. Ambiguities might arise when similar poses are also adopted during the course of different interactions. A single posture by itself might be part of several recordings and, thus, sequential information is necessary in order to distinguish between interaction candidates. To improve state estimation in these situations the global posture space can be enriched with an HMM (see section 4.4.1). Here, key poses are extracted and sequences of such can be used during runtime to infer an interaction demonstration.

4.3.3. Local Posture Spaces

The global posture space captures spatial information of the interaction demonstrations. In order to account for temporal variations and changes in joint importances, an interaction model includes several *local posture spaces* $\mathcal{L}_i$. A local posture space is computed for each interaction demonstration and it allows fine grained temporal adaptation during runtime (see chapter 4.4). It also retains subtle details and fine nuances of motions that are not well preserved in $\mathcal{G}$.

Consider the high-five scenario for example. The leading hand is of high importance since it steers the interaction. In contrast, feet balance the user on the ground, and their motion is considered less important, hence, a lower importance would be assigned to them. The relevance of maintaining certain spatial relationships between joints may however change during an interaction, e.g. when switching the leading hand. If the motion would be considered as a whole, alternations in the relationship of correlating joints would not be taken into account. To capture these shifts in importance during different phases of the interaction, each local posture space $\mathcal{L}_i$ is segmented using Hotellings T-squared statistics. A segment $\mathcal{Q}_k$ with $k \in \{1, \ldots, K\}$ is defined as a sequence of consecutive points $\mathbf{p}^{oa}_{r:v}$ and denoted by

$$\mathcal{Q}_m = \mathbf{p}^{oa}_{r:v} = [\mathbf{p}^{oa}_{r}, \mathbf{p}^{oa}_{r+1}, \ldots, \mathbf{p}^{oa}_{v}] \tag{4.5}$$

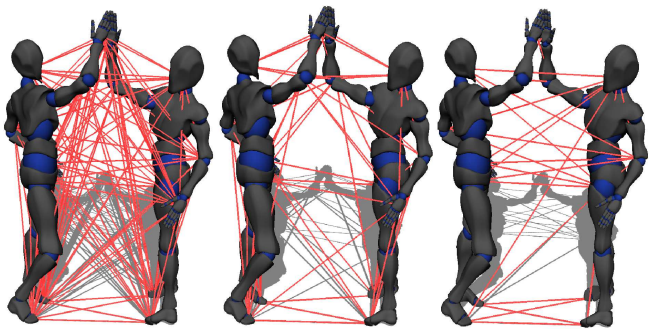


Fig. 4.4.: Left: An IM computed using a fully connected graph. Middle: An IM created using Delaunay triangulation. Right: In the proposed context-dependent IMs connections are added based on joint correlations in motion capture recordings. As it can be seen sparse topologies are created which allow for more joint movement during optimization and, consequently, increasing postural generalization.

with $r, v \in \{1, \ldots, T\}$ and $r \leq v$. The K segments are created by finding a set of sub-principal components that minimize the cost function

$$\Phi_m = \frac{1}{v - r + 1} \sum_{j=r}^{v} \mathbf{p}_j \mathbf{p}_j^T \quad , \; \mathbf{p}_j \in \mathcal{Q}_k. \tag{4.6}$$

In addition to allowing for varying joint correlations (and, thus, enabling the generation of IMs with varying topologies and constraint weights, see chapter 4.3.4), the segmentation of local posture spaces has the advantage of lowering the computational load of performing optimizations. On average $2 - 4$ dimensions provide enough information to reliably account for temporal and spatial variations of user motions.

4.3.4. Context-based Interaction Meshes

In order to achieve spatial generalization of the controlled agent's response a novel variant of IMs is developed. IMs have been extensively used in the computer animation community for adaptation of motion capture data [41, 57]. One eminent feature of IMs is the ability to adapt *full body* behaviors to new situations. So far, however, proposed methods rely on fully connected graphs [57] or Delaunay tetrahedralization [41] for net generation. These approaches include all joints equally into the topology yielding densely interconnected nets as shown in Fig. 4.4, left. Also, additional vertices are sampled on the skeleton's surface and increase the overall amount further, cf. [39]. Since the computational complexity of IM adaption significantly increases with larger numbers of vertices, only a few vertices should be used to ensure optimal response times during runtime. Moreover, joint weights and optimization constraints such as foot or hand contacts are usually modeled manually, thus requiring intervention of a human editor in the IM generation process. Furthermore, dense connection structures hinder joint movement during optimization, whereas manual modification of constraints and weights can be labor-intensive and error prone.

To improve on these limitations the following extensions to IMs are proposed:

- context-based topology generation at frame level for varying correlation structures and sparse marker setups
- data-driven optimization of weights to avoid manual labeling, and,
- an algorithm for automatic soft- and hard constraint generation based on motion capture data.

In the proposed methodology, an IM provides a topological and spatial representation of two humans during a motion capture recording at each time step. An IM topology is constructed using the Cartesian coordinates $\mathbf{p}_t^{oa,u_{min}}$ and $\mathbf{p}_t^{ca,l_{min}}$ of the closest pairs of captured joints u and l at each timestep t

$$u_t^{min}, l_t^{min} = \arg\min_{(u,l)} \left\| \mathbf{p}_t^{oa,u} - \mathbf{p}_t^{ca,l} \right\| \tag{4.7}$$

and their correlating neighboring markers j_1, j_2 of the *controlled* agent ($(\cdot)^{ca}$).

$$\begin{aligned} j_1 &= \arg\max_{j} \operatorname{corr}(\mathbf{p}^{l^{min}}, \mathbf{p}^{ca,j}) \\ j_2 &= \arg\max_{j \neq j_1} \operatorname{corr}(\mathbf{p}^{l^{min}}, \mathbf{p}^{ca,j}) \end{aligned} \tag{4.8}$$

Joints that exhibit correlation and are in close proximity are connected through a tetrahedron. A tetrahedron T_t is defined as a tuple of connected vertices $T_t = (u^{min}, l^{min}, j_1, j_2)$ where u is an index into the joint table of the observed agent M and l,j_1 and j_2 correspond to the controlled agent's joints N.

The topology $\mathcal{T}_t$ of an IM at time step t is consequently defined as a set of tetrahedra $\mathcal{T}_t = \{T_t^1, \ldots, T_t^P\}$, where P denotes the amount of created tetrahedra. The process of adding tetrahedra to the IM is repeated, with previously used joint pairs being excluded from consideration, as long as a pair with correlation above threshold Ψ exists. A good choice of Ψ depends on the velocity of the motion and the frame rate with which the demonstration has been captured. Also Ψ has a strong influence on the density of the topology. As dense connection structures hinder mesh deformation during optimization sparse topologies are desired. IM topologies should however not be too sparse as too sparse connections structures may fail to preserve spatial relationships to a reasonable degree.

To allow for varying joint relationships during runtime a correlation-based weight

$$w^{ca,i=1\ldots N} = \begin{cases} 1 - \frac{\operatorname{cov}(\mathbf{p}^{ca,i}, \mathbf{p}^{oa,j})}{\sigma(\mathbf{p}^{ca,i})\sigma(\mathbf{p}^{oa,j})} & \text{if } \forall j \ \operatorname{corr}(\mathbf{p}^{ca,i}, \mathbf{p}^{ca,j}) < \Psi \\ 0 & \text{otherwise} \end{cases} \tag{4.9}$$

is computed for each joint in the segment $\mathcal{Q}_k$ (see Equ. 4.5). $\sigma^2(\mathbf{p}_k^{ca})$ denotes the variance over poses $\mathbf{p}_k^{ca}$ of segment $\mathcal{Q}_k$ in Cartesian coordinates. Using the weights, each

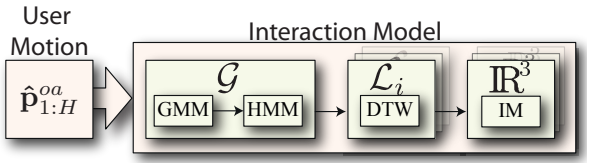


Fig. 4.5.: During runtime the observed agent's motion $\hat{\mathbf{p}}^{oa}_{1:H}$ is projected into the previously created global posture space $\mathcal{G}$. Then, the corresponding local posture space $\mathcal{L}_i$ is retrieved (using the HMM for example) and used to match the observed agent's motion onto motion segments. This yields a point from the initial demonstration that best fits the current situation temporally. To adapt the found reference pose spatially to the new situation the associated IM is optimized.

marker of the controlled agent that does not exhibit a strong correlation is automatically assigned a soft positional constraint according to its weights in each segment $\mathcal{Q}_k$.

$$f^{ca,i=1\ldots N} = \begin{cases} 1 & \text{if } \forall j \ \text{corr}(\mathbf{p}^{ca,i}, \mathbf{p}^{ca,j}) < \Psi \\ 0 & \text{otherwise} \end{cases} \tag{4.10}$$

The soft constraint weights in Equ. 4.9 range from 0 to 1. A weight close to 1 indicates a strong tendency to adapt towards the initial demonstration. Weights close to 0 allow for stronger adaptation towards the current situation. Also, a hard positional constraint c^{ca} is added for each joint of the observed agent, as the observed agent's pose is not subject to optimization.

$$c^{oa,i=1\ldots M} = 1 \tag{4.11}$$

In essence, hard constraints are added to the observed agent and soft constraints to the controlled agent to model the amount of adaptation, depending on joint correlations during motion capture. This allows the preservation of body synchrony as well as spatial relationships of the most important joints.

4.4. Computing Responses for the Controlled Agent

To compute an agent's response to an observed user motion, the interaction model is used threefold. First, it serves to identify a suitable interaction demonstration from the pool of all recorded interactions using the global posture space $\mathcal{G}$. Second, it allows temporal alignment of the user's motion to the matching interaction $\mathcal{L}_i$. This yields a time frame and a set of poses from the initial recording that best fits the current situation temporally as well as spatially. And third, the inferred pair of poses from the demonstration are spatially adapted to the ongoing human-agent interaction using an context-dependent IM (see Fig. 4.5).

In the following, the steps required to use an interaction model for spatiotemporal adaptation during runtime are introduced in detail.

4.4.1. Interaction Selection

To infer a suitable interaction demonstration from the pool of all recorded interactions the global posture space is used. For that each live pose of the observed agent is reduced in dimensionality by projecting it into $\mathcal{G}$, creating new points $\hat{\mathbf{p}}^{oa}_{H-S:H} = [\hat{\mathbf{p}}^{oa}_{H-S}, \hat{\mathbf{p}}^{oa}_{H-(S-1)}, \ldots, \hat{\mathbf{p}}^{oa}_{H}]$ ($\hat{\mathbf{p}}^{oa} \in \mathcal{G}$). H is an index to the most recent pose in the sliding window of evaluated poses and S denotes the amount of poses in the sliding window. Then the state of the ongoing interaction is estimated, i.e. a suitable interaction demonstration and, thus, a local posture space $\mathcal{L}_i$ is selected.

Depending on the scenario at hand, some interactions require sequential orders of subtasks, such as collaborative assemblies for example. For that temporal information and past user poses have to be taken into account so that the current state of the ongoing interaction can be estimated reliably. However, pre-imposing a sequential plan might also be counterproductive. Settings where users decide what action to take next like in competitive games for example (see Chapter 5) are not based on preprogrammed action sequences. In order to consider both scenarios, two interaction selection approaches are proposed. The first method utilizes postural similarities in global posture space, i.e. Euclidean distances, to select a demonstration on a per-frame basis. The second approach on the other hand explicitly includes sequences of previous user poses in an HMM to account for temporal coherences and sequential orders of tasks. Both interaction selection methods are described in the following.

Distance-based Interaction Selection

In order to estimate the state of an ongoing interaction based on postural similarities live user poses are compared to recorded demonstrations. For that a distance matrix $\mathbf{E}$ is generated and evaluated for a sliding window of S timesteps. Its elements capture the distances $\mathbf{d}_i$, $i \in \{1, \ldots, 1, \ldots, I\}$ between $\mathbf{p}^{oa} \in \mathcal{G}$ and the closest segment centroid of each interaction i

$$\mathbf{E} = (\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_i, \ldots \mathbf{d}_I) \ , \ \mathbf{d}_i = (d_{i,1}, \ldots, d_{i,S})^T \tag{4.12a}$$

Each row in $\mathbf{E}$ stores the distances of the user's motion to all segment centroids at each frame of the sliding window. In the end, the column with the lowest mean value over all S timesteps identifies the most suitable interaction demonstration and, thus, the local posture space for temporal alignment.

$$i = \underset{1 \leq j \leq I}{\arg\min} (\sum_{k=1}^{S} \mathbf{E}^{k,j}) \tag{4.13}$$

Using several time steps to estimate the state of the interaction limits the effect of undesired shifts between potential interaction candidates and increases hysteresis, i.e. the controlled agent's commitment to the current interaction. However, situations might arise in which the correct interaction example can only be determined when

larger temporal context and sequences of key poses are accounted for. Towards that end, a HMM-based interaction selection approach is presented in the following.

HMM-based Interaction Selection

The first step towards the creation of the HMM is to identify key poses in the human-human interaction demonstrations. Key poses have a long history in computer animation and robot control [106, 107]. They are typically defined for the entire robot/character and a skilled programmer/animator is often required to model them over time. Recently statistical approaches, such as in Bashir et al. [106], have been proposed to model human motions by estimating their probability density using Gaussians. One benefit that arises from this methodology is that the mean of each distribution can be interpreted as a *key pose* of the underlying motion [108]. However, whereas current approaches use high-dimensional data to extract key poses, interaction models harness the global posture space and, in doing require fewer dimensions. This also forces key poses to focus on joints that contribute to the motion while at the same time reducing the amount of data required to extract the statistical properties accurately. The global posture space also removes redundancy in motion capture data and maximizes the covariance over each latent dimension. The resulting statistical independence is of importance for estimating key poses. Since the covariance matrix of the density estimate features less correlation elements, i.e. non-zero diagonal elements, less components are required to approximate the data. Also the dimensionality of the covariance matrix decreases which reduces the computational load during key pose estimation.

The density of recorded motion trajectories is approximated in the global posture space $\mathcal{G}$ by performing Kernel Density Estimation (KDE). Following the approach of [109], KDE places a Gaussian kernel over each point in the low-dimensional space, thereby reconstructing the probability density function. Using moment-matching, a more compact GMM is extracted which automatically determines the number R of Gaussian needed to represent the data. The KDE yields a GMM for all poses $\mathbf{p}^{oa}_{1:T}$ in latent space $\mathcal{G}$. It is is defined as a sum of R Gaussians $\mathcal{N}(\mathbf{p}^{oa}_{1:T}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ weighted by w

$$f(\mathbf{p}^{oa}_{1:T}) = \sum_{i=1}^{R} w_i \mathcal{N}(\mathbf{p}^{oa}_{1:T}|\boldsymbol{\mu}_i, \mathbf{\Sigma_i}) \quad , \quad \mathbf{p}^{oa}_{1:T} \in \mathcal{G} \tag{4.14}$$

$$\mathcal{N}(\mathbf{p}^{oa}_{1:T}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \frac{1}{\sqrt{2\pi|\boldsymbol{\Sigma}_i|}} \exp(-\frac{1}{2}(\mathbf{p}^{oa}_{1:T} - \boldsymbol{\mu}_i)^T \Sigma_i^{-1}(\mathbf{p}^{oa}_{1:T} - \boldsymbol{\mu}_i)) \tag{4.15}$$

Since the mean of each Gaussian is a key pose, sequences of such encode the temporal properties of the interaction demonstration as illustrated for three interactions in Fig. 4.6 (left). Using the extracted distributions from the GMM, a discrete HMM is constructed to model the probability distribution over observed key pose sequences. The GMM acts as the quantizer for continuous observations and provides the codebook for the HMM. This is a benefit over sc-HMMs and c-HMMs where manual fine-tuning

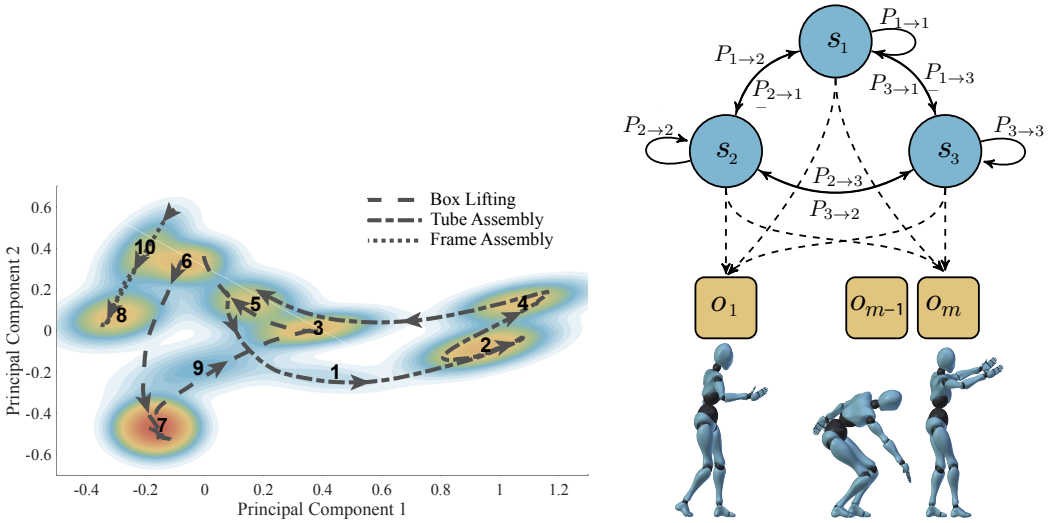


Fig. 4.6.: Left: A global posture space is created by applying PCA to the observed agents motion capture data. The resulting motion trajectories in latent space as well as the computed kernel density estimate is visualized. Color is used to indicate the probability of each distribution.
Right: A HMM is trained to capture the broad context of an interaction based on recorded motions in latent space. The Gaussian distributions of the kernel density estimate act as a vector quantizer and a hidden state s_i is defined for each interaction subtaks.

of the number of Gaussians of the emission distribution is required. In an interaction model each hidden state of the HMM corresponds to an interaction demonstration and is consequently assigned to a local posture space $\mathcal{L}_i$. The elements of the transition matrix thus define the probability of transitioning from one interaction to another. The emission probability distribution is estimated using the KDE and the probability of observing a key pose while in state s_i. The Gaussian with the largest posterior probability is the observation o of the HMM at time step t:

$$n = \arg\max_{\forall i} \mathcal{N}_i(\mathbf{p}_t^{oa} | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \quad , n, i \in \{1, \ldots, R\}. \tag{4.16}$$

The transition and emission probabilities are estimated using sequences of key poses, i.e. a vector of indices n. It is noted that additional motion capture recordings can be utilized to train the HMM but only a single demonstration is required. For training, a unique list of key poses is created by evaluating Equ. 4.16 for each motion capture frame. An estimate of the transition matrix $P_{i\to j}$ is generated by counting the transitions between interactions in the training data set, i.e. the transition from interaction s_i to s_j. In the end, the learned HMM encodes sequences of key poses and, consequently, generalizes their temporal relationships on a sequential level.

Evaluating the posterior probability of each Gaussian distribution per frame yields a set of key poses that resemble the past S user poses (denoted $\mathbf{o}_{1:S}$, see Equ. 4.16 and Fig. 4.7). Given the key poses, the interaction demonstration that matches the current situation best can be inferred by computing the posterior state probabilities of the HMM.

Since each hidden state corresponds to an interaction, the corresponding local pos-

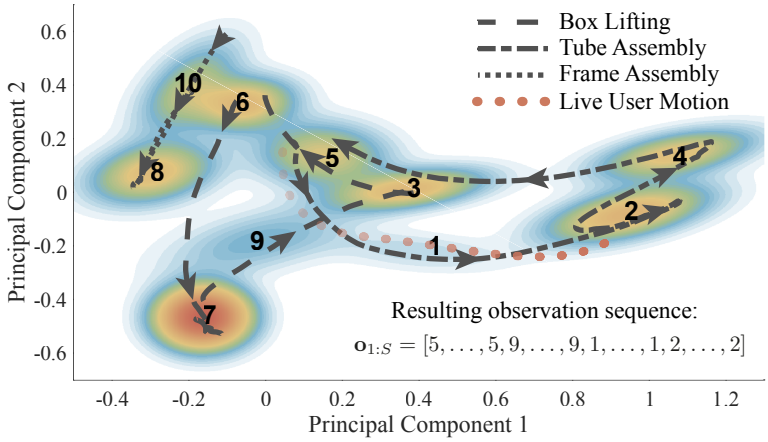


Fig. 4.7.: The live user motion is recorded and projected into the global posture space $\mathcal{G}$ leading to new points in the embedding (depicted red). By computing the posterior probability of each Gaussian of the GMM for all user poses a unique sequence of observations is created.

ture space can be retrieved. Using the HMM, the most suitable candidate is computed without explicit temporal information. Instead, sequential information, in terms of ordered key poses, is used to to infer a matching demonstration. This renders the selection of an interaction independent from the speed with which it was performed during motion capture. To account for additional temporal perturbations and variations during the course of a interaction, temporal alignment of live motions is enforced in the selected local posture space. The required steps are introduced in the following.

4.4.2. Temporal Alignment

In order to temporally align user motions to a specific interaction, poses $\hat{\mathbf{p}}^{oa}_{H-S:H}$ are projected into the local posture space of the selected interaction $\mathcal{L}_i$ and matched against the mean of all segments $\mathcal{Q}_{1:M}$ using Euclidean distances. The segment $\hat{\mathcal{Q}}_m$ that exhibits the smallest distance to $\hat{\mathbf{p}}^{oa}_{H-S:H}$ is on average the most similar motion of the initial recording and the involved poses resemble the current situation best. Temporal perturbations of the user's motions are accounted for by aligning the captured poses $\hat{\mathbf{p}}^{oa}_{H-S:H}$ to the matched segment $\hat{\mathcal{Q}}_m$. For that, a DTW path [110] between $\hat{\mathcal{Q}}_m$ and $\hat{\mathbf{p}}^{oa}_{H-S:H}$ is computed. The overall goal of DTW is to generate a warping path along poses that minimizes the sum of distances between the two motions, i.e. $\hat{\mathcal{Q}}_m$ and $\hat{\mathbf{p}}^{oa}_{H-S:H}$.

An $n \times m$ cost matrix $\mathbf{D}$ is created where each element $\mathbf{D}^{i,j}$ corresponds to the squared distance of the ith element of $\mathcal{Q}_m$ and jth element of $\hat{\mathbf{p}}_{H-S:H}$ respectively. Given the cost matrix, the optimal path is the path that minimizes $DTW(\mathcal{Q}_m, \hat{\mathbf{p}}_{H-S:H}) = \min(\sqrt{\sum_{k=1}^{K} w_k})$. Here w_k is the matrix element $\mathbf{D}^{i,j}_k$ that corresponds to the kth element of the warping path [111]. The end of the optimal path yields an index $\hat{t}$ to the motion capture frame of the initial recording that best fits the current situation temporally and spatially (see Fig. 4.8).

Computing the warping path however is computationally expensive due to its quadratic time and space complexity ($O(n^2)$). As DTW is applied in low-dimensional

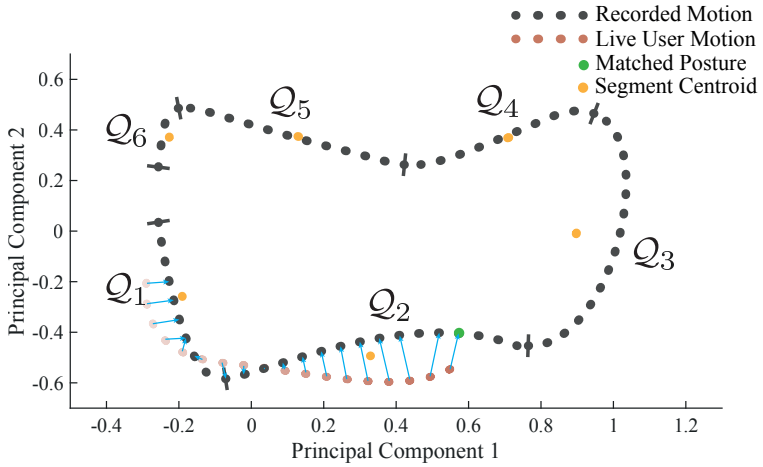


Fig. 4.8.: During runtime the user's motion is matched against an inferred interaction demonstration in local posture space $\mathcal{L}_i$. The figure depicts the optimal path that minimizes the sum of distances between the recording (gray dots) and the live motion (red dots). The end the path is highlighted green. It is associated with pair of poses from the initial recording that will be used to adapt an agents response spatially. Due to the computational complexity, only segments whose centroid (orange dots) are in close proximity are considered for DTW.

space the computational load is reduced. Also, a path is only computed for parts of the recorded motion, i.e. for segment $\mathcal{Q}_m$ and $\hat{\mathbf{p}}_{H-S:H}$, rather than the complete trajectory in $\mathcal{L}_i$. The optimizations above render it applicable in the considered real-time applications and allows temporal alignment of user motions during the course of an interaction. Yet, the user's motion still varies with respect to the initial recording and adopting the found pose directly will most likely fail to preserve the interaction's intent. As a result additional spatial optimization is required before the pose can be transfered to the robot. For spatial adaptation, context-dependent IMs are used in the following.

4.4.3. Spatial Adaptation

After inferring an interaction demonstration and temporally aligning the user's motion, the most suitable time frame the initial recording is selected. Given $\hat{t}$, an associated pair of poses $\mathbf{p}_{\hat{t}} = [\mathbf{p}^{oa}_{\hat{t}},\ \mathbf{p}^{ca}_{\hat{t}}]$, the corresponding IM topology $\mathcal{T}_{\hat{t}}$, the weights $w^{ca,i=1,\dots,N}_{\hat{t}}$ as well as the constraints $f^{ca,i=1,\dots,N}, c^{oa,i=1,\dots,M}$ are retrieved. Consider the difference between poses $\mathbf{p}_{\hat{t}}$ from the training recording with the poses in the current situation $\hat{\mathbf{p}}_H = [\hat{\mathbf{p}}^{oa}_H, \hat{\mathbf{p}}^{ca}_{current}]$, where $\hat{\mathbf{p}}^{oa}_H$ and $\hat{\mathbf{p}}^{ca}_{current}$ are the postures of the observed and controlled agent in Cartesian space. In order to adapt the retrieved IM to the current situation, essentially, its deformation energy is minimized

$$\min_{\hat{\mathbf{p}}_H} \sum_{i=1}^{3(M+N)} \frac{1}{2}\|L(\hat{\mathbf{p}}^i_H) - L(\mathbf{p}^i_{\hat{t}})\|^2 + \sum_{i=1}^{3(M+N)} \mathbf{W}^{i,i}_{\hat{t}}\|\hat{\mathbf{p}}^i_H - \mathbf{p}^i_{\hat{t}}\|^2 \tag{4.17}$$

while at the same time ensuring the validity of its associated constraints (right side of Equ. 4.17). L denotes the Laplacian operator which deforms poses into local coor-

dinates using the topology $\mathcal{T}_{\hat{t}}$ (see chapter 3.3.3). Equ. 4.17 is then reformulated to a SLE

$$\begin{pmatrix} \mathbf{M}_{\hat{t}}^T\mathbf{M}_{\hat{t}} + \mathbf{F}_{\hat{t}}^T\mathbf{W}_{\hat{t}}\mathbf{F}_{\hat{t}} & \mathbf{C}_{\hat{t}}^T \\ \mathbf{C}_{\hat{t}} & 0 \end{pmatrix} \begin{pmatrix} \hat{\mathbf{p}}_H \end{pmatrix} = \begin{pmatrix} \mathbf{M}_{\hat{t}}^T\mathbf{b} + \mathbf{F}_{\hat{t}}^T\mathbf{W}_{\hat{t}}\mathbf{p}_{\hat{t}} \\ \mathbf{p}_{\hat{t}} \end{pmatrix} \tag{4.18}$$

where $\mathbf{C}_{\hat{t}}$ and $\mathbf{F}_{\hat{t}}$ denote the hard and soft constraint matrix respectively. $\mathbf{C}_{\hat{t}}$, $\mathbf{F}_{\hat{t}}$ and $\mathbf{W}_{\hat{t}}$ are constructed using the following form:

$$\begin{aligned} \mathbf{C} &= \text{diag}(c^{oa,1}, \ldots, c^{oa,M}, 0, \ldots, 0) \\ \mathbf{F} &= \text{diag}(0, \ldots, 0, f^{ca,1}, \ldots, f^{ca,N}) \\ \mathbf{W} &= \text{diag}(0, \ldots, 0, w^{ca,1}, \ldots, w^{ca,N}) \end{aligned} \tag{4.19}$$

The elements of $\mathbf{C}_t$ and $\mathbf{F}_t$ are binary matrices that indicate if a joint is attributed a hard or soft constraint. The matrix $\mathbf{M}_{\hat{t}}$ is the expanded Laplacian L and transforms the current pair of poses $\hat{\mathbf{p}}_H$ into topology coordinates using the Topology $\mathcal{T}_i$. In essence, this transformation accounts for the conversion from Cartesian space into neighborhood-preserving differential coordinates using the context-dependent topology (see chapter 4.3.4). The weight matrix $\mathbf{W}_{\hat{t}}$ captures the importance of each joint and it is extracted from the segment weight $\mathbf{W}_m$ that contains time step $\hat{t}$ (see Equ. 4.9).

$\mathbf{C}_{\hat{t}}$, $\mathbf{F}_{\hat{t}}, \mathbf{W}_{\hat{t}}$ and $\mathbf{M}_{\hat{t}}$ are sparse matrices and the system in Equ. 4.17 is overdetermined by at least one hard constraint. As a result, it can be efficiently solved using least squares. In contrast to other IM approaches, cf. chapter 2, the topologies and optimization scheme in the presented context-dependent methodology are adapted at every frame t in order to create the robot behavior based on the current user pose allowing for context-sensitive and instant responses. Also constraints are updated at segment level which allows for varying joint weights during the course of an interaction. Furthermore, it is be noted that other IM methods assume that each joint is attributed a vertex in the net topology [39, 57, 86]. As a result, a large number motion capture markers are required to record interactions. This is, in addition the increased computational costs, often unfeasible as most capturing systems support only a limited amount of markers. This limitation becomes particular important for two-person recordings where twice the amount of markers is needed.

The SLE in Equ. 7.9 accounts for the adaptation of a previously recorded poses $\mathbf{p}_{\hat{t}}$ to the current situation while preserving constraints of the initial demonstration. In the end, this optimization process yields a pair of poses that minimizes the deformation energy with respect to the motion capture recording. As the sparse IM topology features only a subset of all joints of the agents, the final joint angles are generated by employing an IK solver. Each joint position of the optimized posture is thereby attached to the target skeletal structure of the agent. Naturally, this mapping becomes obsolete whenever the two structures resembles each other so that the correspondence problem can be neglected. This, however, is generally not the case for most humanoid robots since they still offer less degrees of freedom. For that reason, the optimized posture needs to be transformed into the kinematic structure of the robot, rendering

IK indispensable. A benefit of this is however that physical limitations of a robot such as joint velocities and torques can be accounted for. These features are generally not captured in IMs, yet they are particularly important for safe human-robot interaction. Also, using IK to compute the final agent posture provides means of transferability to other skeletal structure. In chapter 5 for example, virtual characters learn how to respond to user motions in a human-agent interaction setting. The same interaction model approach is used in chapter 6 and 7 to control a 6-DOF robotic arm. For that only the kinematic chain of the IK solver is altered to match the required structure of the controlled agent.

4.5. Conclusion

In this chapter a novel interaction learning framework has been presented that, at its core is based on human-human behavior demonstrations. The temporal and spatial body synchrony of two-person interactions are recorded using motion capture and structured in such a way as to allow a robot to engage in similar interactions with a human counterpart. Essentially, the model provides a robot the ability to coordinate its own actions during a joint task with a human. It offers a generation scheme for instant responses so that seamless human-robot collaboration can take place. It thus combines behavior recognition as well as response generation in a single model.

Structurally, the interaction model consists of several data representations to efficiently encapsulate task demonstrations and provide different levels of generalization. First, a global posture space is computed and all interaction demonstrations are projected into low-dimensional space. Optionally, an HMM can be trained to enhance state estimation by representing motions on a sequential level based on key poses. The HMM associates sequences of key poses to interactions for fast and efficient classification during runtime. Temporal generalization is achieved in local posture spaces. Here, each interaction recording is reduced in dimensionality and motion segments are temporally aligned during runtime in order to extract a pair of poses from the initial recording that best fits the ongoing human-agent interaction.

Spatial generalization is achieved by deforming context-dependent IMs at each frame. In contrast to other adaptation approaches the topology of the net is created automatically from motion capture data without the need of additional user interference. Instead of relying on extensive marker setups, the topology is created only with joints that contribute to the interaction. As a result, the approach is usable with motion capture systems that do not track a large amount of markers.

Using a combination of Cartesian coordinates for motion capture, low-dimensional spaces for motion recognition as well as temporal alignment and topology coordinates for spatial adaptation, the benefits of each data representation are fused and harnessed in a single methodology. This allows a robot to recognize human motions efficiently and select an interaction demonstration accordingly. The ability of spatiotemporally adapting task demonstrations to new situations is a core feature of interaction models.

They allow robots to seamlessly interact with people in a natural and intuitive way.

The feasibility of the proposed methodology is evaluated in the following chapters where several application scenarios are presented. First, in a virtual reality scenario a character is trained to response to various user interactions in a believable and natural manner. VR simulations provide unique opportunities for evaluating temporal and spatial generalization capabilities without additional latencies or hurdles of real-world robots. Several human-robot collaboration examples are presented thereafter. In chapter 6, a robot learns to assist during two complex assembly tasks which require body synchrony and continuous behavior control. Chapter 7 presents the application of interaction models in triadic human-robot handover settings. Here, the influence of objects is analyzed in detail and a user study is conducted to reveal the improvements over traditional handover methods that are not based on human-human task demonstrations.

5. Behavior Generation for Interactive Virtual Humans

In this chapter the interaction model approach is applied and evaluated in a Human-Agent Interaction (HAI) setting. The model is employed in a fully immersive virtual world to allow virtual characters to respond to human motion in a natural and intuitive way. One of these application scenarios is depicted in Fig. 5.1, where a virtual agent reacts to casual high-fives.

Fig. 5.1.: A virtual character's animation is calculated based on an interaction model. The model has been generated with human-human task demonstrations. During the course of an interaction with the virtual agent, the user's current and previous postures are analyzed to select an interaction demonstration and temporally align the observed motion. Then, using context-dependent IMs, the character's posture is spatially adapted to best fit the current situation, e.g. to match the user's hand in a high-five interaction.

5.1. Introduction

Intelligent virtual agents have found widespread applications ranging from computer games [112, 113], to educational software [114, 115] and shopping assistants [116]. In order to be able to engage in interactions with a human user, they require, similar to collaborative robots, intuitive programming interfaces that minimize the amount of programming. **In the following, the interaction model approach is applied in a VR setting to allow virtual characters to seamlessly respond to human motions.** The application in HAI scenarios offers additional analysis options over real-world robot experiments. It allows the examination of spatiotemporal generalization

capabilities without technical hurdles that are typically introduced by robots. Physical aspects such as control latencies or safety limitations can be avoided which simplifies the testing procedure. Using the approach in VR applications further highlights its ability of generating full-body motions. Whereas most robots feature only a limited set of DOFs, virtual characters are highly articulated and resemble the human skeletal structures closely.

In the following, steps required to generate an interaction model for HAI scenarios are described. Subsequent sections discuss the specifics for fully-body control and show the characteristics of the posture spaces as well as context-dependent IMs in detail. In section 5.4 the model is utilized in different applications. Focusing on spatiotemporal generalization, the adaptation process is thereby evaluated and compared to traditional IMs.

5.2. Learning an Interaction Model for Human-Character Interaction

Before an interaction model is learned, human-human interaction demonstrations are acquired using motion capture. Based on these recordings the library of responses, i.e. the posture spaces for state estimation and temporal alignment as well as context-dependent IMs for response adaptation, are computed.

5.2.1. Data Acquisition

To create the library of interactions two humans performing collaborative tasks are recorded using motion capture. For that, an optical tracking system by A.R.T is used to capture both interaction partners at 30 frames per second. Markers are attached to each extremity of the interactants with additional markers on the head, pelvis and elbows. Each motion capture marker is treated as an IM vertex, which differs from other methods, i.e. [41] and [117] where additional vertices are sampled on the surfaces of the characters' meshes to create required vertices. Using a sparse marker setup instead of a comprehensive full body capturing system, leads to less computational load during IM generation as well as during human-agent interaction. The layout of motion capture markers that is utilized to record two-person interactions is illustrated in Fig. 5.2.

During a live human-agent interaction the user assumes an active role and its motion is captured using the aformented marker configuration. Based on these live recordings the virtual agent reacts to executed user motions. To preserve consistency to earlier chapters, the user will be referred to as the observed agent whereas the virtual agent is called the controlled agent.

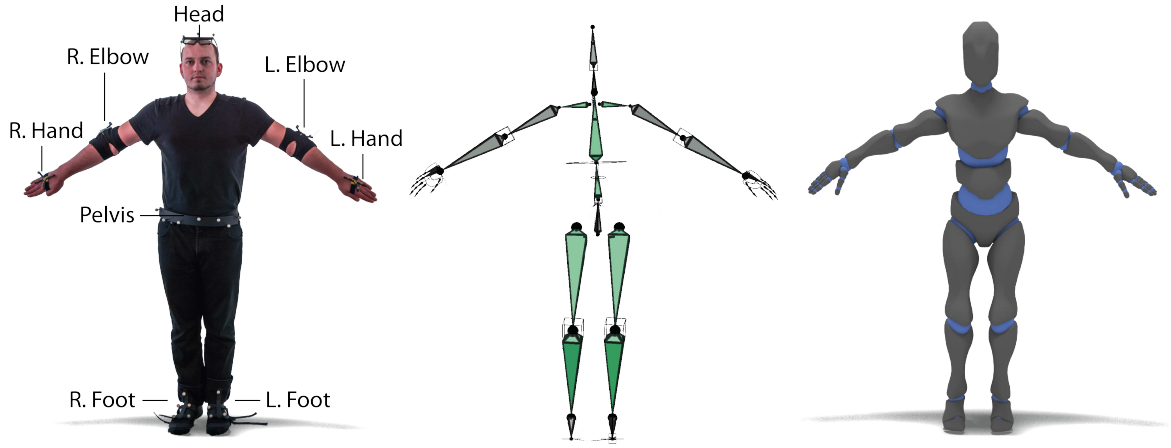


Fig. 5.2.: The behavior of both interactants is recorded using motion capture. Left: A marker is attached to each extremety and additional motion capture targets are placed on the head, pelvis and the elbows. Middle: The skeletal structure of the virtual characters used for animation. Colored regions indicate joints that are computed IK. Right: A virtual character that is used for displaying agent responses.

5.2.2. Posture Spaces Creation and their Segmentation

A crucial part of the interaction model framework is to extract information about the importance of joints with regard to their contribution to the overall motion. In a fighting scenario, for example, the leading hand would be attributed with a high importance because it steers the interaction. In contrast, feet balance the character on the ground, and their motion is considered less important. Hence, a lower importance would be assigned to them. Maintaining certain spatial relationships between joints may, however, change during an interaction, for example, when switching the leading hand or including a kick.

In an interaction model this relationship is preserved, allowing for varying joint importances during the course of an interaction. For that PCA and motion segmentation is applied in several ways. In the first step PCA is applied to all motion capture data of the observed agent yielding a global posture space $\mathcal{G}$. Naturally, a trajectory in low-dimensional space corresponds to a fully-body motion when projected back to its original dimension and when applied to the observed agent's motion capture data, the most relevant information $H(\mathbf{p}^{oa})$ on how the user moved during all recorded interactions is preserved in low-dimensional space. However, since the transformation into low-dimensional space results in information lost, i.e. $H(\mathcal{R}) > H(\mathcal{G})$, fine details and nuances of the motions are in general not well preserved. Towards that end, PCA is then applied in the second step to each recorded interaction yielding several local posture spaces $\mathcal{L}_i$. The global posture space $\mathcal{G}$ as well as all local posture spaces $\mathcal{L}_i$ are subject to segmentation to allow for varying joint importances (see Equ.4.6). Transitions between two adjacent segments correspond to the most important postural changes of the motion in high-dimensional space.

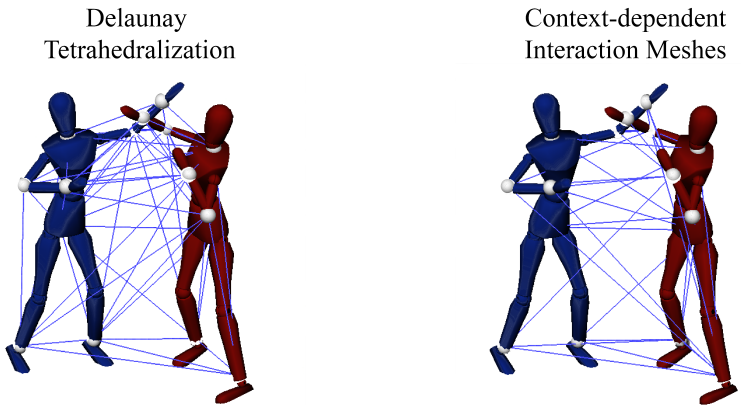


Fig. 5.3.: The figure shows two IMs created for a punch motion. Left: Delaunay tetrahedralization is used to compute a mesh topology. Right: The context-dependent mesh generation. Here, joints that correlate during motion capture are connected by an edge producing a more compact representation of the mesh.

5.2.3. Interaction Selection

In the considered human-agent interaction scenarios the observed agent retracts to a rest pose in all motion capture recordings (see chapter 5.4). As a result motion trajectories in global posture space have similar start and end points. Also, the recorded interactions do not require a sequential ordering and posture similarities on a per-frame basis provide enough information to reliably infer an interaction demonstration during runtime. As a result, the distance-based interaction selection approach is used. For that the observed agent's posture $\mathbf{p}^{oa} \in \mathcal{R}$ is projected into the global posture space $\mathcal{G}$ during human-agent interaction leading to a new point $\mathbf{p}^{oa} \in \mathcal{G}$. The distances of $\mathbf{p}^{oa}$ to all trajectory segments is computed for each frame and added to the distance matrix $\mathbf{E}$. In the end, the index of the column with the smallest mean value corresponds to the interaction that is most similar to the poses in the sliding window.

5.2.4. Interaction Mesh Creation

In an interaction model for human-character interactions a context-dependent IM is computed for each motion capture frame of the human-human interaction demonstrations. Its topology is thereby generated using joint correlations over segments $\mathcal{Q}_m$ and distances in local posture space $\mathcal{L}_i$ (see chapter 4.3.4). In doing so, the joints that correlate and are in close proximity are connected by an edge, forcing the structure to focus on the most relevant pairs of joints. Essentially, tetrahedra with a single vertex on the observed agent and 3 vertices on the virtual character are generated. The resulting topology for a single timestep of a punch motion is depicted on the right of Fig. 5.3. In comparison to Delaunay triangulation (Fig. 5.3 left), context-dependent IMs generate sparse net topologies while at the same time allowing for a wider range of motion (see section 5.4).

In addition to varying net topologies, constraints are automatically defined at segment level. Joints that are not correlating to other joints are attributed a soft constraint

depending on their amount of correlation (see Equ. 4.9). In the aforementioned punch interaction, the controlled agent's feet, head, hip and left arm are attributed a soft constraint. The right elbow is also assigned a constraint since it naturally correlates to the right hand. However, its weight is significantly lower as it features a strong correlation. At the same time all joints of the observed agent are attributed a hard constraint as they cannot be subject to optimization. The remaining hand joint of the controlled agent is not attributed any constraint. It is thus freely adapted during optimization to best fit the current situation.

The topology of the net possibly changes in each motion capture frame and constraints vary in each segment. This allows for varying joint importances during the course of an interaction and enables virtual characters to retain the body synchrony and spatial relationships of the initial motion capture recording. As a result more natural and intuitive interactions take place.

The posture spaces and context-dependent IMs are the interaction model. It is used in the following to generate a controlled agent's response to an observed agent's motion. It thereby accounts for spatial as well as temporal variations of the interaction partner at each frame.

5.3. Live Human-Agent Interactions

During runtime the learned interaction model is employed in a human-agent setting to compute a controlled agent's full-body response in real-time. For that the user's motion has to be classified, an interaction has to be selected and the user's motion has to be matched against previously recorded motion capture data before a suitable response can be optimized. The interaction model is thereby utilized in a hierarchical fashion. First, the user's live poses $\mathbf{p}^{oa}_{H-S:H} \in \mathcal{R}$ are projected into the global posture space $\mathcal{G}$, yielding a new trajectory $\mathbf{p}^{oa}_{H-S:H} \in \mathcal{G}$. Evaluating the distance matrix $\mathbf{E}$, i.e. the proximity to all segment centroids (see Equ. 4.13), gives the interaction demonstration that is most similar to the current situation.

When an interaction demonstration has been successfully determined the user's motion is projected into the corresponding local posture space $\mathcal{L}_i$ for temporal alignment. Within $\mathcal{L}_i$, a pair of consecutive segments $\mathcal{Q}_1, \mathcal{Q}_2$ from the initial recording is selected by computing the Euclidean distances to $\mathbf{p}^{oa}_{H-S:H} \in \mathcal{L}_i$. In order to account for temporal variations of the user's motion, $\mathbf{p}^{oa}_{H-S:H}$ is matched against the joined segments $\mathcal{Q}_1$ and $\mathcal{Q}_2$ using DTW. The end of the warping path provides a point - and consequently a time step $\hat{t}$ - from the initial interaction recording that best fits the current spatial and temporal context of the observed agent's motion.

Since each motion capture frame is associated with a context-dependent IM and a set of constraints, an optimization problem is formulated that adapts the pose of the controlled agent to the new situation (see chapter 4.4). The resulting posture is then transformed to match the controlled agent's skeletal structure using an IK solver. In doing so, joint limits and physical constraints of the controlled agent are maintained

in the virtual world. This contrasts to traditional IM approaches where additional vertices are sampled on the virtual character's mesh surface to reconstruct joint angles [39, 57, 86]. Here, joint velocities and limits are added to the optimization problem using additional constraints in order to preserve the visual appeal of the final motion. This, unfortunately, increases the computational demand during runtime since more restrictions are burdened on the SLE. At the same time the amount of change at each frame is limited to a user defined threshold which requires manual fine tuning for each interaction.

5.4. Evaluation

In the following key characteristics of the interaction model are evaluated in different human-agent settings. The experimental setup and its impact on an interaction model implementation are discussed first (see section 5.4.1). Then, spatial and temporal properties of posture spaces (see section 5.4.2) and the generalization capabilities of context-dependent IMs (see section 5.4.3) are evaluated. Finally, current drawbacks and limitations are discussed.

5.4.1. Experimental Setup

Spatiotemporal generalization is an important ability of interaction models. In order to evaluate its capabilities several interactions are recorded using two-person motion capture. W.r.t. the classification in chapter 1.3, the captured human-human demonstrations are divided into two groups, which will be introduced first. The visualization system used to display agent responses and the resulting technical hurdles are described afterwards.

Human-Human Interaction Demonstrations

To evaluate the proposed method several two-person interactions have been recorded using an optical tracking system by A.R.T. The human-human demonstrations are composed of various interdependent behaviors that follow two main goal structures:

- First, a set of *collaborative interactions* including high fives, a hand clapping game, waving at each other and a jive dance are recorded.
- And second, *competitive interactions* including two different kicks, an upper cut, left/right punches and their appropriate defenses motions are captured.

The motions of both human demonstrators have been tracked at 30 frames per second, utilizing the marker layout depicted in Fig. 5.2. Fig. 5.4 shows example motion capture recordings for the cooperative as well as competitive scenario. Each motion capture target provides a 3-dimensional position - and an unused rotation - leading to an overall state space of 24 dimensions for each interactant.

Fig. 5.4.: The figure shows three two-person motion capture recordings. Left and middle: Casual high-fives and clapping games are recorded. Right: An example kick motion for the competitive scenario.

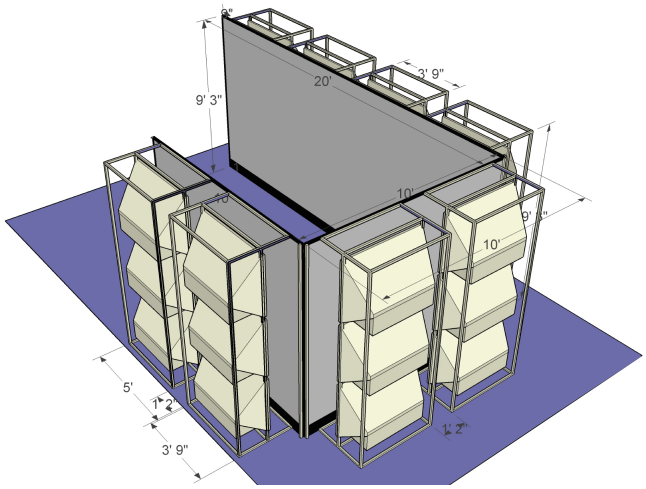


Fig. 5.5.: Illustrated is 4-sided CAVE installation that utilizes back projection televisions to generate a seamless high-resolution image.

Visualizing Agent Responses

During the interaction with a virtual character, agent responses are visualized in a CAVE environment to allow for natural and intuitive interactions. Composed of 24 Full HD projectors, the XSITE CAVE features high-resolution renderings of agent responses at almost 50 Mega pixels (see Fig. 5.5). Such a visualization system offers several benefits over traditional desktop environments. First of all, it allows renderings of virtual characters at natural heights and in doing so presents a much more realistic view of the interaction partner. This enhances the perception of the user considerably and he feels no longer present in the physical world but instead immersed in the virtual world. The resulting *immersion* is key for intuitive and natural human-agent interaction [118]. And second, CAVEs - in combination with motion capture - allow users to move freely without being constrained to traditional desktop settings. Instead of entering commands with gamepads or joysticks, users are able to use their own body to control an avatar. Their motions are transmitted and adopted by the character in real-time, boosting embodiment of the individual in the simulation [118]. This strengthens

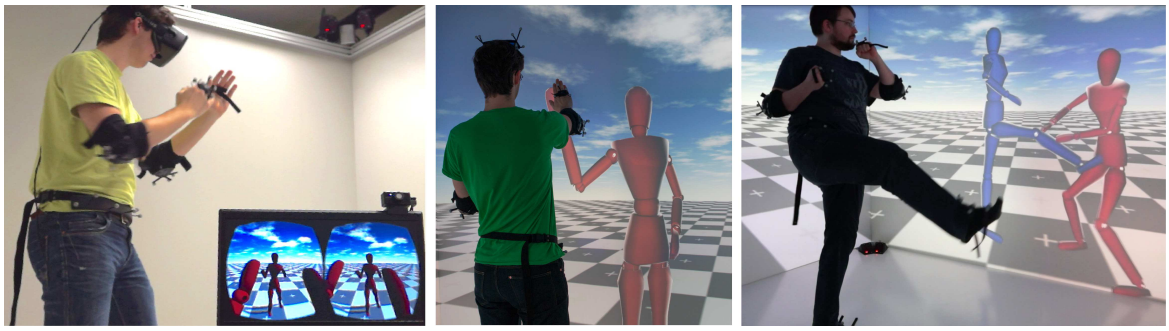


Fig. 5.6.: Interaction models for human-agent interaction are applicable in various visualization systems. Left: a fighting interaction is displayed in a HMD. Middle: a high-five interaction in a fully immersive CAVE environment. Here, the contact point between both interactants is on the projection screen. Right: a fighting interaction where the user interacts with the virtual character through an avatar.

the user's feeling of immersion in the scene and improves his performance in achieving tasks, e.g. high-fifing a virtual character.

An alternative to CAVE installations are Head Mounted Displays (HMDs) which offer a similar degree of immersion (despite a narrower field of view). In the end both visualization techniques offer their own benefits as well as limitations and the presented interaction learning methodology is applicable in both scenarios. For reasons of availability and presentability all experiments in a fully immersive CAVE environment.

In order to animate the virtual character in either visualization systems using an interaction model several difficulties need to be taken into account. CAVE installations typically require several networked computers to render images seamlessly and network delays as well as transmission times add to the overall latency that is introduced by an interaction model implementation. At the same time, latencies towards observed user motions need to be lower than 250 ms so that agent responses appear interactive instead of preprogrammed [119]. In the end, an algorithmic foundation as well as an efficient implementation is called for in this time critic environment.

5.4.2. Evaluating Interaction Selection

Recognizing and classifying user motions during human-agent interaction is an important feature of interaction models. The global posture space $\mathcal{G}$ that is generated by applying PCA to the observed agent's motion capture data is used to infer an interaction demonstration. The corresponding local posture space $\mathcal{L}_i$ of the matching interaction is then employed for temporal alignment of observed behaviors.

For the collaborative interaction scenario PCA created a 15-dimensional space $\mathcal{G}$ that resembles how the user moved during all four interaction demonstrations. The first 3 principal components are depicted in Fig. 5.7. In a live human-agent interaction a user was tasked to high five the virtual agent. As expected its motion varied from the initial recording. However, its trajectory in low-dimensional space stills followed the same direction. This is due to the fact that similar postures were adopted which in turn lead to neighboring low-dimensional points.

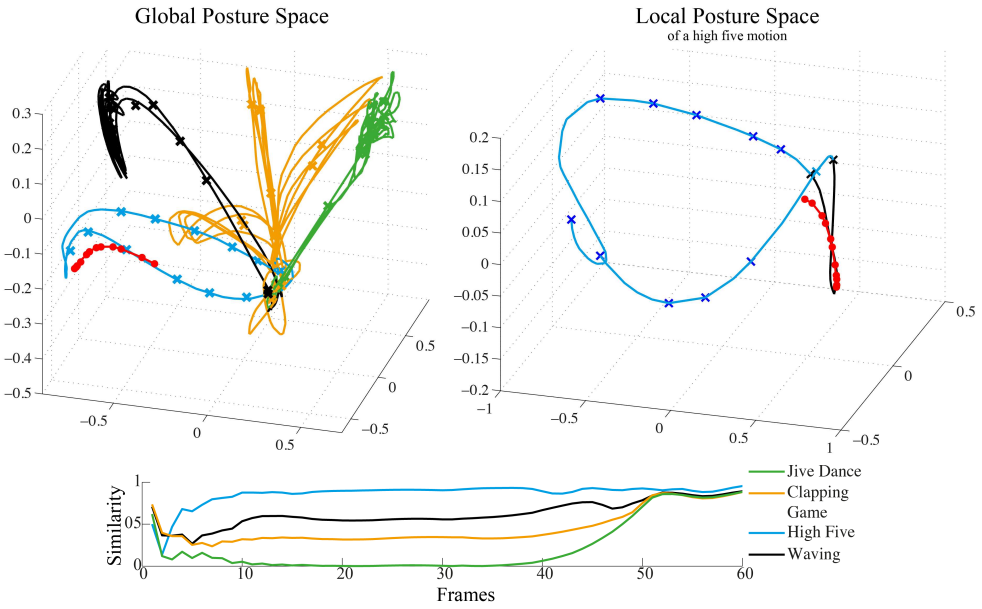


Fig. 5.7.: In the figure on top the global as well as the selected local posture space is shown. A user was tasked to high five the virtual agent and its motion is indicated by red dots. Below the normlaized mean distance to the closest centroid (denoted similarity) of each interaction is visualized. The high five motion is most similar to the executed user motion. However other motions also exhibit similar poses especially around frame 50 to 60.

Fig. 5.8.: The virtual agent's postures are optimized for live human agent interactions. In this example a user high fives a virtual character successfully. The agent adopts its motion to meet the users hand at the right time and position.

On the right hand side of Fig. 5.7 the local posture space of the selected interaction is visualized. Here the closest matching motion segment of the initial recording is marked (black trajectory). As can be seen, the user motion (indicated by the red trajectory) also follows the path of the closest segment. After calculating the DTW cost matrix, a matching point is selected and its associated context-dependent IM is optimized. The resulting character responses can be seen in Fig. 5.8 for 3 frames.

In a second example the same global posture space is utilized to detect a ongoing jive dance motion. The projection of current and recent user postures into the global low-dimensional space are shown in Fig. 5.9 top left. As depicted, its motion matches the shape of the jive template which has been generated from the initial recording. Additionally, the local posture space corresponding to the selected interaction is shown. The most similar segment is highlighted. The similarities of the live user motion to recorded interaction examples are in Fig. 5.9 bottom. As can be seen other motions

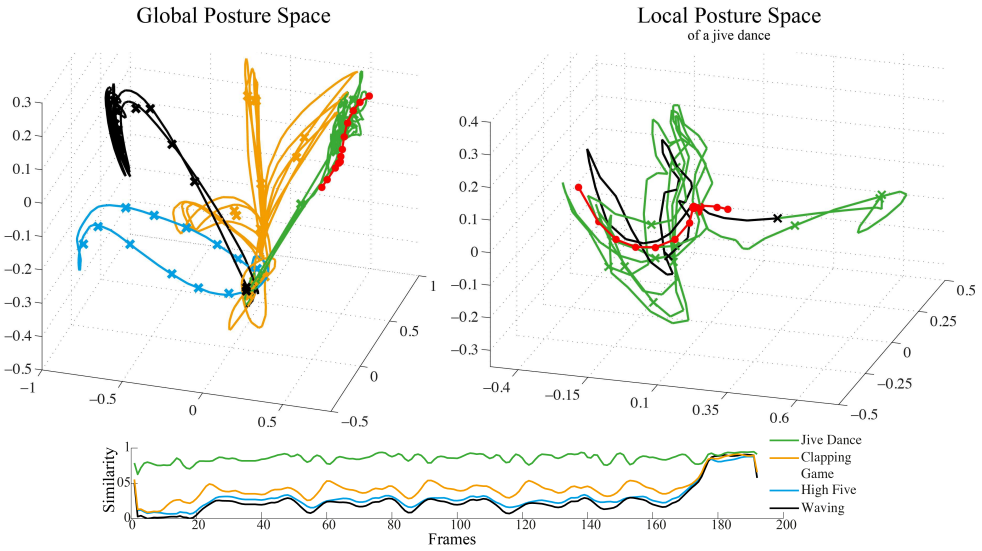


Fig. 5.9.: The global and local posture space of a Jive dance motion are shown on top. The live user postures are highlighted red. The normalized mean distance to the closest centroid (denoted similarity) of each interaction is outlined below.

Fig. 5.10.: The motion of a virtual character is optimized in real-time using the interaction models approach. As can be seen the agent successfully imitates the behavior shown in the initial recording.

also exhibit similar postures but remain inactive due to their larger mean distance in the global posture space (c.f. Equ. 4.13). The reason for the large similarities towards the end of the interactions is that in all recordings, the observed agent returned to a rest pose. The final character response can be seen in Fig. 5.10.

In a third example a hand clapping game is performed with a virtual character. Here the same global posture space $\mathcal{G}$ from the collaborative interaction demonstrations is used. As shown in Fig. 5.11 the projected user postures (highlighted red) match the template created from a clapping game motion. Essentially, the selected interaction demonstration has been a high five at first (see frame 1 to 20) but changed later to the correct interaction. The reason for that is that similar postures have been obtained in both motion capture recordings. It is noted that the HMM-based state estimation approach could be used to eliminate the initial ambiguity. However, since the virtual agent's response posture is similar in beginning of both interactions, no additional differentiation is required. The final character responses are illustrated for 4 key postures in Fig. 5.12.

For the competitive interaction scenario a 12-dimensional space $\mathcal{G}$ was created based

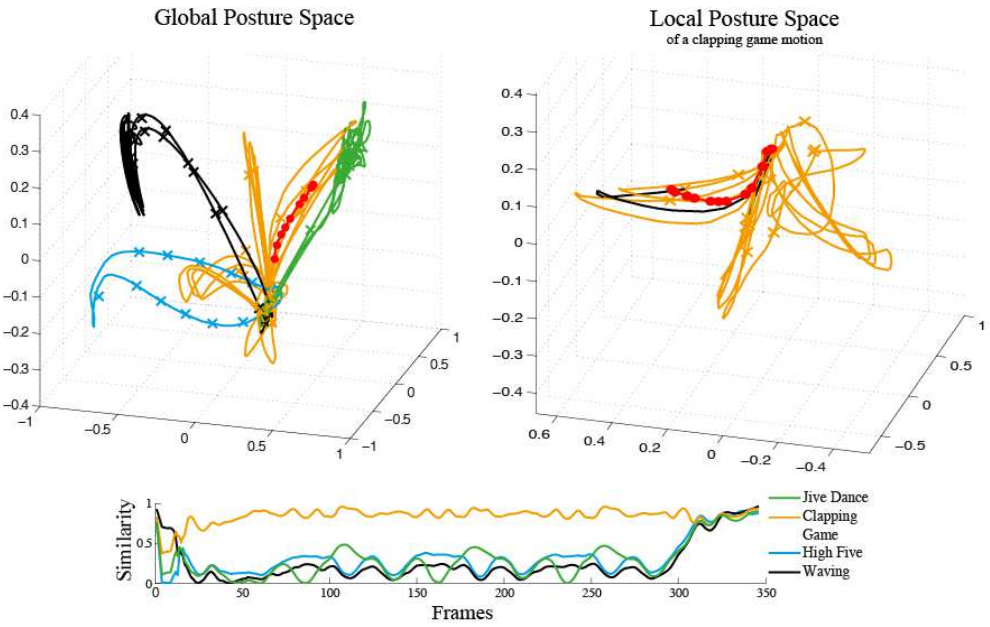


Fig. 5.11.: The figure shows the global posture space and projected live user postures (highlighted red). On the right hand side the selected local posture space of a clapping game motion is visualized with 10 previous user poses for motion matching. Additionally, the normalized mean distance to the closest centroid (denoted similarity) for each interaction type are illustrated below. As can be seen a high five motion is selected at first but changed later to the correct clapping game.

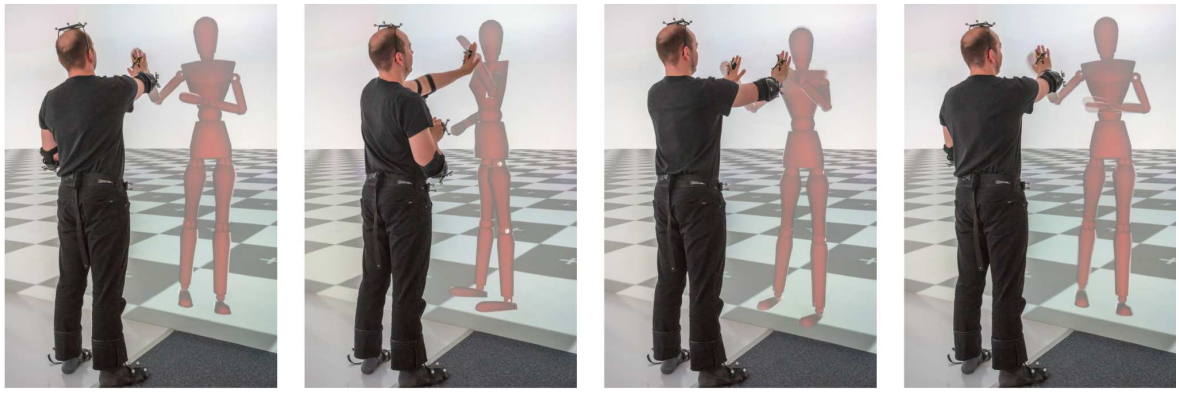


Fig. 5.12.: With an interaction model a virtual character can respond to complex interactions like a clapping game as shown in the figure for 4 key postures. Here the agent's hand has to meet the users palm at the right time and at the right position. In contrast to the examples above no avatar is added to mediate interaction with the virtual character. Instead the human user touches the controlled agent on the projection screen.

on all 6 interaction demonstrations. It is visualized on left of Fig. 5.13 for the first 3 PCs. On the right of Fig. 5.13 the local posture space of a right-hand punch is shown. Here, only 2 dimensions are required to represent 98 % of the motion. This is due to the simple nature of the right punch behavior which only involves one extremity with both feet firmly resting on the ground (see Fig. 5.14 for 6 points in time). In the figure the left character is controlled by the user and the right agent is animated using the interaction model. As expected the motion of the controlled agent resembled the initial motion capture recording while at the same time being spatially and temporally

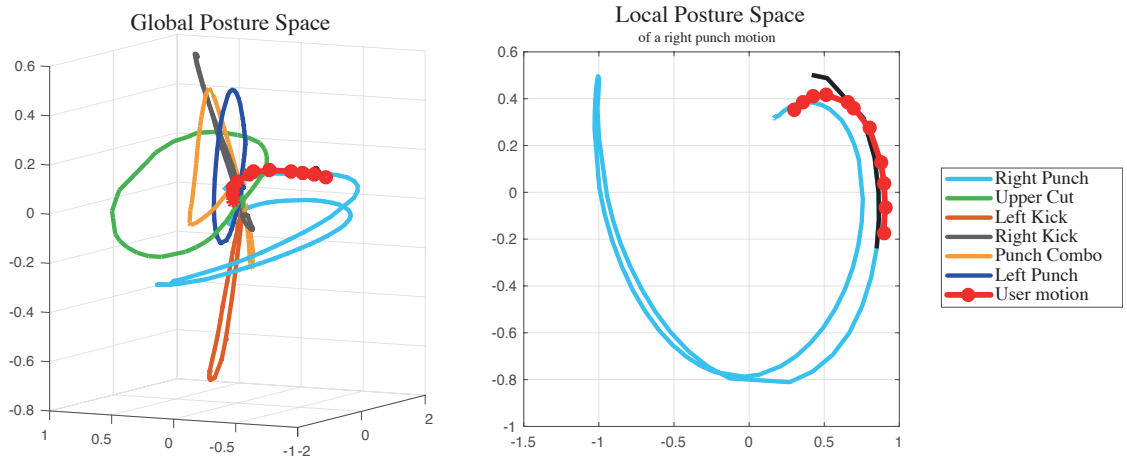


Fig. 5.13.: In the figure on the left illustrates the global space for the competitive interaction scenario. On the right the the local posture space for a left-handed punch is shown. Interestingly, two PCs are sufficient to represent 98 % of the interaction in the local posture space. An excerpt of the live user motion is highlighted red.

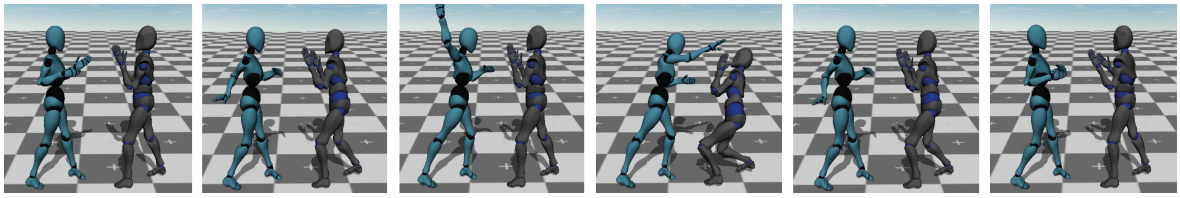


Fig. 5.14.: The figure illustrates different stages of an upper punch motion. The left agent is controlled by the user whereas the right character is animated using an interaction model. The controlled agent moves to a crouching stance as soon as the user's hand moves closer, just as demonstrated during motion capture.

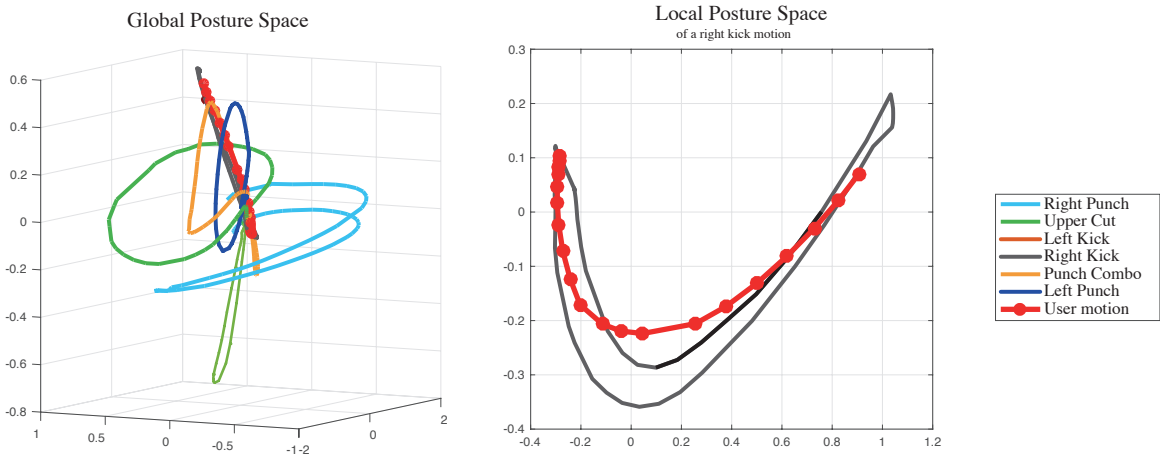


Fig. 5.15.: The figure illustrates the global posture space of the competitive interaction scenario. The user's motion is highlighted red. The interaction model correcly identified the kick interaction and the corresponding local posture space (right).

adapted to the current situation.

Fig. 5.15 illustrates the same global posture space but a different live user motion. Here the user was tasked to kick the virtual agent (see Fig. 5.16). The interaction model matched the correct interaction demonstration from the pool of all interactions. The

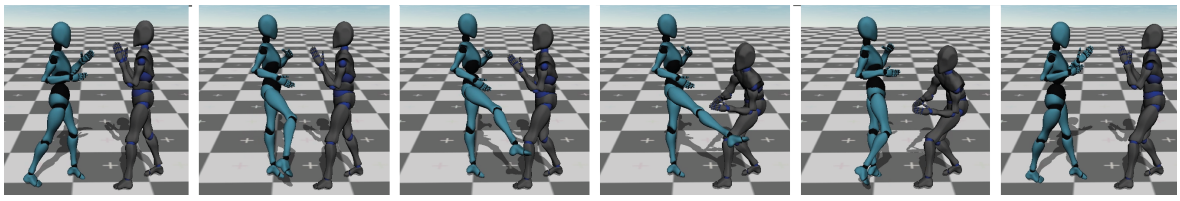


Fig. 5.16.: In the figure several stages of a kick motion are shown. The left agent is controlled by the user and the right character is animated using an interaction model. Just as shown during motion capture, the controlled agent defends the kick with a blocking motion using both arms.

selected local posture space is illustrated on the right of Fig. 5.15. A two-dimensional local posture space was sufficient to represent 98 % of the information.

To summarize the evaluation shows that interaction examples are reliably identified in the global posture space and temporally aligned in local posture space. For the cooperative example, local posture spaces of average dimensionality 5 were sufficient, while for the competitive examples only 2-d local posture spaces were needed.

5.4.3. Evaluation of Context-Dependent IMs

From the recorded example interaction scenarios, two sets of IMs were created: One set of IMs built with Delaunay triangulation and one set with context-dependent IMs. In both cases, the vertices of the IM correspond to the markers of the A.R.T. tracking system. At runtime, for each time-step, the current user posture is queried against the database of prerecorded interaction examples using the posture spaces, yielding the IM that is suited best for the current situation.

For the collaborative scenario a clapping game interaction is depicted in Fig. 5.17 (left). During the interaction, the virtual agent successfully reaches the user's hand at the right time and position when context-dependent IMs are used. Joints of the observed agent that are in close proximity to joints of the controlled agent are attributed a hard constraint. At the same time, joints of the controlled agent that correlate to the closest joint of the observed agent are provided a soft constraint (denoted *virtual agent constraint*). Overall, the constraints for the left and right hand change regularly. The most important constraints of both agents are shown in Fig. 5.17 (right). IMs generated with Delaunay triangulation, failed to preserve spatial relationships when the user motion differed from the initial demonstration. At the same time, constraints can only be set for the entire interaction and, thus, fail to preserve the alternating nature of the game. This resulted in visually unappealing results as highlighted by red circles in Fig. 5.17 (left).

For the competitive scenario the upper row of Fig. 5.14 shows the controlled agent (right) successfully defending the user's punch motion (left). For this example IMs generated with Delaunay triangulation were attributed a soft constraint on left foot of the controlled agent to balance the character on the ground. Still, when the user's motion differed from the initial recording significantly visually unappealing results were produced. This can be seen in the second and third row of Fig. 5.18 (left).

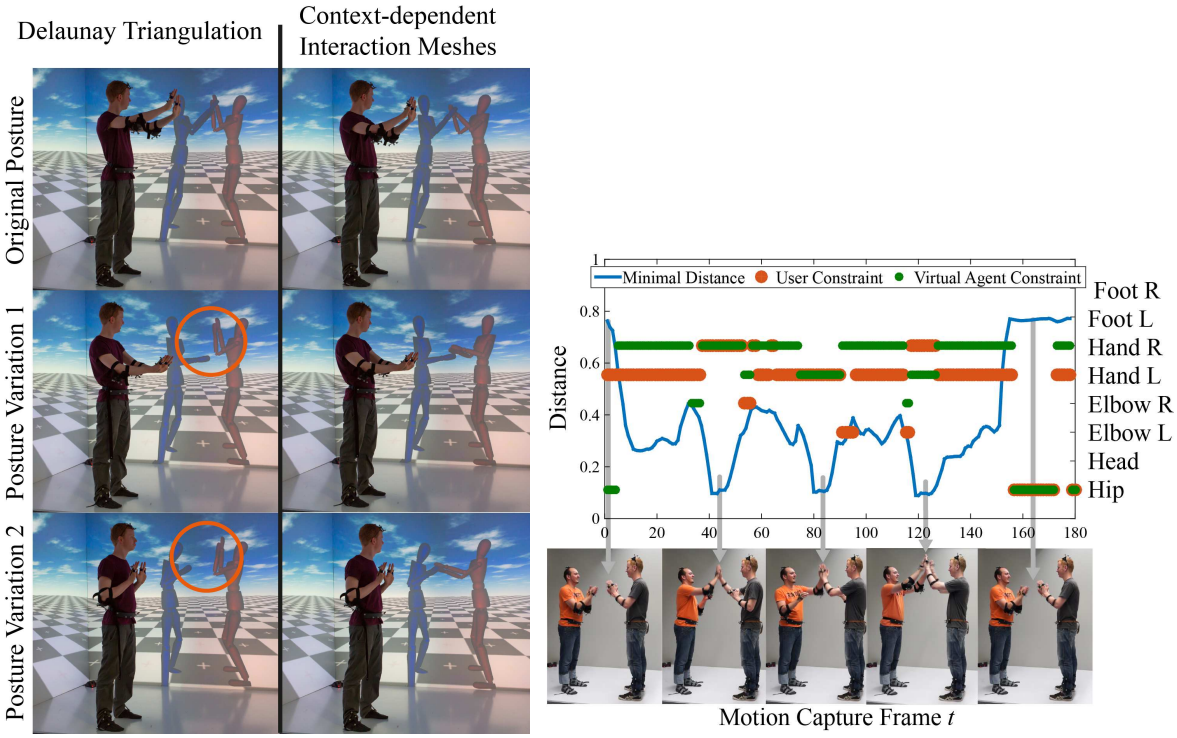


Fig. 5.17.: Left: A clapping game between a human and a virtual agent. Due to strong interconnectivity and missing constraints, interaction meshes using Delaunay triangulation failed to preserve the interaction context. Context-dependent IM automatically attached constraints to both hands. As result, the virtual agent successfully meets the user's hands during the live interaction.
Right: The figure illustrates how the two main positional constraints change during the clapping game. The importance of left and right hands alternates during the interaction.

Here, circles indicate locations where spatial relationships have not been met due to the strong interconnectivity of the IM. In contrast, with the automatically created mesh topologies and associated constraints of the proposed method important spatial relationships of interactions are captured more concisely. Which joints correlate most in the punching example and, hence, are afforded a constraint, is illustrated in Fig. 5.18 (right). As can be seen, different joints are calculated as most important during the different phases of the interaction. In the beginning of the punch motion, the user's left elbow as well as the reactive interactant' right hand are closest and, thus, attributed with constraints that contribute most to the motion adaptation solution. During the climax of the punching motion, the user's left hand and the virtual agent's right elbow are most important. At the end of the interaction, the correlation changes back to the initial conditions.

In the same human-agent interaction session the user was tasked to kick the virtual character as shown in Fig. 5.16 for several time frames and different heights in Fig. 5.19 (left). As expected, IMs created with Delaunay triangulation and the context-dependent IMs approach performed well, allowing the virtual agent to ward of the kick motion if the user's motion resembled the original recording. However, if the kick height varies significantly Delaunay triangulation tends to fail to produce reliable results (see Fig. 5.19 left). It leads to motions of the controlled agent that still resemble the ones

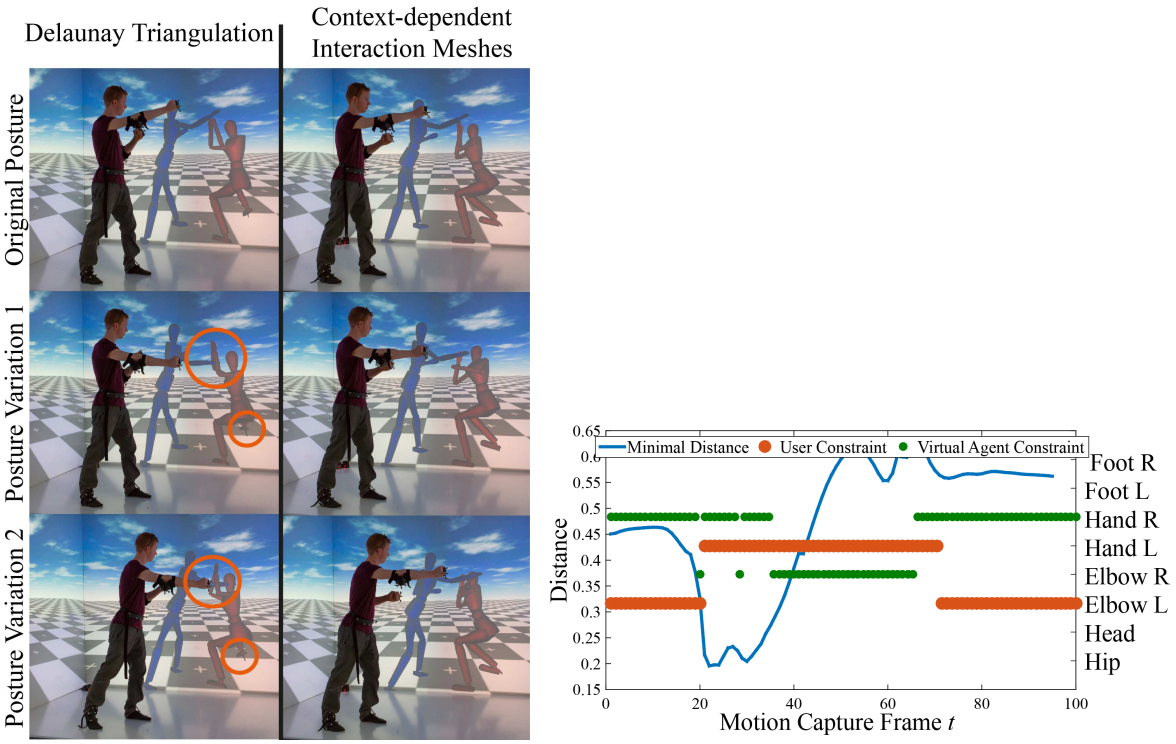


Fig. 5.18.: Left: The user executes right handed punches in varying heights, resulting in corresponding defenses by the virtual agent. IMs created with Delaunay triangulation fail to produce reliable results if the user motion varies too much from the initial recording. Circles indicate where errors occurred. Right: The figure illustrates how the two most important constraints change during a punch interaction for the user (red dots) and the virtual agent (green dots). As can be seen the main constraints are set from the right hand to the right elbow for the virtual agent. Joints with smaller correlations are omitted for reasons of simplicity. Additionally, the minimal distance between joints of the two interactants is shown.

from motion capture but do not preserve all spatial relationships as indicated by circles. The context-based optimization approach, in contrast, creates more plausible character responses as shown on the right. Which joint of the agent has been provided with the most important constraint is depicted in the right of Fig. 5.19. As in the punching examples, the relative importance of spatial relationships changes during the kicking interaction and, hence, different positional constraints are defined during the course of the interaction.

A feature of IMs is the ability to adapt spatially to different users as indicated in Fig. 5.20 for a kick behavior. Here, knee bending angles and kick heights that differed from the initial motion capture recording were adopted. Using the interaction model the appropriate interaction demonstration and the corresponding context-dependent IMs were selected so that the characters response matched the user's particular motion.

5.4.4. Discussion

Due to the sparser mesh topology and fully automated constraint generation, the proposed context-dependent IMs are more susceptible for undesired collisions than standard IMs based on Delaunay triangulation and manual constraint definition. However,

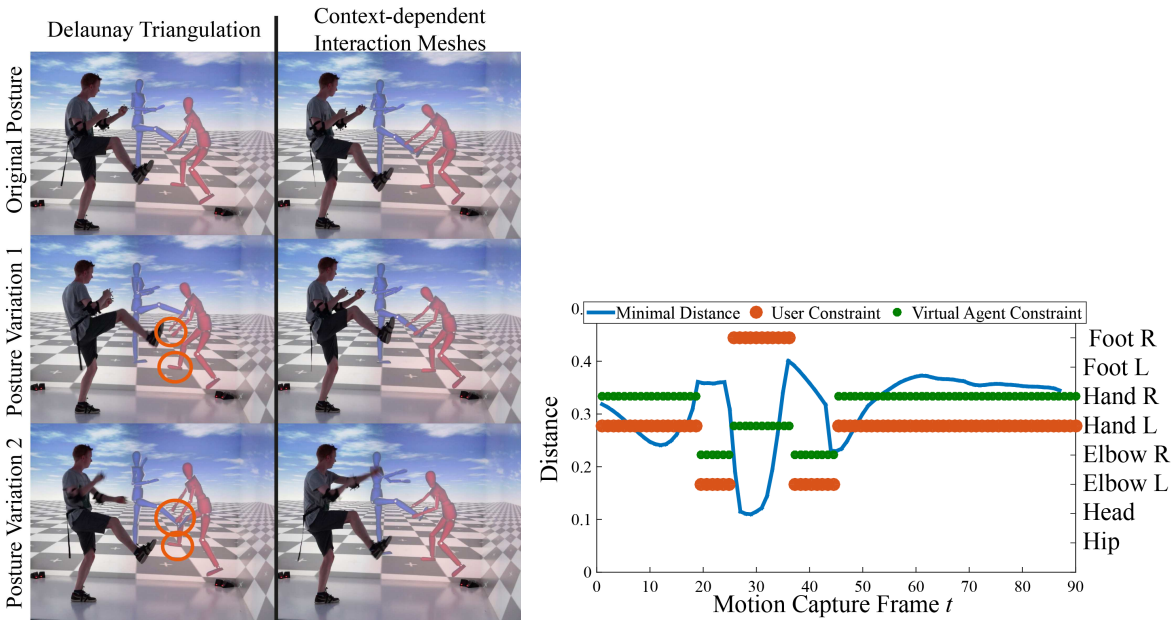


Fig. 5.19.: Left: As part of a fight scenario, the user is kicking a virtual agent. The figure shows varying kick motions and how the virtual character responded. Red circles indicate where optimization errors occurred (left: second and third row). Right: The figure illustrates how the two main constraints change during a kick motion for the user (red dots) and a virtual agent (green dots). It can be seen that user's right foot is of high importance in the middle part of the interaction. This changes during the beginning and the end of the motion to each hand since they correlated more.

Fig. 5.20.: The first row in the figure shows a motion capture recording of a kick motion. During both interactions with the virtual agent users adopted different kick heights and knee bending angles (see row 2 and 3). Still, the interaction model successfully detected the corresponding interaction demonstration and suitable character motions that take the specifics of the current situation into account were created.

a possible solution to this drawback is the usage of a physics simulation during animation. Here the agent's skeletal structure is modeled as kinematic chain and constantly checked for collision. In doing so, joint movement is limited with plausible natural constraints avoiding collisions while at the same time ensuring end effector positions as well as rotations. This strategy is pursued in the traditional non-real-time IM implementation where computational costs play a subordinate role [120].

The interaction model approach for HAI settings utilizes the distance-based approach for selecting a suitable interaction demonstration. In doing so, the minimal distance towards a demonstration in global posture space over a sequence of poses yields the most similar recording. Depending on the sliding window size, it can potentially lock the agent in one interaction. In the experiments a memory size S of 15 has been proven to be well suited. This leads to memory length of approximately 0.5 s at 30 frames per second. Selecting a posture in global posture space as well as motion matching in local posture spaces takes on average 0.008 s. Optimizing a context-dependent IM takes 0.01 s which is similar to Delaunay-based IM approaches. In the implementation used to perform the above experiments, transforming the resulting vertex coordinates to joint angles utilizing the IK solver takes twice as long (0.016 seconds). All in all the presented human-agent interaction setup operates at 25 frames per second with a latency of approximately 230ms on a 2011 MacBook Pro. This includes motion capture, reference posture search, response posture optimization as well as inverse kinematics to reconstruct both character poses in order to visualize them in the immersive virtual environment.

The interaction model used to animate virtual agents does not allow for additional objects to be included in human-agent interactions in virtual reality. This is due to lack of tracking capabilities, required to recreate hand shapes. Even so, it is noted that interaction models are nonetheless capable of reconstructing interactions involving objects (see chapter 7).

5.5. Conclusion

In this chapter, the interaction model approach from chapter 4 has been applied in an HAI setting. It is used for generating real-time responses of an interactive virtual human in various scenarios. Using training data acquired from human-human demonstrations, the model was generated to capture how the interactants moved during demonstration. It has been shown that low-dimensional posture spaces allow for efficient recognition of the observed behaviors during runtime while generalizing to different variations thereof. Based on the same demonstrations, context-dependent IMs and associated optimization constraints were automatically generated. The topologies where created for each interaction to preserve spatial and temporal relationships of the joint behavior. The approach extends previous methods, e.g. [41] to situations where the temporal context of interactions plays an important role. The generation of the interaction model required no user intervention which also differs form existing methods

where skilled animators are required to fine tune optimization parameters [39].

Using the interaction model during runtime in full-body interactions between a human and a virtual character, smooth and instant responses where created in various scenarios. Thereby, the virtual character's motions resembled the previously recorded example interactions between two humans.

Experiments in a fully immersive CAVE environment confirmed that the approach is able to synthesize context-aware and real-time responses for virtual characters. An important insight is, that by using human-human demonstration natural and intuitive virtual character behaviors can be trained. Furthermore, user motions are efficiently recognized without significant delay due to low-dimensional spaces and sparse marker setups. A valuable side-effect of this is, that the amount vertices in an IM topology is reduced, decreasing the computational load during optimization. The approach is thus suitable for real-time applications such as computer games or interactive assistants.

In the following the methodology will be applied in a HRI setting in order to compute seamless and natural responses of a robot.

6. Learning Continuous Human-Robot Interactions

Building upon the methodology from chapter 4, this chapter applies the interaction model approach in the field of robotics. A robot is thereby trained to jointly assembly objects with a human user as depicted in Fig. 6.1.

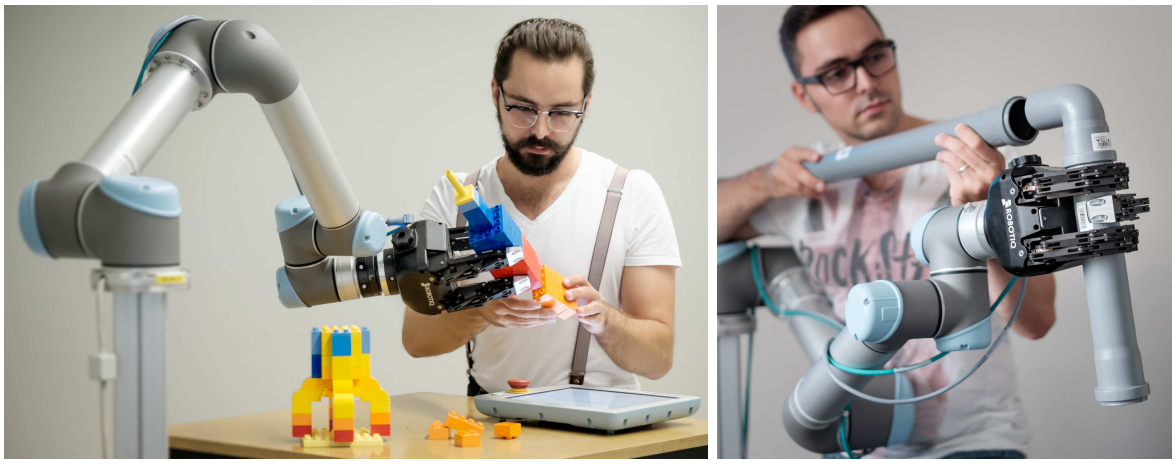


Fig. 6.1.: In a collaborative assembly task, the robotic assistant continuously coordinates its behavior with the human co-worker. The robot's behavior is learned from human-human demonstrations of the various subtasks.

6.1. Introduction

Collaborative human-robot tasks as shown in Fig. 6.1 require seamless behavior control and real-time response generation of a robot in order to be successful. It has been shown in recent literature that the temporal relationships required in these interactions can be efficiently extracted using HMMs for example, c.f chapter 2. These results also indicate that models based on temporal properties alone do not generalize sufficiently to postural changes in typical human-robot interaction tasks [54]. Further spatial optimizations to the robot's posture and movement are required in order to ensure efficient and safe physical interaction. Towards that end, **the interaction model approach is employed in the following in an HRI setting to seamlessly adapt a robot's behavior spatiotemporally to that of a human collaboration partner.** Whereas other imitation learning methods almost exclusively focus on a single agent, an interaction model is based on parallel behavior demonstrations by two interaction partners. In doing so, it inherently captures important spatial relationships and

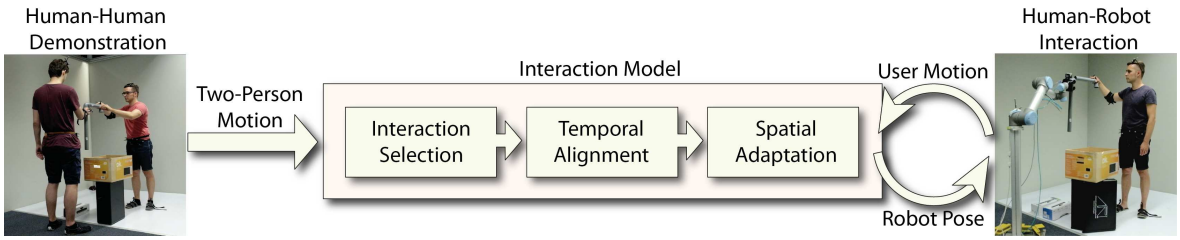


Fig. 6.2.: Overview of the interaction model. It captures, based on human-human demonstrations, how each person moved during the joint execution of a physical collaboration. Using the learned model in a human-robot interaction the robot's role is infered and its controls are computed continuously. The state of the HRI, i.e. a subtask, is derived and then aligned locally in low-dimensional space. Spatial adaptation is achieved using context-dependnt IMs.

synchrony of body movements between two interacting partners. It uses a combination of motion recognition in low-dimensional space and context-dependent IMs for seamless pose generation. This allows a robot to respond appropriately to observed human motions, taking into account temporal *and* spatial relationships. Spatial generalization of task demonstrations with context-dependent IMs has been shown in the previous chapter for human-character interactions.

The contribution of this chapter can be summarized as follows:

- An application of interaction models in complex human-robot assembly tasks
- Quantitative evaluation of interaction models w.r.t their capabilities for spatial and temporal adaptation of two-person motion capture recordings.

In the following an interaction model is generated for human-robot collaboration. The differences to human-character interaction are emphasized in section 6.2 and the generalization capabilities are demonstrated in section 6.3. Here, a robotic arm learns to build a tube frame and a Lego rocket jointly with a human user. Finally, current limitations and directions for further research are discussed in section 6.5.

6.2. Learning an Interaction Model for Continuous Human-Robot Interaction

To generate an interaction model, *human-human* demonstrations of two users performing cooperative tasks are first recorded using motion capture. Several tasks can be demonstrated and only one example demonstration per task is necessary. Generally, a leader-follower type scenario is assumed, where one person acts as an assistant. During the later human-robot interaction, the robot will assume the role of the assistant. In the training phase, an interaction model is learned that describes how the two interactants synchronize their movements. At runtime, it is used to continuously adapt the robot's movements spatially and temporally to that of human partner (see Fig. 6.2).

Fig. 6.3.: The figure illustrates the marker layout used to capture the robot's motion during a two-person task demonstrations. In contrast to interaction demonstrations for virtual character animation less markers are tracked. Here, only the right arm with additional targets on the wrist and thumb are recorded. The motion of the second interactant is recorded using 6 markers attached to both feet, the right elbow and hand with an additional marker on the head and the waist.

For consistency with earlier chapters, the first interaction partner, i.e., the human, is referred to as the *observed agent*, while the second interaction partner, i.e., the robot, will be called *controlled agent*.

In the following key differences of an interaction model for robot control to the one used for character animation are highlighted.

Motion Capture Similar to an interaction model for character animation, two-person motion capture recordings are also used to extract interaction dynamics for adaptive robot control. Each marker provides a position at 30 Hz. During human-human demonstrations and human-robot interactions, each human wears six markers as illustrated in Fig. 6.3. This contrasts to recordings for character animation where a full humanoid skeletal structure is captured. Since the robot that will be used in this chapter is a UR5 robotic arm, 3 markers are attached to the user's hand so that the gripper orientation can be reconstructed in the IM. It will be shown during experiments (see section 6.3) that a total number of 6 markers provides enough information so that a robot's response can be seamlessly inferred and adapted in various tasks.

Interaction Selection At the core of an interaction model, several low-dimensional posture spaces serve to identify relevant interaction demonstrations during HRI. They are computed by applying PCA to the observed agent's motion capture data in several ways. First, a global posture space is computed based on all interaction demonstrations and, second, for each motion capture recording a local posture space is created.

In contrast to the HAI setting, the considered HRI scenarios are collaborative assembly tasks that require sequential interactions. Whereas experiments in VR showed that the distance-based selection method reliably determined the correct interaction in human-character interaction settings, it proved insufficient in HRI settings (see section 6.3). If the selection would be based on posture similarities alone, ambiguities might arise where a single pose of the observed agent leads to different poses of the controlled agent. This is due to the sparser motion capture marker layout and the

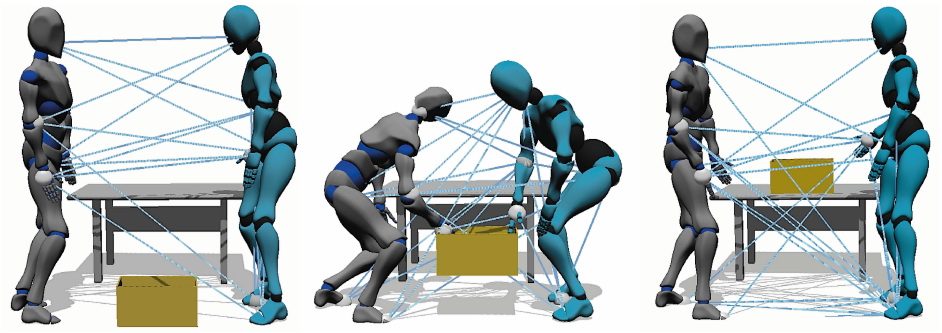


Fig. 6.4.: The rendering shows 3 stages of a box lifting interaction with their corresponding context-dependent IMs. The topology changes based on joint correlations as can be seen for the feet for example. Here, an edge connecting pairs of feet has been added for the last part of the collaboration (right).

strong posture similarities in the assembly tasks. As a result, the robot would constantly switch between interactions even when the user is not moving. The overall accuracy with which interactions are selected correctly increases by a magnitude when sequential information about the order of subtasks is included. Thus, the HMM-based selection method is used (see chapter 4.4.1). For that the trajectories in the global posture space are approximated and compressed using a GMM. Each Gaussian represents a key pose and their sequential order is learned with an HMM. The HMM encodes sequences of key poses and how they relate to interaction demonstrations (each hidden node corresponds to an interaction). During runtime the HMM is continuously evaluated, yielding appropriate interaction demonstrations. The corresponding local posture space is then employed to temporally align the observed user motion (see chapter 4.4.2).

Context-based Interaction Meshes For a human robot interaction to be successful, the motions of the human and the robot partner have to be continuously coordinated. In particular, spatial constraints such as hand contacts have to be retained. In an interaction model spatial adaptation and coordination is tackled using context-dependent IMs. For robotic applications, an IM provides a topological and spatial representation of two humans during a motion capture recording at each time step and vertices of both interactants are defined as Cartesian positions of motion capture markers. In contrast to an interaction model generated for character animation, net topologies for human-robot interactions exhibit less vertices due to the altered motion capture marker layout. As a result, vertices do not necessarily correspond to joints of the robot as the kinematic chain may vary with respect to the human's skeletal structure that has been used during recording. However, pairs of motion capture markers that correlate most are important and their spatial relationship is preserved in a context-dependent IM.

Fig. 6.4 illustrates the evolution of an IM topology for a box lifting interaction between two human partners. Most notably, both pairs of feet and the right arms are considered important and, thus, connected by an edge during net generation.

6.3. Evaluation

The interaction model for HRI is evaluated in two complex assembly tasks involving several manipulated objects. In the first example, a Lego rocket is collaboratively assembled with the help of a robot. In the second example, a tube frame is put together in collaboration between the user and a robot. During demonstration, a box containing a set of pipes is first placed on a stand and two pipes are assembled collaboratively, before the final tube frame is constructed. As illustrated in Fig. 6.6 and Fig. 6.5, the robot imitates the demonstrated motion successfully and the objects for both examples are constructed jointly with the user.

Fig. 6.5.: A Lego rocket consisting of four parts is assembled with a user. The figure shows that the robot imitates the demonstrated behavior successfully while at the same time adopting its poses to the new situation.

Fig. 6.6.: A tube frame is assembled with a user. Top row: Recording of human-human interactions using optical motion capture. In the experiments three interactions were recorded, lifting a box, extracting and assembling a tube before finally constructing a pipe frame. Bottom row: The interaction model is utilized during human-robot interaction and the robotic arm is continuously reacting to observed postures.

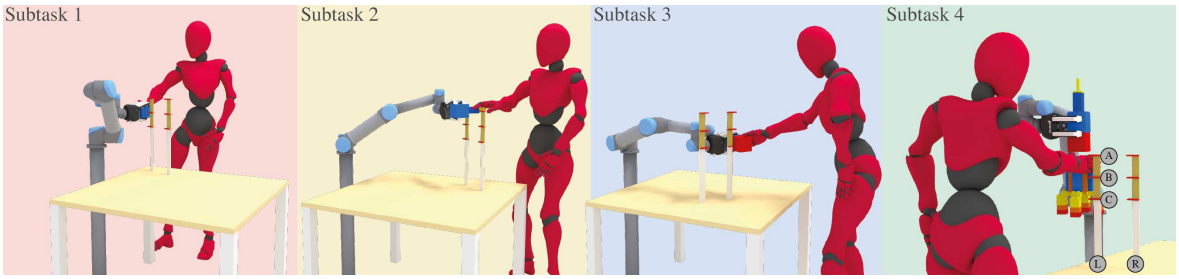


Fig. 6.7.: The rendering shows 4 stages for the rocket assembly task. The joint interaction has been recorded 14 times for 3 heights (A,B,C) at two positions (L,R) on the table each approximately 20 cm apart. For training the system a single demonstration with medium height B at location R was utilized.

6.3.1. Experimental Setup

In general, no specific motion capture hardware is assumed and several have been tested with interaction models successfully. In the experiments presented in this chapter a tracking system by A.R.T. is used. During human-human demonstrations and human-robot interactions, each human wears six markers ($N^o = 6$, $N^c = 6$). In the considered human-robot collaboration tasks marker-based tracking outperformed other capturing solutions, i.e. Kinect depth sensors, in terms of reliability and accuracy and was, in consequence, used in the experiments. Below, the main focus is on the Lego rocket assembly interaction. Similar results have also been obtained for the tube assembly, but they are omitted for the sake of presentation and readability. In order to evaluate the generalization capabilities of the system, the rocket assembly task was recorded 14 times at different positions (see Fig. 6.7) resulting in 41000 motion capture frames (approx. 22 min). Using only a single demonstration of each assembly step to learn the interaction model, 13 repetitions of the joint task are available as validation data. Given these as input, a robot's response is compared with that of a human interaction partner. Based on the training demonstration the preprocessing step created a significantly reduced state space $\mathcal{G}$ from initially 18 dimensions to 6 and 5 dimensions for the Lego rocket task and the tube assembly, respectively. The dimensionality of $\mathcal{L}_i$, i.e. the local posture space for temporal alignment, was 4 for both assembly tasks. Fig. 6.8 and Fig. 6.9 illustrate $\mathcal{G}$ and $\mathcal{L}_i$ that has been selected based on the user's motion (red trajectory) during an interaction.

Approximating the density of user motions in $\mathcal{G}$ using KDE yielded $K = 15$ and $K = 10$ GMM kernels which provided reliable results while compressing the motion to the most relevant key poses (see Fig. 6.10). An HMM with 4 hidden states was created, corresponding to the 4 subtasks of the complex assembly. Segment matching as well as temporal pose matching using DTW in $\mathcal{L}_i$ was performed with a pose history of $H = 30$, resulting in a temporal history of approximately one second. This differs from other approaches where each pose is optimized based on a single observation and, in doing so, temporal and contextual importances of joints are neglected.

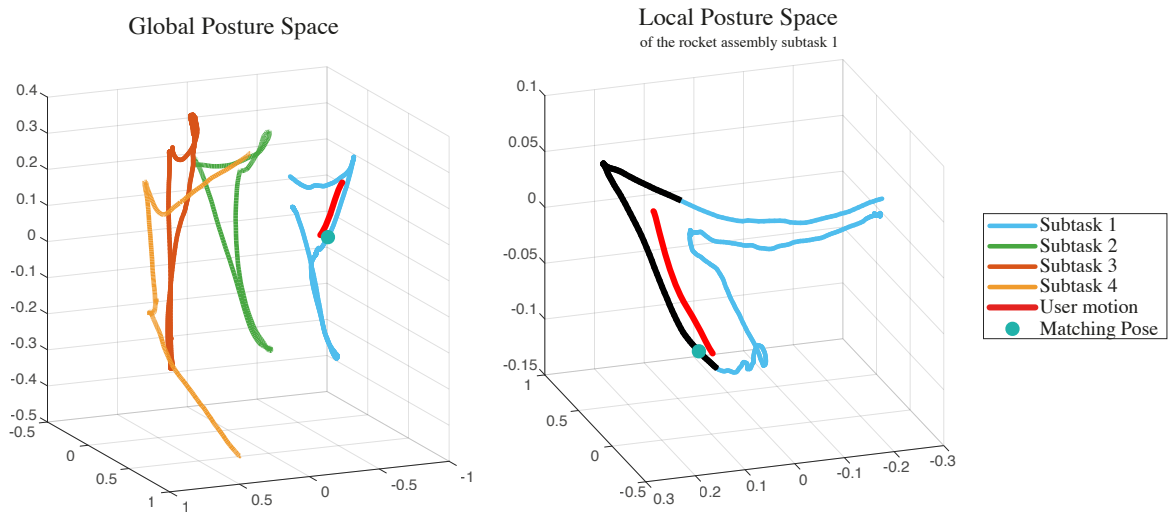


Fig. 6.8.: Left: The global posture space of the Lego assembly task is illustrated. Here, the first 3 PCs are shown. Based on the posterior state probability of the HMM, given the user's motion (red trajectory), a suitable interaction demonstration is inferred. From the selected demonstration the corresponding local posture space is shown on the right. Temporally aligning the observed motion to a matching motion segment using DTW yields a time frame from the initial recording that best fits the current situation (highlighted blue).

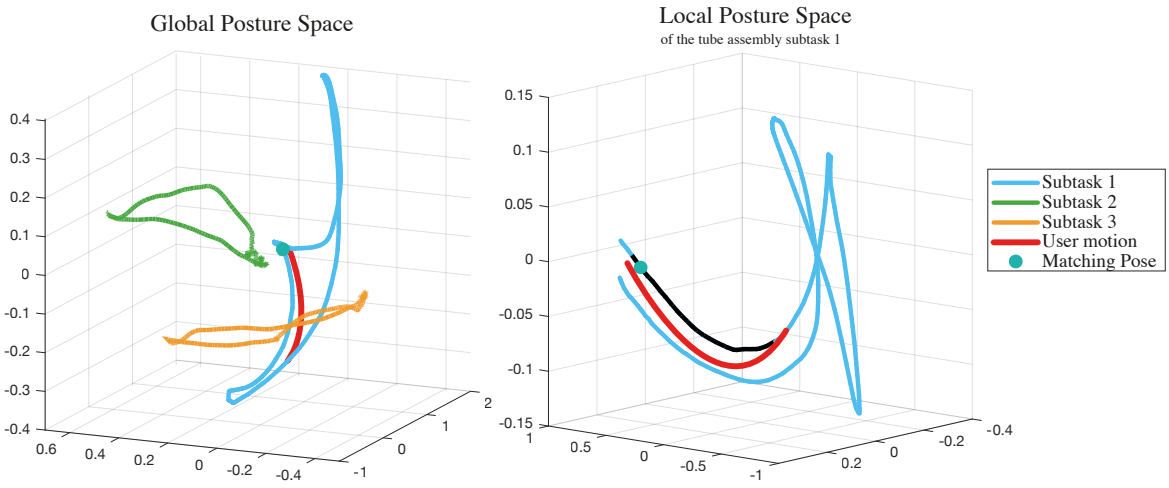


Fig. 6.9.: Left: The first 3 PCs of the global posture space for the tube assembly task are shown. Right: The local posture space of the first subtask is illustrated for the first 3 PCs. The black trajectory corresponds to the matchting motion segment that is used to temporall align the user motion (red trajectory).

6.3.2. Interaction Selection

Selecting an appropriate interaction demonstration from the pool of all interactions is the first step of the interaction model methodology. During runtime the user's motion is reduced in dimensionality and the corresponding key postures are inferred based on Gaussian distributions. As a result a list of key poses of the initial recording are created that resemble how the user moved in the current interaction. The confusion matrix in Tab. 6.1 compares the classification accuracy using HMMs (green values)

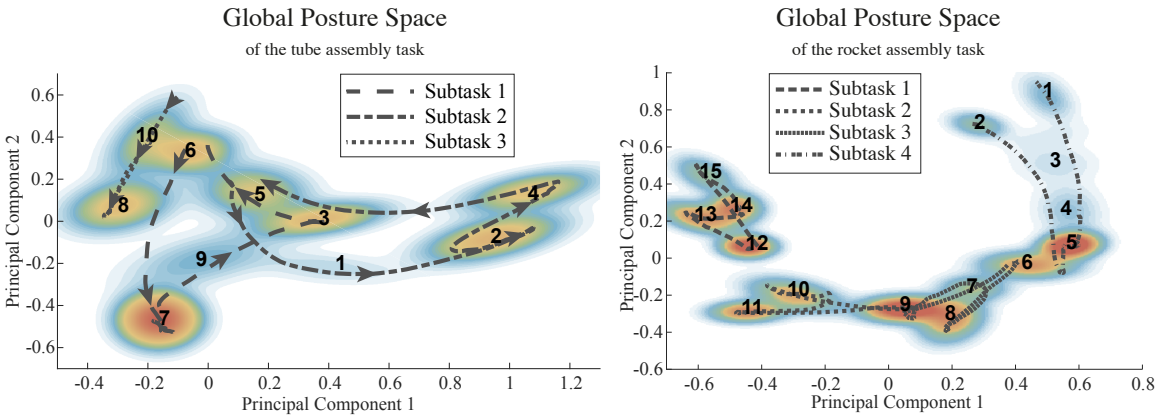


Fig. 6.10.: The figure illustrates kernel density estimates ($K = 10$, $K = 15$) in the global posture space of both interaction scenarios. Each Gaussian thereby resembles a key pose and their sequential order is captured in an HMM to infer suitable interaction demonstrations during runtime. Color is used to indicate the probability of each distribution.

Tab. 6.1.: Confusion matrix of the proportion of correct guesses using **HMMs** and *Euclidean distances (DSTC)*. Color is used to indicate correct/incorrect classifications. The subtasks correspond recorded interaction demonstration of the rocket assembly example (see Fig. 6.7)

		Actual Class			
	HMM/*DSTC*	Subtask 1	Subtask 2	Subtask 3	Subtask 4
Predicted Class	Subtask 1	**0.94** *0.15*	**0.07** *0.42*	**0.04** *0.29*	**0.00** *0.38*
	Subtask 2	**0.00** *0.00*	**0.93** *0.05*	**0.00** *0.15*	**0.00** *0.42*
	Subtask 3	**0.00** *0.22*	**0.00** *0.18*	**0.96** *0.27*	**0.06** *0.25*
	Subtask 4	**0.06** *0.64*	**0.00** *0.35*	**0.00** *0.29*	**0.94** *0.05*

with the k-d tree approach (using Euclidean distances in high-dimensional space) as employed in [39]. As indicated the overall number of false classifications is significantly higher using the methodology in [39].

As poses at the beginning and the end of each interaction are similar (both arms resting aside), without incorporating past poses as context, selections of the current active interaction are unreliable. Using the HMM, sequential information of key poses is inherently incorporated and, thus, allows selection on a broader contextual level. This, in turn, has a strong influence on the selection hysteresis, i.e. the robot's tendency to remain committed to an interaction where the current pose of the human by itself might indicate a different interaction. In the Lego rocket assembly for example, each interaction requires the robot to hold the rocket differently despite that each starts with similar user motions. Without sequential information, i.e. by using the distance-based selection method for example (see Chapter 4.4.1), the robot would switch between interactions and consequently its response poses. As the result the robot would continuously adapt and align the partially built Lego rocket even when the user is not moving.

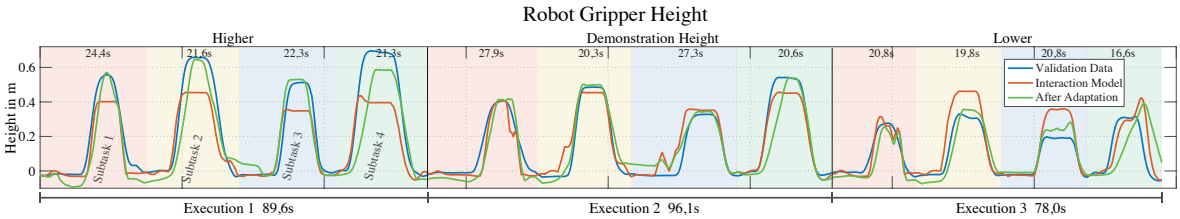


Fig. 6.11.: An important feature of interaction models is the generalization of learned behaviors to new situations. To test the generalization capabilities of the system several repetitions of the rocket assembly task are recorded. The figure shows 3 variations with differing hand-over heights. On the left, input user motions significantly higher than the original recording are shown. In the middle hand-over heights similar to the training data and on the right lower hand-over heights are illustrated. The figure depicts how a frame from the interaction model (red trajectory) is adapted to the new situation (blue trajectory). The green trajectory shows gripper heights after adaptation. The presented approach successfully generalizes up to −25 cm to 25 cm in height, outperforming approaches with conventional, not context-based IMs topologies by a large margin.

6.3.3. Spatial Generalization

Spatial generalization was evaluated for the Lego assembly task using one motion captured demonstration as well as additional 13 motion capture recordings of human-human interactions as validation data. Between the 13 task executions, the handover positions of the manipulated object were varied both in height (see Fig. 6.7 A,B,C) and location (L,R). In a simulation environment, the recorded motions of the observed agent were applied to a simulated human while the responses of a simulated robot were computed using the interaction model. The simulated robot's motions were then compared to the motions of the human assistant in the validation cases. Fig. 6.11 depicts the robot's response in three executions of the assembly tasks. In the figure, blue trajectories depict the height of the human hand during validation while red trajectories show hand hight in the demonstration used to train the interaction model. The robot's adapted gripper height is shown in green. In almost all examples, the robot optimized its position to match the human's hand and reached heights similar to the validation data. However, in some situations spatial adaptation was insufficient. E.g. in execution 1, interaction 4, the robot only adapted to a height of about 19 cm where 27 cm would have been required. In all other examples, the robot adapted its behavior so that the interaction could be completed successfully, i.e. the object was assembled jointly.

Fig. 6.12 illustrates the variance in spatial generalization for different net topology generation methods. Compared to alternative topologies, context-dependent IMs exhibit the largest variance and, thus, offer spatial generalization to a larger range of positions. Since traditional IMs do not focus on important joints, varying user torso rotations force the reconstruction to adapt and change robot hand positions (−10 cm to 10 cm) even when user hand positions do not change.

Further, situations emerged where adaptation of more than 10 cm was required, i.e. height A and C in Fig. 6.7. Here, nets generated with Delaunay triangulation and fully connected topologies did not provide the required degree of adaptation prompting the user to adapt to the robot instead. This unnecessarily requires users to match the

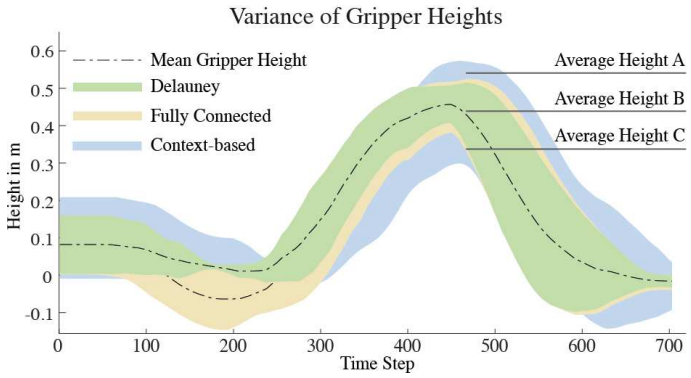


Fig. 6.12.: The figure shows heights that can be generalized to using IM with different topology generation methods. Delauney triangulation (green area) only allows hand-overs differing up to 10 cm in height. Using the data-driven topology generation scheme of the interaction model (blue area) context-sensitive edges and weights are computed. These allow for much wider postural generalization of up to 25 cm. Interestingly, fully connected topologies (yellow area) provide better generalization than nets created with Delauney triangulation.

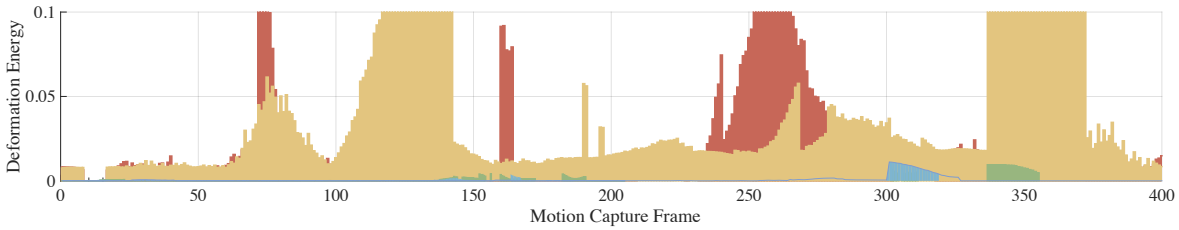


Fig. 6.13.: The deformation before (orange, red) and after optimization (green, blue) for a box lifting and tube assembly task. The deformation energy is computed in Laplacian space and corresponds to the difference between the users current pose and the closest matching motion capture frame from the initial recording. As it can be seen optimizing the posture to the current situation minimizes the postural differences drastically. However, poses during the beginning of the interaction already exhibited a small deformation, thus, less optimization is required.

original motion capture recording closely, in order to interact with the robot successfully. The context-dependent topology generation approach in interaction models on the other hand weight each marker based on its contribution to the overall motion and adapts only relevant joints. This allows users a broader range of movement and a more natural interaction.

For the tube assembly task, Fig. 6.13 illustrates the evolution of the IM deformation energy during two different interactions. As can be seen, postures at the beginning of the respective interactions do not differ much and consequently less optimization is required. This changes during the course of the interaction when more optimization is needed to adapt the IM to the current situation. This is due to the fact that each interaction started with similar poses but differed more significantly from there.

Reconstruction of robot postures based on optimized IMs is achieved with a physics-based inverse kinematics solver. For that, joints are modeled using forces in a simulated environment to stretch the robots skeleton to desired locations, cf. [121]. In doing so, poles are easily dealt with and the robot can be controlled with respect to its force limits. This approach for joint angle reconstruction, however, introduces an error since joints can be stretched when pushed to the limit of the simulated robot's reach. As

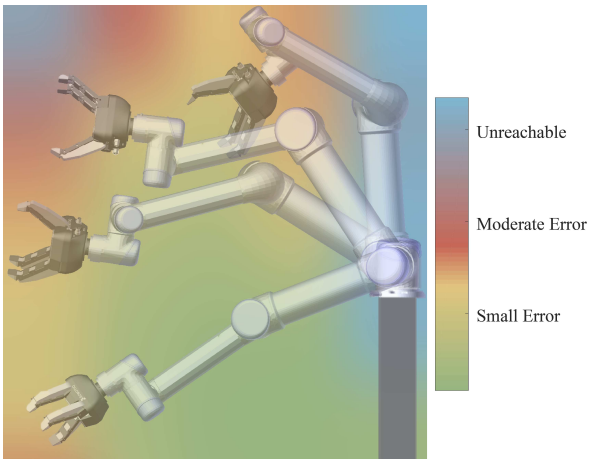


Fig. 6.14.: The difference between the TCP of the real-world robot and the computed physics-based inverse kinematics solution is illustrated. Red and blue correspond to end effector locations that are hard to reach (red) or impossible to move to (blue). Green areas indicate where computed TCPs and the robots end effector are equal with respect to their position and orientation.

a result the tool center point coordinate of the computed and real world robot differs as illustrated in Fig. 6.14. Here, blue and red regions indicate target coordinates which could not be reached due to a large error ($\epsilon > 50$mm). Green and yellow areas correspond to poses with error below 15mm.

6.3.4. Temporal Generalization

Temporal generalization is achieved using a two stage process. First, the user's motion is matched against interaction demonstrations using the HMM and, then, aligned locally in $\mathcal{L}_i$ using DTW. Fig. 6.11 shows three repetitions of the Lego assembly task with varying execution speeds, with a time difference between the slowest and fastest task completion of $\sim$20 s. The actual execution time in the validation cases is indicated by the blue trajectory. As can be seen in the figure, the shapes of the red trajectory (selection of an appropriate IM) and the green trajectory (adaption of the selected IM to the current situation) closely match the shape of the blue trajectory. This shows that the context-dependent generation approach is able to maintain a close temporal synchrony between the movements of the human and the robot even if the task execution time is quite different from the training example.

In a similar vein, Fig. 6.15 shows the temporal generalization of three human-robot interactions of the Lego rocket assembly tasks. Here, each behavior varied in length. Nevertheless, the correct interaction demonstration has been selected and the motion has been temporally aligned, so that appropriate IM could be selected.

6.3.5. Computational Performance

As context-dependent IMs use a sparse set of motion capture markers instead of a comprehensive set of human joints as vertices, an inverse kinematics solver must be employed to compute the joint angles of the robot's target pose. Using inverse kine-

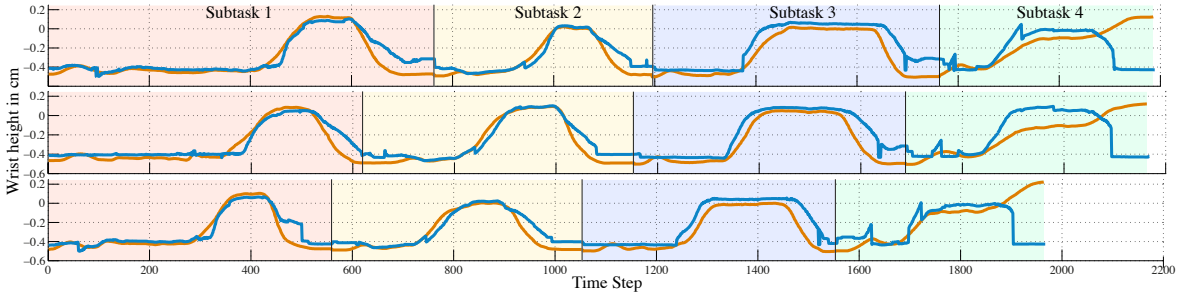


Fig. 6.15.: The figure shows the z-position of the user (orange trajectory) and the robot (blue trajectory). Color is used to highlight the different subtasks of the behavior. As it can be seen the robot extracted the temporal context of the interaction successfully and moved its gripper at the right time.

matics solvers circumvents the challenge of mapping human data onto a robot (the correspondence problem) and ensures that joint limits are not violated.

On average, computing the final configuration of the robot using inverse kinematics accounts for 50 % of the computation time (16 ms) whereas inferring an interaction and computing suitable responses using context-based IMs requires an additional 17 ms[1]. Using the proposed methodology the robot is continuously controlled with a latency of approximately 150 ms towards observed user poses. This results from the employed sliding window that is used to temporally align user motions as well as communication overheads.

6.4. Discussion

Training interactive robots by providing human-human interactions as demonstrated above is a promising approach towards the specification of interaction dynamics. The experiments, however, have also revealed various technical challenges. For example, the availability of human-human recordings is very limited and most databases include only single person movements. Also, recording motion capture data with two humans is challenging, due to self-occlusions and line-of-sight problems.

The experimental results show a reasonable ability of learned interaction models to generalize to spatial and temporal changes. However, the system cannot deal with large spatial adaptations. In these cases, generalization to new positions cannot be achieved successfully without violating constraints. Overall, spatial adaptations within −25 cm to 25 cm of the initial demonstration are feasible in most cases.

A general insight of the experiments is that human-robot interaction can greatly benefit from mesh-based spatial representations. The relationship between two interaction partners can be modeled as a mesh between joints, which is then analyzed via the graph Laplacian, and other well-established graph-theoretic measures. It has also been shown that the topology of the mesh has a strong influence on the generalization capabilities of demonstrated tasks. Defining correlations among joints explicitly using context-dependent IMs has proven to yield good generalization results.

[1] All experiments were performed on a 2011 MacBook Pro.

The opening and closing of the hands is currently not captured during human-human demonstrations and the UR5 gripper is therefore controlled manually. However, this could potentially be modeled as a binary emission within the HMM.

6.5. Conclusion

In this chapter an application of interaction models for human-robot collaboration has been presented. In contrast to motion generation approaches based on kinesthetic training, the learned interaction models inherently capture the important spatial relationships and temporal synchrony of body movements between two interacting partners. This enables the robot to continuously adapt its behavior, both spatially and temporally, to the ongoing actions of the human interaction partner. Using the HMM-based interaction selection method, the presented approach is therefore well suited for collaborative tasks requiring continuous body movement coordination of a human and a robot.

Similar to experiments in VR context-dependent IMs also provided a reliable degree of spatial adaptation in HRI scenarios. In contrast to character animation however meshes were generated with less vertices and additional motion capture markers on the right hand. This increases spatial generalization and allows the reconstruction of gripper orientations during runtime.

A drawback of the method is, that due to the kinematic chain of the robot an inverse kinematics solver is needed to generate the final joint angles from the more abstract human pose extracted from the IM. However, since the approach is fundamentally based on the reconstruction of a humanoid skeletal structure it has the potential of being applicable to anthropomorphic robots as well. By changing the kinematic chain of the inverse kinematics solver one could create seamless controls for complex articulated humanoids as experiments in virtual reality suggest (see chapter 5).

The current chapter addresses human-robot collaboration in a dyadic fashion involving two interactants. Since objects are not modeled, the robot is not able to detect and account for object rotations during the interaction with the user. This limitation is discussed in the following chapter where interaction models are extended to triadic settings. Here, objects are explicitly integrated into the response generation scheme, allowing seamless adaptations to varying object rotations.

7. Triadic Human-Robot Interactions

This chapter presents an interaction model variant specifically targeting human-robot handovers. Whereas the previous models focused on dyadic interactions, i.e. collaborations involving two interactants, the following introduces an extension to triadic and possibly n-adic settings. In addition to the interaction dynamics between both interactants, the triadic variant of an interaction model also captures the object that is passed, allowing for fine-grained adaptation of a robot's gripper during runtime. As a result, more intuitive and time-efficient human-robot handovers emerge.

Fig. 7.1.: A robotic assistant retrieves an object from the user in a typical handover interaction. The triadic interaction model methodology allows robots to engage in such by extracting relevant information about body synchrony and spatial relationships from human-human demonstration. As a result, seamless and natural handovers with the robot emerge, evoking the same effortlessness and ease to which people are so accustomed to from interactions with other humans.

7.1. Introduction

Handing over an object to another person is arguably one of the most essential physical interaction skills. Independently of whether people are at home, in the workplace, at a restaurant, or at the hospital, they are often faced with situations in which they either receive or hand-over an object to another person. Hence, for robots to be reliably used as assistants to humans, they have to be able to engage in similar interactions and deal with the large variability inherent to such handover tasks. Hand-overs are joint

tasks in which the *giver* and *receiver* coordinate their movements in order to ensure the successful transition of the object from one to the other. This requires the interaction partners to react and adapt to each others' movement, timing, style, and posture.

With the advent of collaborative robots, research on human-robot handovers has found increased interest in the robotics community. Various strategies for specifying and learning such behavior have been put forward, such as in [45, 122]. While these approaches have produced important insights, they mostly model human-robot handover as a *dyadic* interaction process – the process parameters are solely influenced by the two interaction partners and not the handled object. However, especially in situations in which an object is handed *from* a human *to* a robot, it is important to incorporate the object as an additional element in the interaction process. In addition, the majority of approaches focuses on the spatial relationship of the end-effectors during the task and only the position of the human hand is used to identify the robot's response.

In this chapter, a triadic interaction model variant is proposed that captures both interactants *and* a manipulated object for seamless human-robot handovers. In particular, the focus is on scenarios in which the robot receives an object from a human partner. Given a single demonstration, information about the synchrony in movement between different body parts of the two interactants, spatial relationships between interactants and the object at different moments in the interaction as well information about object possession is extracted. In turn, these parameters are used to synthesize similar behavior during human-robot handovers. In contrast to previous chapters, the triadic interaction model constantly tracks both, the human interaction partner *and* the manipulated object. This allows for fine-grained adaptation of the robot's end-effector during the course of an interaction with a human partner.

The main contributions of this chapter can be summarized as follows:

1. A one-shot learning methodology for seamless human-robot handovers based on interaction models.

2. Generation of triadic Interaction Meshes that incorporate spatial relationships between interactants and a manipulated object.

3. Automated extraction of body synchrony and object state parameters from human-human demonstrations.

The method builds upon on earlier findings from previous chapters (see Chapter 5 and 6) and extends them to triadic setups. The remainder of the chapter will introduce relevant related work, present the interaction model for triadic interactions, and perform a set of experiments to evaluate the methodology using objective and subjective measures.

7.2. Related Work on Human-Robot Handover

A variety of approaches have been used in the past to study handovers between humans and robots. A common methodology is to perform user studies so as to extract

basic insights and design recommendations for how to implement such behaviors. For example, [123] identified that deliberately adding delays to handover tasks can increase human awareness and success. In [124], a number of both physical and social-cognitive aspects were studied in order to provide recommendations for plausible handover policies. The work in [125] focuses mostly on the importance of gaze as a cue for the projected handover location. Such non-verbal cues can help align the intentions of the robot and the human partner. Intention inference and legibility of motion has been addressed in the works by Dragan and colleagues [126]. In a similar vein, Dehais et al. [127] argue for the importance of the robot's posture during handovers. In particular, legible and human-like motions are preferred over goal-oriented and time-efficient behavior, since they help enable shared mental models between the partners.

Another line of research focuses on designing path-planning algorithms and heuristics that are specific to handover tasks. The work in [128], for example, generates robot plans so as to increase safety and comfort of the human. To this end, the human is treated as part of the planning process. Optimal handover locations are, in turn, planned by taking both the perspective of the robot as well as that of the human into account. Such an increase in state-space, however, also comes with increased computational costs. Another planning-based approach is presented in the work by Quispe et al. [129]. The authors identify optimal handover locations by searching for high manipulability positions for both the giver and the receiver. Ideally, both giver and receiver should have a high manipulability score during the object exchange.

A third line of research focuses on machine learning approaches for identifying optimal handover parameters. Kupcsik et al. [130] use noisy human feedback to infer a latent reward function. In turn, the estimated reward function is used within a policy search algorithm in order to learn parameters for motor primitives. The work in [131] discusses the use of interaction primitives [44] in order to realize handovers during collaborative manufacturing tasks. Learning is performed by observing the dyadic interaction between two humans. Action generation is achieved using Bayesian inference in a probabilistic model. However, since the object is not part of the model, important elements of the interaction are missing. In fact the majority of approaches for human-robot handovers do not actively incorporate the object into the action generation process. It is argued, however, that the inclusion of the manipulated object can have important benefits for modeling the interaction process. Indeed, the pose of the object is a clear cue for the state of the interaction. In the remainder of this chapter, it will be discussed how objects can be incorporated into the handover-generation process using interaction models.

7.3. Interaction Models for Triadic Human-Robot Handovers

The methodology in this chapter follows the interaction model approach and generalizes it to scenarios that involve two interaction partners and a manipulated object. Similar

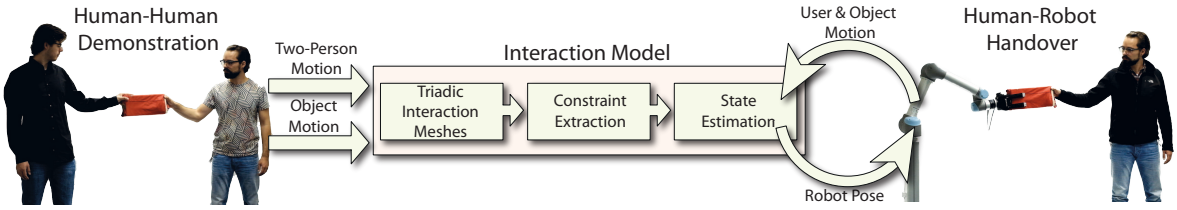


Fig. 7.2.: Based on a single human-human handover demonstration an interaction model is computed that captures spatial and temporal dynamics of both interactants *and* the additional object. Spatial relationships are encoded in a so called triadic IM at each frame of the recording. Utilizing the model during runtime, the state of the interaction is continuously infered, yielding a best matching IM before a robot's response is seamlessly generated.

to dyadic settings, motion capture is used to record *human-human* demonstrations of two users demonstrating a handover. In addition, the movement of the manipulated object is also recorded and tracked.

Given this recording, an interaction model is extracted that describes how the two interaction partners synchronized their movements w.r.t. each other and the manipulated object. In contrast to dyadic interaction models, the triadic variant is generated from a single interaction demonstration without any subtasks. As a result, the global posture space features a single trajectory, rendering interaction selection during runtime unnecessary. The triadic interaction model is thus, a local posture space of the initial handover demonstration and a set of *triadic interaction meshes* that model spatial relationships of the involved interactants. At runtime, the extracted model is used to continuously recognize the state of the handover and adapt the robot's movements to the human interactant (see Fig. 7.2).

In the remainder of this chapter, each step will be described in more detail. Analogous to dyadic interaction models, the first interaction partner, i.e., the human, is referred to as the *observed agent*, while the second interaction partner, i.e., the robot, will be called the *controlled agent*. The third entity in the triadic interaction is the *passive object*. Variables with superscript $(\cdot)^{oa}$ denote the observed agent, $(\cdot)^{ca}$ the controlled agent and $(\cdot)^{po}$ the passive object.

7.3.1. Data Recording

Data for learning is recorded using a setup involving multiple Microsoft Kinect time of flight cameras and the Microsoft Kinect SDK. 8 body part positions are recorded per person ($M = 8, N = 8$)[1] and 7 points on the object ($U = 7$) using depth and color imaging. The motion capture data of the agents is represented as a time series of poses $\mathbf{p}^{oa}_{1:T} = [\mathbf{p}^{oa}_1, \mathbf{p}^{oa}_2, \ldots, \mathbf{p}^{oa}_T]$ and $\mathbf{p}^{ca}_{1:T} = [\mathbf{p}^{ca}_1, \mathbf{p}^{ca}_2, \ldots, \mathbf{p}^{ca}_T]$ respectively. Poses $\mathbf{p}^{oa}_t$ and $\mathbf{p}^{ca}_t$ are composed of M and N joints, yielding $\mathbf{p}^{oa}_t \in \mathbb{R}^{3\mathrm{M}}$ and $\mathbf{p}^{ca}_t \in \mathbb{R}^{3\mathrm{N}}$ for each time step $t \in \{1, \ldots, T\}$, where T is the number of frames in the recording. In a similar fashion, the motion of the object is denoted by $\mathbf{p}^{po}_{1:T} = [\mathbf{p}^{po}_1, \mathbf{p}^{po}_2, \ldots, \mathbf{p}^{po}_T]$ with $\mathbf{p}^{po}_t \in \mathbb{R}^{3\mathrm{U}}$.

[1] For each agent feet, hands, elbows, head and pelvis are captured.

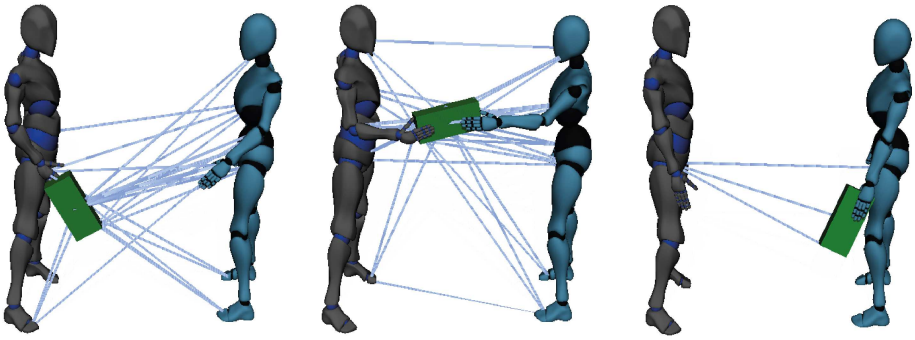


Fig. 7.3.: Based on human-human demonstration triadic IMs are created. The figure shows three stages of a handover and the generated IM topology.

7.3.2. Triadic Interaction Meshes for Human-Robot Handovers

In order to model the spatial relationships between the interactants and the object in a human-robot handover, a mesh triangulation over the involved body parts and object vertices is used. The previous chapters have shown that IMs are suitable for adapting recorded motions spatially to new situations. So far, however, objects are not considered in the IM generation scheme.

In this chapter it is argued that for handovers objects mediate the interaction and that they play a vital role for the success of the collaboration task. Towards that end, a data-driven IM topology generation method is developed that extends context-dependent IMs to triadic interactions involving two agents and an object. A depiction of such triadic IMs at different moments of an interaction can be found in Fig. 7.3. Nodes in an IM represent joints of the human body and vertices on the object surface. Connections between these nodes are generated at different time steps based on the synchrony in movement as indicated by the corresponding covariance.

Generation of triadic IMs is performed by evaluating the pairwise covariances between both users and the object at every frame with a sliding window of size S:

$$\begin{aligned} \mathrm{cov}(\mathbf{p}^{oa,i}, \mathbf{p}^{po,j}) &= \mathbf{E}[(\mathbf{p}^{oa,i} - \mathbf{E}(\mathbf{p}^{oa,i}))(\mathbf{p}^{po,j} - \mathbf{E}(\mathbf{p}^{po,j}))] \\ \mathrm{cov}(\mathbf{p}^{ca,k}, \mathbf{p}^{po,j}) &= \mathbf{E}[(\mathbf{p}^{ca,k} - \mathbf{E}(\mathbf{p}^{ca,k}))(\mathbf{p}^{po,j} - \mathbf{E}(\mathbf{p}^{po,j}))] \end{aligned} \tag{7.1}$$

where $\mathbf{p}^{ca,i} \in \mathbb{R}^3$ is the position of a joint i of the controlled agent, $\mathbf{p}^{oa,k} \in \mathbb{R}^3$ the position of a joint k of the observed agent, and $\mathbf{p}^{po,j} \in \mathbb{R}^3$ a point on the tracked object. Similar to the dyadic HRI setting in the previous chapter, a window size of $S = 30$ points or 1 s yields reliable results while providing enough spatial variability during runtime. Fig. 7.4 depicts the pairwise covariances between the hands of two interactants and the object over 30 trials. In all of the trials the handover starts with a bell-shaped trend in correlation which then decreases to zero during the actual transition of the object between the giver and the taker. In the retraction phase, the covariances describe a similar bell-shaped pattern but have different signs.

The IM topology $\mathcal{T}_t$ at timestep t is constructed by successively adding tetrahedra, where each tetrahedron connects one joint of the controlled agent, two points on the

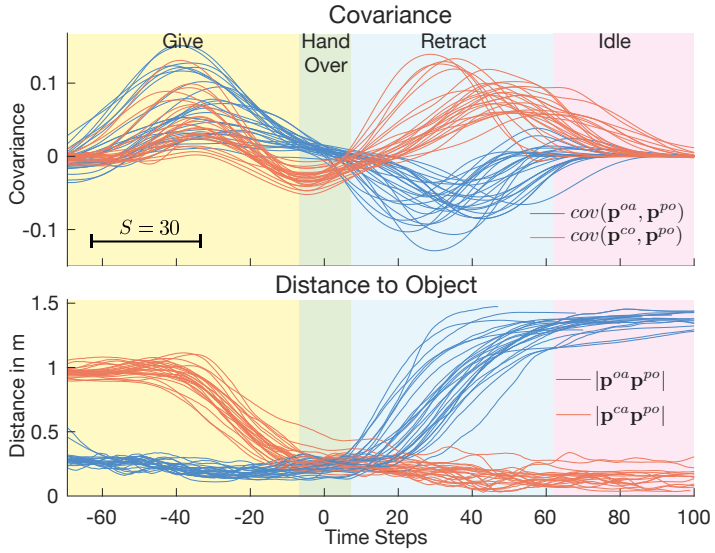


Fig. 7.4.: Top: The covariance between the hands of both users with respect to the object's motion for 30 handovers is illustrated. The symmetry of the changes in the covariance indicates body synchrony between both interactants and it is argued that by retaining this feature during human-robot handover more natural and intuitive interactions take place. Bottom: The distance between the closest pair of joints of both users towards the object is shown. First, the object is passed to the robot before it is handed over (yellow and green area). The blue area indicates a retracting motion, where the arm is moving towards its idle pose. In all recordings the receiving human anticipated the interaction and started moving towards the object as soon as the passing motion was initiated.

object and one joint of the observed agent. Concretely, the joint/point pair with the maximal covariance between the controlled agent and the object is computed:

$$i_{max}, j_{max} = \underset{(i,j)}{\arg\max}\ \mathrm{cov}(\mathbf{p}^{ca,i}, \mathbf{p}^{po,j}) \tag{7.2}$$

Similarly, the pair with maximal covariance between the observed agent and the object is selected, where the point on the object must not be the same as the point chosen in the above pair:

$$k_{max}, l_{max} = \underset{(k,l \neq j_{max})}{\arg\max}\ \mathrm{cov}(\mathbf{p}^{oa,k}, \mathbf{p}^{po,l}) \tag{7.3}$$

The tetrahedron $T = (i_{max}, k_{max}, j_{max}, l_{max})$ is then added to the IM topology. Note that by including two distinct points of the object in the tetrahedron, the object's orientation is implicitly represented in the IM.

The process of adding tetrahedra to the IM is repeated, with previously used joint/point pairs being excluded from further consideration, and as long as a pair $(\mathbf{p}^{ca,i}, \mathbf{p}^{po,j})$ with covariance above threshold Ψ can be found. Since a good choice Ψ depends on the velocity of the motion and the frame rate with which the behavior is captured, it has to be set based on the individual setup. Also, Ψ has a strong influence on the density of the topology. As dense connection structures hinder mesh deformation during optimization sparse topologies are desired (see chapter 5). However, sparse structures potentially fail to preserve spatial relationships to a reasonable degree, forcing the robot to misalign its gripper. In the experiments a value of $\Psi = 0.1$ resulted

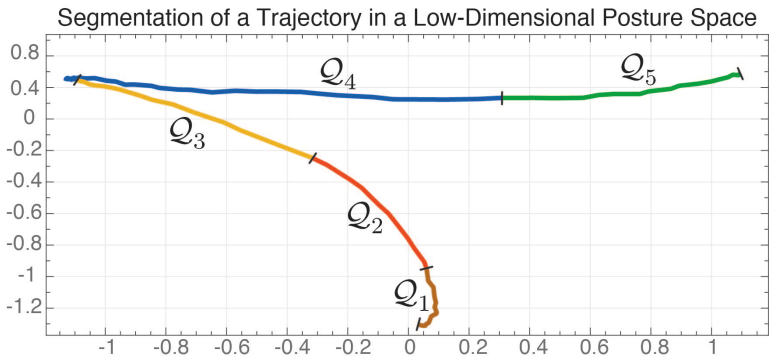


Fig. 7.5.: The observed agent's motion during the handover demonstration is projected into the local posture space. In order to account for varying joint importances during different phases of the interaction, the low-dimensional motion trajectory is further segmented where each segment is associated with a different set of constraints (see text). The figure illustrates the first two principal components of a low-dimensional handover trajectory divided into five segments.

in IMs composed of $E = 1$ to $E = 12$ tetrahedra per timestep (see Fig. 7.3), which produced reliable spatial adaptability towards unseen user motions.

While the approach is in principle able to work with just one joint from each agent and two object vertices, IM topologies with several tetrahedra are generally advantageous. One reason is that additional information on how to approach the object can be provided to the robot, e.g. when not only the desired hand position is provided but also an elbow position. Another reason is the increased robustness against tracking errors at runtime by making the system less dependent on the accurate tracking of a single body part.

7.3.3. Data-Driven Triadic Constraint Extraction

Hard and soft constraints are defined in order to inform the runtime optimization process about the degree to which different parts of an IM may be deformed to match the current situation. In human-robot interaction settings, as opposed e.g. to offline generation of computer graphics animations, the poses of the human are not under control of the optimization algorithm. This can be modeled with hard positional constraints. Soft constraints can be used to control the spatial adaptivity of the respective joints of the controlled agent (robot) in the handover task. As a particularity of the handover task, the ownership of the object being handed over changes during the triadic interaction, thus requiring changing constraint definitions between different phases of the interaction. Traditional IM approaches either assume manual editing of positional constraints [39] or constant constraints [59] for the whole interaction. In triadic IMs however, an automatic constraint extraction process similar to context-dependent IM is used and constraints may vary over time.

In order to distinguish between different phases of the handover task, the example interaction is projected into a local posture space where the resulting motion trajectory is segmented into different parts (see chapter 4). Fig. 7.5 shows an example low-dimensional motion trajectory and its division into five segments $\mathcal{Q}_1 \ldots \mathcal{Q}_5$.

For each time step, hard and soft constraints are extracted from the task demonstration based on the respective roles of the two agents and the object as well as the covariances and distances between their joints/points. Covariances are evaluated over motion segments $\mathcal{Q}_k$ whereas distances are evaluated for each time step (see chapter 4.3.3).

Hard constraints are assigned to joints of the observed agent as well as the object while being held by the observed agent. In principle, all joints of the observed agent could be associated with a hard constraint, as none of the joints is under the control of the optimization algorithm. In practice, better results can be achieved by only assigning hard constraints that are important to the interaction, as determined by covariance and distance measures.

A hard constraint $c^{oa,i}$ is assigned to joint i of the observed agent if it exhibits a large covariance to some point j on the object:

$$c^{oa,i=1...M} = \begin{cases} 1 & \text{if } \exists j \, \text{cov}(\mathbf{p}^{oa,i}, \mathbf{p}^{po,j}) \geq \Psi \\ 0 & \text{otherwise} \end{cases} \tag{7.4}$$

Also, a hard constraint $c^{po,i}$ is assigned to point i on the object if it is close[2] to some joint j of the observed agent with which it also has a large covariance:

$$c^{po,i=1...U} = \begin{cases} 1 & \text{if } \exists j \; (\; \text{cov}(\mathbf{p}^{po,i}, \mathbf{p}^{oa,j}) \geq \Psi \; \wedge \; ||\mathbf{p}^{po,i}, \mathbf{p}^{oa,j}|| \leq \Omega \;) \\ 0 & \text{otherwise} \end{cases} \tag{7.5}$$

Soft constraints are assigned to the joints the controlled agent. A soft constraint $f^{ca,i}$ is assigned to joint i of the controlled agent if it is far away and exhibits small covariance with the points on the object:

$$f^{ca,i=1...N} = \begin{cases} 1 & \text{if } \forall j \; (\; \text{cov}(\mathbf{p}^{ca,i}, \mathbf{p}^{po,j}) < \Psi \; \wedge \; ||\mathbf{p}^{ca,i}, \mathbf{p}^{po,j}|| > \Omega \;) \\ 0 & \text{otherwise} \end{cases} \tag{7.6}$$

Additionally, soft constraints are attributed with weights in the range 0 to 1. A weight close to 1 means that the joint will have a strong tendency to reach a position resembling the demonstration, whereas a weight close to 0 means that it has a strong tendency to adapt towards the actual situation. Let i denote a joint of the controlled agent and j the point on the object with the largest covariance to i. Then weight $w^{ca,i}$ is defined as

$$w^{ca,i=1...N} = \begin{cases} 1 - \frac{\text{cov}(\mathbf{p}^{ca,i}, \mathbf{p}^{po,j})}{\sigma(\mathbf{p}^{ca,i})\sigma(\mathbf{p}^{po,j})} & \text{where } f^{c,i}=1 \\ 0 & \text{otherwise} \end{cases} \tag{7.7}$$

For the optimization process described (see chapter 4), hard constraints c, soft constraints f and weights w are expanded into square diagonal matrices $\mathbf{C}$, $\mathbf{F}$ and $\mathbf{W} \in \mathbb{R}^{\mathrm{Z} \times \mathrm{Z}}$ of size $Z = M + N + U$. The first M elements of the diagonals contain constraint information about the observed agent, the next N elements information about

[2] Distances below $\Omega = 0.75\,\text{cm}$ are considered to be relevant for the interaction.

the controlled agent, and the last U elements information about the object:

$$\begin{aligned}\mathbf{C}&=\text{diag}(c^{oa,1},\ldots,c^{oa,M},0,\ldots,0,c^{po,1},\ldots,c^{po,U})\\ \mathbf{F}&=\text{diag}(0,\ldots,0,f^{ca,1},\ldots,f^{ca,N},0,\ldots,0)\\ \mathbf{W}&=\text{diag}(0,\ldots,0,w^{ca,1},\ldots,w^{ca,N},0,\ldots,0)\end{aligned} \tag{7.8}$$

7.3.4. Local Posture Space Generation

During runtime an appropriate IM from the pool of all IMs is to be selected in order to generate the robot's response (see section 7.3.5). For that the interaction is temporally aligned, i.e. the time frame from the initial recording that best fits the current situation, is selected. In contrast to earlier dyadic implementations, where objects are neglected (potentially resulting in crude state estimates during runtime [5, 122]), user motions *and* object trajectories are explicitly included in a triadic interaction model in a joint posture space. This allows more precise state estimates during runtime and provides increased robustness towards occlusions that typically occur during handovers, e.g. when the object occludes the hand.

A joint posture space $\mathcal{L}$ is calculated by applying PCA to the combined motion capture data of the observed agent $\mathbf{p}^{oa}_{1:T}$ and the object $\mathbf{p}^{po}_{1:T}$. After transformation[3] both motions are compactly represented as a single trajectory $\mathbf{p}_{1:T} \in \mathcal{L}$. For calculation of time-dependent constraint definitions (see section 7.3.5), the trajectory in $\mathcal{L}$ is further segmented using Hotellings T-squared statistics (see Fig. 7.5 and chapter 4). A segment $\mathcal{Q}_k$ with $k \in \{1,\ldots,K\}$ is defined as sequence of consecutive points $\mathbf{p}_{r:v} \in \mathcal{L}$ in the joint posture space, where K is the number of segments.

On average 2 to 3 dimensions are required to represent up to 98 percent of the information in the user's and object's tracking data. Based on 30 validation recordings (see Fig. 7.4) $K = 4$ to $K = 6$ segments are typically created.

7.3.5. Generating Robot Responses

During runtime the user's $\hat{\mathbf{p}}^{oa}_{H-S:H}$ as well as the object's motion $\hat{\mathbf{p}}^{po}_{H-S:H}$ is captured, combined into $\hat{\mathbf{p}}_{H-S:H} = [\hat{\mathbf{p}}^{oa}_{H-S:H}, \hat{\mathbf{p}}^{po}_{H-S:H}]$ and projected into the local posture space $\mathcal{L}$. Here, H is an index to the most recent pose in the sliding window of size S of evaluated poses. The resulting trajectory in $\mathcal{L}$ is then matched against segments $\mathcal{Q}_k$ using similarities in the posture space (see chapter 4.3.3). Similarities are only computed for segments whose centroid is in close proximity to the mean of $\hat{\mathbf{p}}_{H-S:H}$. In essence, a high similarity value is assigned to segments that point in the same direction and, thus, describe similar postural changes of the object and the user over time.

A temporally suitable time step of the recording $\hat{t}$ is found by computing a DTW path between the matched $\hat{\mathcal{Q}}_k$ and $\hat{\mathbf{p}}_{H-S:H}$. It is noted that the number of poses in a segment must not necessarily equal the sliding window size S since the used DTW

[3]The transformations of $\mathbf{p}_{1:T}$ into different coordinate systems can be achieved with a single matrix operation and a precise marking will be henceforth omitted for reasons of readability. Instead the corresponding space is referenced at each occurrence

implementation is able to account for different sequence lengths [132]. Given $\hat{t}$, a pose $\mathbf{p}_{\hat{t}} = [\mathbf{p}_{\hat{t}}^{oa}, \mathbf{p}_{\hat{t}}^{ca}, \mathbf{p}_{\hat{t}}^{po}]$ from the initial task demonstration, the corresponding IM topology $\mathcal{T}_{\hat{t}}$, the weight matrix $W_{\hat{t}}$ and matrices with hard and soft constraints $C_{\hat{t}}$, $F_{\hat{t}}$ can be retrieved.

Consider the difference between $\mathbf{p}_{\hat{t}}$ from the training recording with the pose in the current situation $\hat{\mathbf{p}}_H = [\hat{\mathbf{p}}_H^{oa}, \hat{\mathbf{p}}_H^{ca}, \hat{\mathbf{p}}_H^{po}]$. In order to adapt the retrieved IM to the current situation, essentially, its deformation energy is minimized while at the same time ensuring the validity of the associated hard constraints. Following earlier chapters, the minimization problem is solved with a sparse system of linear equations

$$\begin{pmatrix} \mathbf{M}_{\hat{t}}^T\mathbf{M}_{\hat{t}} + \mathbf{F}_{\hat{t}}^T\mathbf{W}_{\hat{t}}\mathbf{F}_{\hat{t}} & \mathbf{C}_{\hat{t}}^T \\ \mathbf{C}_{\hat{t}} & 0 \end{pmatrix} \begin{pmatrix} \hat{\mathbf{p}}_H \end{pmatrix} = \begin{pmatrix} \mathbf{M}_{\hat{t}}^T\mathbf{b} + \mathbf{F}_{\hat{t}}^T\mathbf{W}_{\hat{t}}\mathbf{p}_{\hat{t}} \\ \mathbf{p}_{\hat{t}} \end{pmatrix} \tag{7.9}$$

where $M_{\hat{t}}$ and $\mathbf{b}$ are obtained by expanding the Laplacians of (7.9)[4].

Solving the system for $\hat{\mathbf{p}}_H$ yields an adapted pose for the robot $\hat{\mathbf{p}}_H^{ca}$, given the poses of the human interactant and the object. This robot pose is then transformed into joint space using IK. Similar to the dyadic implementations in earlier chapters, IM topologies $\mathcal{T}_t$ are computed at every frame t of the initial motion capture recording in order to create context-sensitive and instant responses.

7.4. Evaluation

In the following section, the proposed methodology is evaluated and compared with other handover approaches reported in the literature in a within-subject user study. In particular, it will be shown how triadic IMs affect spatial and temporal generalization to new situations.

7.4.1. Methods

The triadic interaction model is compared with two state-of-the-art approaches that have recently been proposed. The first comparison is with traditional IMs which uses Delaunay triangulation to generate a mesh out of the poses of virtual characters [39]. During runtime a suitable IM is retrieved by querying a k-d tree, cf. [39].

The second approach for comparison is a responsive control scheme as described in [133]. Huang and colleagues organize poses of the observed agent in a k-d tree. At runtime, the k-d tree is repeatedly queried with user poses, yielding a pair of postures from the initial recording. The pose of the second interactant is then mapped to the robot's kinematic configuration using an inverse kinematics solver. Since no optimization or learning is performed, the k-d tree needs to be seeded with a sufficiently large set of different handovers to achieve generalization. In the presented implementation of

[4]For (7.9), $Z \times Z$ constraint matrices C, F, and W are transformed into $3Z \times 3Z$ shape to account for each Cartesian component of the joint positions separately. The $3Z \times 3Z$ diagonal matrices are created as in (7.8) but with each element being repeated three times.

Fig. 7.6.: Two human users demonstrate a handover interaction. It is composed of a reach/give motion, the actual handover and a retract motion that stops when both arms rest aside.

this approach, a k-d tree is constructed using 30 motion capture recordings at different heights and positions (see Fig. 7.7).

The same dataset of motion capture recordings used throughout the experiment. The movements contain the reach motion, the actual handover, as well as a retraction of the hand (see Fig. 7.6). However, whereas the k-d tree structure from Huang et al. is constructed using 30 motion capture recordings, both IM methods utilize a single task demonstration setting up the interaction mesh. Note that only the triadic IM approach incorporates the object's position during mesh generation. The training example of the IM methods has been randomly selected from the pool of all motion capture recordings with medium handover heights (see Fig. 7.4 and Fig. 7.6). As a general rule however, only smooth data sets should be used for model learning to minimize the risk of jerky robot responses. The triadic IMs are parametrized with $\Psi = 0.1$, $\Omega = 0.75\,\text{cm}$ and $\Phi_k = 0.05$ respectively. This yielded a 2-dimensional low-dimensional space with $K = 5$ segments based on $T = 177$ motion capture frames (see Fig. 7.5).

7.4.2. Measures

To evaluate performance interaction times as well as the number of successful interactions are measured. A handover is considered successful if the robot is able to receive the object without dropping it. To measure subjective user experience and performance, a NASA TLX survey is conducted for each of the tested control schemes.

7.4.3. Procedure

10 participants (all male; students) were recruited, all of whom had previous exposure to robots. Participants were first given 5 min to get familiar with the robot and the task. In the experiments a UR5 arm and a Robotique 3-finger gripper is used. Both are not equipped with a force-torque sensors. Since motion capture cannot reliably detect opening/closing of hands, users were handed a WiiMote bluetooth sender to operate the gripper manually.

In order to familiarize oneself with the gripper operation and experience robot responses beforehand, each subject passed the object several times to the robot, before the experiment started. For that a control scheme was chosen randomly. After familiarization, 3 rounds of interactions were performed, with one round for each control

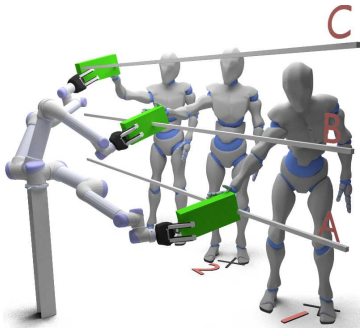


Fig. 7.7.: In the user experiment each participant was tasked to handover a green object at three locations (1,2,3) at three different heights each (A,B,C). This procedure was repeated for all three response generation approaches.

Fig. 7.8.: The figure illustrates 7 handovers. At each location the robot aligned its gripper to match the object's rotation as well as the user posture so that the handover is fluent and natural.

scheme (see section 7.4.1). For each control scheme, each user was tasked to hand a box-shaped object to the robot at 3 different locations and 3 different heights (see Fig. 7.7). During the interaction with the robot, the participants were not told which scheme was active nor did they know which method was developed by the authors. The participants were also told to avoid corrections when the robot is already retracting and when the object is to slip out of the gripper. It was suspected that they would try to catch the object and were thus explicitly told not to engage in these situations for safety reasons.

After each experiment, the participants where asked to fill out a NASA TLX questionnaire with feedback regarding the mental demand, physical demand, temporal demand, performance, effort, and frustration. In total, the experiments lasted approximately 40 min per participant.

Tab. 7.1.: Comparison of successful handovers

	Proposed Method	Ho et al.	Huang et al.
Attempts	83	80	83
Successful	77	33	30

7.4.4. Results

During the course of the experiment 246 handovers where recorded with 83 using triadic IMs, 80 using IM topologies created by Delaunay triangulation[5] [41] and 83 using the approach proposed by [133]. The overall amount of successful interactions is shown in Table 7.1. Naturally, participants handed over the object differently as shown in Fig. 7.9. Still, triadic IMs reliably generated suitable response motions so that the gripper orientation and robot arm motion matched the current situation. This is due

[5] 3 recordings were dropped due to inconsistencies in motion capture readings.

Fig. 7.9.: The figure shows participants of the survey and how they handed the object over. It can be seen that each of them passed the object at different locations and rotations. Still, triadic IMs reliably generated robot responses that were appropriate for each situation.

to the object- and user-aware optimization scheme. The resulting range of motion, i.e. the potential handover location is indicated in Fig. 7.8 for 7 handovers at different positions.

Table 7.1 reveals significant differences in success rates between the compared approaches. Using triadic IMs, the robot was able to rotate and align its gripper so that the object could be grasped tightly in approx. 92 % of the handovers. In the remaining 8 % percent, the object either slipped out of the gripper or dropped immediately. In contrast, using the approaches of [41] and [133], only $\approx$ 40 percent of all interactions succeeded. Since the object is not explicitly modeled there, the robot has no means of detecting object rotations and locations but instead fully relies on the user's posture to determine its gripper pose. However, since each participant handed the object differently to the robot, e.g. by varying height or rotation, the robot misaligned its gripper frequently and the object dropped down. Fig. 7.8 illustrates 7 handovers at different locations using our approach. Similarly, Fig. 7.9 shows the variation in movements and orientations over different users.

7.4.5. Interaction Times

The durations required to hand the object over, i.e. accumulated times needed to reach towards the object, grasp it and then retract back to an rest pose, are depicted in Fig. 7.10 for all 3 methods. During all experiments, the velocity of the UR5 robot was limited to 50 percent of its maximum speed to ensure safety.

Using the triadic IM approach, users needed on average 9.7 s to complete the task, in comparison with approx. 13 s for the other two methods. Hence, the interaction was about $\approx$25 % faster using triadic IMs. In addition, the interaction times exhibited a significantly smaller variance which can be attributed to the time-dependent constraint

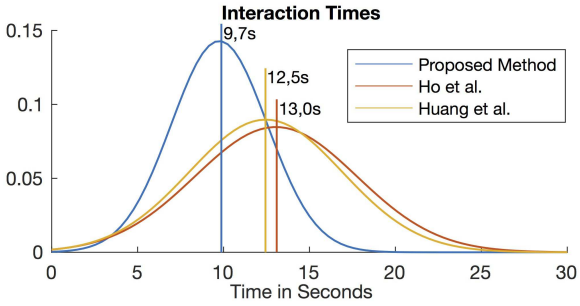


Fig. 7.10.: The graph shows the interaction times for all recorded interactions. On average triadic IMs yield the shortest interaction times (approx. 9.7 s) with the robot while at the same time exhibiting the highest success rate of 97 %. Despite similar task completion times the object has been dropped significantly more often using traditional IMs or a k-d tree resulting in a success rate of 42 % and 36 % respectively.

assignment in the presented approach. In contrast to generic constraints, the data-driven triadic methodology produces constraints that focus on the essential elements of the interaction.

7.4.6. Spatial Generalization

Fig. 7.11 depicts handover locations for all participants and control schemes. Again, differences in variance for these approaches can be observed. Despite occasional occlusions and tracking errors, the proposed triadic method consistently offered a range of ±37 cm based on a single task demonstration, whereas traditional IMs and the k-d tree method allowed for a range of about ±24 cm.

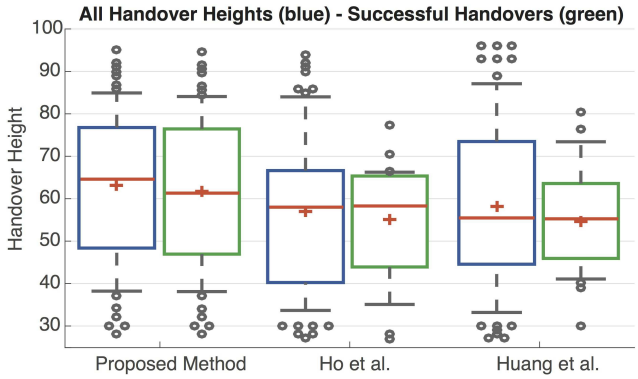


Fig. 7.11.: The figure illustrates handover heights for all three methods (blue all recorded handovers, green successful handovers). The triadic approach offers an increased range of heights for which the robot is able to respond appropriately without dropping the object. This shows that the interaction model method generalizes better to unseen situation.

The increase in spatial generalization results can be explained by the time-varying constraint assignments. During the *giving* motion of the observed agent, the robot roughly follows the initial task demonstration but strongly adapts it behavior to the user's motion (due to active hard constraints). Over time, however, this changes and the robot focuses more on the object rather than the user (due to the added soft

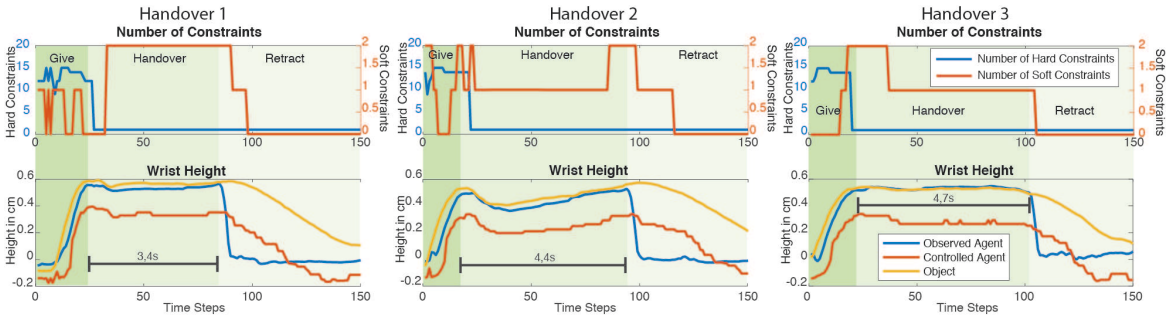


Fig. 7.12.: Top: The diagrams show the number active constraints for three handover interactions. During the *give* phase, the dominating hard constraints cause a strong spatial adaptation of the robot's motion towards the user motion. In contrast, during the final *retract* motion, only few constraints are active which allows the robot to closely follow the task demonstration without much spatial adaption. The *handover* phase is dominated by soft constraints, which indicates that spatial adaption of the robot's motion is a compromise between the task demonstration and the demands of the current situation. Bottom: The plots show wrist height trajectories used for determining the boundaries between the three phases of the handover tasks in the diagrams in the top row.

constraints). As soon as the user retracts, the overall number of constraints immediately drops and the robot can now adopt a behavior that strongly resembles the task demonstration. A visualization of this change in mesh topology is shown in Fig. 7.3.

Fig. 7.12 shows how the number of active constraints changes during the course of three handovers with different speeds and similar heights. The high number of active hard constraints at the beginning of the interaction demonstrates that the robot starts early to synchronize its movements with the user, re-targeting its gripper movement towards the actual object position. In the middle part of the interaction, the number of hard constraints is low while more soft constraints are activated. Accordingly, spatial adaption of the robot's motion is now more balanced between the task demonstration and the actual situation. In the last part of the interaction, due to the lack of active constraints, the robot's gripper essentially follows the spatial trajectory of the demonstration. The number of activated constraints also varies over different users which reflects the different ways the handover movements were performed (cf. Fig. 7.9).

7.4.7. User Experience and Task Performance

After being exposed to all three approaches each participant was surveyed using a NASA TLX questionnaire. The subjective evaluation of robot performance by the users can be seen in Fig. 7.13. They follow the general trend of the experiments – the approaches of [39] and [133] produce approximately similar performance. Triadic IMs, by contrast received significantly better scores by the users. As mentioned before, the users where not informed which method was used at each time. A similar trend can also be found in Fig. 7.14 which depicts the self-reported frustration of the users.

The triadic method also produced low levels of frustration. The methods of Huang et al. and Ho et al. produced higher levels of frustration, with the latter also showing a larger variance in the reported scores.

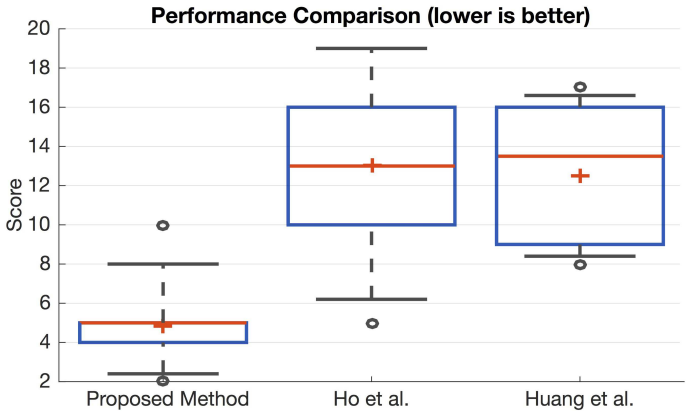


Fig. 7.13.: TLX performance scores indicate that users felt more confident when the robot was controlled using the interaction model method. Interestingly, the variance of individual performances was significantly lower compared to other methods. This indicates a increased confidence and repeatability of triadic IMs.

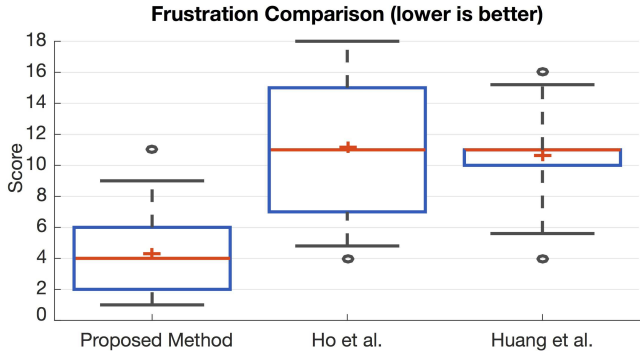


Fig. 7.14.: The figure shows NASA TLX questionnaire frustration levels for all 3 methods. The results indicate that users were more stressed and irritated using traditional IM or the approach from Huang and colleagues.

7.5. Discussion and Limitations

The results of the experiment above validate the hypothesis that incorporating the object into the interaction process can have various benefits w.r.t. interaction times, human frustration, and overall success. However, the presented approach is still only a first step and is limited in some ways.

First, only a limited representation of the object's shape is used. In particular, since all experiments are performed with a box, a small number of vertices was sufficient to represent the shape. It would, however, also be important to explore how object morphology affects the generation of the triadic IM. It is not clear at the moment if a detailed shape needs to be employed or whether a convex hull is sufficient. At the same time, continuous tracking of a complex object also poses various challenges w.r.t. to computer vision.

Further, it should be noted that other methods can be used for estimating the state of the interaction, i.e. the time step from the initial recording that best fits the current situation. In the past, *interaction primitives* [44] and HMMs (see chapter 6) have been successfully employed in HRI to perform state estimation. In a triadic interaction

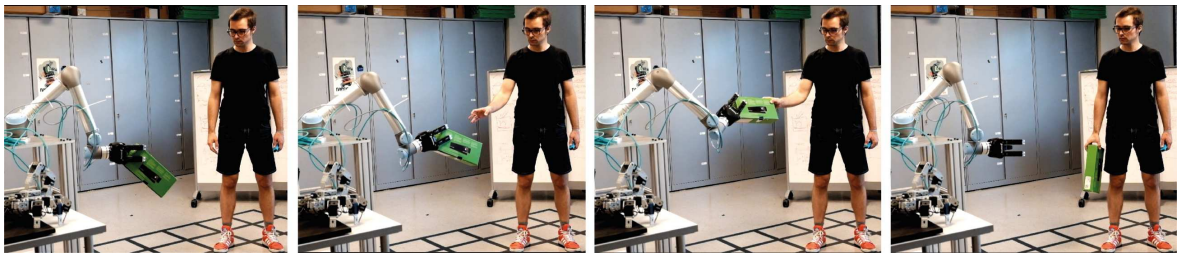


Fig. 7.15.: Triadic interaction models can also be used by the robot to pass object to a human as depicted in the figure. Similar to human-robot handovers, the robot is continuously adapting its pose to match the velocity and position of its interaction partner.

model a PCA-based approach is used due to its low computational demand.

An eminent benefit of the interaction model approach is that it's bidirectional: it can also be used independently of whether the human or the robot are initiating the handover (see Fig. 7.15). For robot-human handovers, recent literature suggests that humans typically prefer standard orientations of objects [134]. Also, the robot is able to rely on the user's ability to adapt during the handover. During human-to-robot handovers however, the robot has to align its gripper to best fit the objects shape and anticipate where the object will be handed over. This requires continuous spatial adaption with respect to the user's *and* the object's motion.

The generalization of a learned handover to different locations was shown to be around $+/-37\,\text{cm}$. For future work, further investigation into how generalization can be extended to larger envelopes is intended. Also, for safety reasons, the speed of the robot during experiment was deliberately limited. Given the encouraging results, future research directions will investigate fast handovers and their requirements.

The experiments where conducted with a simple box shaped object. In order to apply the triadic variant of an interaction models to the assembly tasks from chapter 6 more sophisticated object tracking and recognition capabilities are required.

7.6. Conclusion

The experiments showed that during human-robot handovers object positions and rotations are vital for the success of the interaction. A one-shot learning approach in which the important parameters for a successful handover are extracted in a data-driven way has been presented in this chapter. Whereas current approaches for human-robot handover almost exclusively focus on a single agent, triadic interaction models capture the synchronous movements of two interaction partners as well as a manipulated object. As an extension to the approach presented in chapter 4, triadic interaction models also extract positional constraints for the object that is passed. During runtime the model is employed in a human-robot setting to infer a robots response continuously and seamlessly. In comparison to other robot control schemes the presented interaction model implementation yields more intuitive and time-efficient handovers, while also increasing user satisfaction and overall success ratios.

8. Conclusion

This chapter summarizes and concludes the presented thesis. After highlighting main contributions, key research directions that evolve around interaction learning are discussed.

8.1. Summary and Main Contributions

Robots are more and more entering human workplaces and intuitive programming interfaces are called for in order to enable them to cooperate with people. Current programming-by-demonstration approaches mainly make use of kinesthetic teaching and focus mostly on a single agent. In this thesis however an interaction learning framework has been developed that is based on parallel behavior demonstrations by two interacting partners. An important benefit of the methodology is, that the robot's behaviors are trained by means of two-person motion capture. This not only removes the burden of kinesthetically moving the robot during training but also reduces the cognitive load of the demonstrators and the amount of programming involved.

It was argued that by using two-person task demonstrations, a robot is able to imitate interaction dynamics and body synchrony to provide more natural and intuitive responses during collaboration tasks with users. At the core of the proposed method, an interaction model is generated to capture body synchrony and spatiotemporal characteristics of two demonstrators. During the course of an interaction with a robot, the model is used to spatially adapt the initial two-person recording to match the current situation at each frame. It thereby generates smooth and instant responses that take the specifics of the human interaction partner seamlessly into account. An advantage of the approach is that due to the underlying human-human task demonstrations the robot's motion is human-readable and, thus, leads to intuitive collaborations. As a result of this, the interaction has the potential to evoke similar effortlessness and ease to which humans are so accustomed to from interactions with other people.

From an algorithmic point of view, an interaction model serves state estimation for ongoing interactions and provides data structures for seamless response generation. It is composed of several low-dimensional spaces and a set of context-dependent Interaction Meshes. During the interaction with the robot, the state of the interaction is estimated based on the user's motion in the global posture space. It is generated from all interaction demonstrations and a suitable example is inferred based on similarities of the user's live motion. For that either the distance-based or HMM-based selection approach is used, depending on the complexity and similarity of poses. Situations which

feature ambiguous user postures or require sequential orders of interactions require an HMM in order to yield correct and appropriate robot motions.

Given the observed user postures and a suitable interaction demonstration, the user's motion is temporally aligned in the corresponding local posture space. In the end, temporal alignment yields a pair of poses from the initial two-person recording that resembles the current situation best. One of these is the unadapted response pose of the robot. However, the initial recording certainly differs from what is required in the current context and spatial adaptation of the response is needed. For that the robot's pose is optimized using a novel context-dependent IM structure at each frame of the interaction. The IM captures spatial relationships among body parts and defines positional constraints based on two-person motion capture recordings. The preservation of such is most important during physical human-robot teaming where close contact is often unavoidable. Harnessing human-human task demonstrations, a context-dependent IM focuses on correlating joints and automatically attributes non-correlating body parts positional constraints. Whereas traditional methods use distance based generation schemes, context-dependent IMs are based on relevant joints instead of pre-programmed topologies. They yield more robust nets that increase spatial generalization and decrease computational load during runtime.

In essence, the key methodical contributions of this thesis are:

- A programming-by-demonstration framework based on human-human motion capture recordings
- An algorithm for efficient state estimation in human-robot interactions using low-dimensional spaces and trajectory segmentation.
- Context-dependent IMs for body-synchrony preserving and seamless robot responses
- An algorithm for automated constraint extraction based on joint correlations in human-human task demonstrations
- A one-shot learning scheme for simple and straightforward triadic human-robot handovers.

The proposed method was evaluated in three application scenarios. In a VR setting, virtual characters were adaptively animated in competitive and cooperative interactions. Here several punches and kicks were taught and for each the controlled agent responded with appropriate defense motions. In a casual interaction scenario high fives, dances and a clapping game were trained. Both settings required the virtual character to respond with adapted behaviors while still retaining body synchrony and spatial constraints.

The experiments revealed that the interaction model offers increased spatial generalization over traditional IM approaches. In particular, context-dependent IMs retain

joint correlations extracted from two-person motion capture recordings. This is a valuable advantage as it preserves the synchronicity found in natural human-human interactions. It also equips the virtual character with the ability to imitate the smoothness and ease of motion to which people are so accustomed. A result of the VR experiments is the insight that the topology of an IM plays a crucial role for the ability of the mesh to adapt to varying user motions. Traditional IM generation methods fail to preserve spatial details and require manual fine tuning of constraints for each interaction scenario. Context-dependent IMs on the other hand automatically attribute positional constraints, reducing training times significantly.

The conducted VR experiments showed the applicability and potential of interaction models in controlled virtual worlds. Given the encouraging results, the framework was subsequently deployed in human-robot settings, adding the physical aspects of the real-world. Among others, occlusions of motion capture markers and robot response timings are particular prominent examples that need to be taken into account. From a methodological point of view however, no alternations of the interaction model structure are required when transitioning from character animation to robot control. To evaluate the generalization capabilities of interaction models in robotics applications two examples were chosen. In the first application two complex assembly tasks were considered. Each task required fine-grained adaptation of the robot's motion so that objects could be jointly constructed with the human user. In the first assembly task a tube frame was assembled and the second example utilized Lego blocks to build a mock-up rocket. Both scenarios feature sequential subtasks and, as a result, the HMM-based interaction selection method was chosen. Based on the two-person motion capture recording, postures spaces as well as context-dependent IMs were automatically generated without user intervention. Despite different motion capture marker layouts and a reduced set of motion capture markers, the model generated reliable robot responses and the Lego rocket as well as the tube frame were jointly assembled with a human user. A result of the experiments is a quantitative analysis of the spatial generalization capabilities of interaction models. Context-dependent IMs offered $-25\,\text{cm}$ to $25\,\text{cm}$ of spatial adaptation which exceeds alternate IM approaches by a large margin. This showed that the proposed state estimation process and the novel IM generation scheme are also applicable in robotic applications involving close human-robot contact.

In a third application the interaction model approach was extended to a triadic setting. Whereas previous examples focused on body synchrony of two interacting partners, the triadic interaction model variant explicitly integrated an additional object. For that the object's motion was tracked during training and included into the posture space to enhance temporal alignment. At the same time, relevant points on the object's surface were considered during IM generation to improve response generation. Similar to dyadic settings, covariances among joints and object vertices provided important insight into interaction dynamics. The triadic interaction model variant thus models object motions and allows a robot to focus on the user's behavior *and* the object's rotations for more precise interactions. The inclusion of the object in to the IM generation scheme also results in an improved spatial adaptation capability

of the robot's gripper during a handover task. In the end, firm grasps and intuitive handovers take place since the robot's gripper closely matches the object's shape even for changing rotations.

The performance of the approach was confirmed by a user study. Participants were tasked to repeatedly handover a box-shaped object to the robot at different heights and orientations. This in turn prompted the robot to adapt its motion along the way. During the experiment all participants were also exposed to alternate control schemes which were recently reported in literature. A comparing view on all evaluated schemes revealed that triadic interaction models outperformed others by a large margin. Not only where twice as many interactions successful ($> 95\%$) but users reported further that they were more satisfied with their overall performance. This shows improved handover precisions with respect to accuracy and, according to the NASA TLX questionnaire, decreased cognitive load. Since users do not have to focus on an object's orientation during the handover, less time was required to pass the box yielding decreased interaction times. In the end, more intuitive and natural interactions emerged when the robot was controlled using interaction models.

8.2. Discussion and Future Research Directions

In the following, research ideas which emerge from theoretical and practical contributions of this thesis are discussed. They are designed to give an impetus to future research. However, they provide by no means an exhaustive view on the matter, but offer starting points for new considerations.

8.2.1. Application-oriented Research Directions

Most HRI research aims at algorithms and systems that can be deployed in real-world applications. For that, laboratory prototypes need to evolve into mature platforms that reliably offer robustness and user safety. To assess the readiness for deployment 10 Technology Readiness Levels (TRLs) have been formulated [135]. At the time of writing, interaction models fall into TRL 4. They reliably produced encouraging results in laboratory environments and a proof-of-concept has been developed. In order to mature into the next level, real-world studies and field tests are necessary. However, among other technical hurdles, a sophisticated tracking system is the first and foremost requirement before the next level can be achieved. Cluttered factory workplaces and unpredictable lighting conditions render available tracking technologies often unreliable and error-prone. Motion capture, which is fundamentally based on tracking hardware, suffers from additional line-of-sight problems and robots, tools as well as objects can potentially occlude the user's body. Given the resulting uncertainty about its posture, robust estimation algorithms are called for so that the user's behavior can be safely monitored. An additional benefit that results from advances in tracking technology is, that they not only improve the quality with which user motions are estimated, but also

the quality of training data. More reliable and accurate task demonstrations immediately improve the precision of a robot's response as it has a significant influence on its ability to detect and anticipate actions.

Possible research directions could also emerge from the interaction model's ability of reconstructing full body motions. In chapter 5 different VR applications for virtual characters were presented and in each, agent responses were seamlessly generated for the entire anthropomorphic skeletal structure. Given these encouraging results, it stands to reason if the framework is also applicable for full body control of humanoid robots. In comparison to their simulated counterparts, humanoids are inherently hard to balance and require sophisticated stability control mechanisms. Also, they are able to develop large momenta during motions which calls for additional safety measures. Nevertheless, interaction models are capable of generating full body controls for bipedal, bi-manual humanoids but their feasibility has yet to be confirmed with real-world experiments.

8.2.2. Theory-driven Research Directions

From Reactive to Proactive Behavior Generation

The proposed interaction model methodology implements a tight coupling of perceived human motions with robot responses. User poses are directly matched to task demonstrations and context-dependent Interaction Meshes generate appropriate robot controls. In the considered human-robot teams, it often suffices to rely on the humans' high-level ability of determining the next assembly step, focusing on the responsiveness of the robot's behavior instead. With that, the robot is able to respond instantly in a wide variety of situations. However, since the user steers the interaction with its own motion, it is his own burden to decide what action to take next. It can be argued that this task can become overwhelming if the number of steps increase.

Future research directions could evolve around proactive planning that is often found in human teams. Consider an assembly line where a robot works collectively with a user. Here, a proactive robotic co-worker would suggestively handover tools and objects that are required next in the task. In doing so, it would not only assist reactively but participate actively by suggesting subsequent steps. At the end, mixed initiative situations could emerge where robots proactively initiate interactions.

Applications in Shared Workplace Ergonomics Optimization

Another area of research could evolve around workplace ergonomics optimization. Despite a formal definition of ergonomics at workplaces, no real world implementations with robots have been presented so far [136]. The reason for that result from the complexity of the robot's control routines. When robot co-workers are deployed in industrial assembly lines, they naturally have to deal with ergonomic conditions of different human colleagues. Since each human worker has its own physical limitations and varying comfort zones, ergonomics parameters are hard to foresee during program-

ming. Besides the physical integrity of the human co-worker, other constraints such as body weight distribution or stance influence workplace ergonomics during runtime. In a typical handover scenario where a tool is passed from the robot to the human for example, a handover location might be optimal with respect to spatial relationships but requires the worker to lean forward unnaturally. This can lead to back pain and increases the risk of injury. A robot should consequently account for these situations and adapt its own behavior accordingly, so that the tool is passed ergonomically.

It has been shown throughout this thesis that IM topologies play a vital role for the success of a human-robot collaboration. Future work could extend and elaborate these findings by developing additional comfort constraints that not only account for spatial relationships but also ergonomics properties such as foot pressure and joint bend angles. In addition, alternate sensors could also be integrated. Pressure sensors in the worker's shoes for example could measure the weight distributions and comfort constraints could support the user's body balance as to avoid unnecessary physical load.

Life-long Learning and Adaptive Refinement of Interaction Demonstrations

Using an interaction model a robot spatially adapts recorded task demonstrations and there is consequently a clear distinction between the learning and reproduction phase. For robots to be placed in human society, expert knowledge might not be readily available for training and one can argue that there should be a continuum between the two phases. A new research direction could emerge around life-long refinement of interaction models. On top of the initial task demonstrations, additional knowledge about the interactions can be gained continuously during the course of multiple collaborations. Over time, existing interaction capabilities could potentially be refined, improving a robot's overall performance and interaction skills. Also, when no suitable interaction demonstration fits the current situation, alternate behaviors could be trained on-the-fly by non experts, building upon previous knowledge. For this to become reality an algorithmic foundation is needed that seamlessly generalizes interaction demonstrations. A potential implementation could harness the compact representations of motion trajectories in low-dimensional space to efficiently model probability distributions over task demonstrations. In doing so, the parameters could be learned and refined over the course of numerous interactions while at the same time adapting to the characteristics of the human interaction partner.

Multi-Human-Robot Teams and Applications of n-adic Interactions

Current HRI research for industrial applications focuses on diadic human-robot teaming. Yet, most factories feature multiple robots which, at this point, mainly work by themselves. These multi-robot environments that already exist open up new opportunities for human-robot teams where all participants are jointly working on a single collaborative task. Whereas current interaction learning applications essentially consider only one robot, future applications might rest upon symbiotic robot teams. In

assembly lines for example robot motions are typically computed individually and work areas are not shared with humans. In multi-robot environments one robotic arm could pass a tool to the user whereas a second robot assists during assembly by holding the to be assembled object. A third robot could meanwhile fetch the next object on a conveyor belt proactively, yielding a seamless and smooth interaction between all interactants. However, in order to plan the robots' trajectories, external influences such as belt timings, user motions as well as its working pace have to be taken into account. As a result the underlying controls require continuous planning and optimization for groups of interactants as specially when they are in close proximity.

The interaction model implementation in chapter 7 already showed the applicability of the proposed learning framework in a triadic setting, involving three interactants. Here, the model inherently captured the object that mediated the interaction, i.e. the handover as well as both human interaction partners. Based on these results it stands to reason if the framework is also applicable in n-adic settings involving multiple robots that jointly cooperate with a human.

8.3. Concluding Remarks

Future societies will thrive on robots that simplify peoples' lives in all of its facets. A starting point for this thesis was that pre-programming each possible situation is not practical and sophisticated programming techniques were called for. Turning towards nature for help, imitation learning gained increasing popularity in recent research [11, 137]. It allows robots to gradually increase their skill sets by imitating a human teacher [11]. Despite the recent results, existing methods are mainly restricted to a single trainer, resulting in robot behaviors that require them to be constrained in caged environments. For robots to be included in human day-to-day activities however, interaction skills are needed that promote teaming and collaboration with others.

Building upon the concept of imitation, the presented interaction learning methodology broadens the perspective of programming-by-demonstration. It provides a framework for learning interaction skills, enabling robots to actively participate in interactions with people. An important insight is that by using human-human demonstrations knowledge about collaborative tasks can be efficiently imparted while at the same time yielding intuitive and human-readable responses. Experiments in VR as well as real-world HRI settings demonstrate the feasibility of the methodology and its benefit for robot learning. While the results are encouraging, there are still open research questions and technical hurdles to be tackled before learning robots can be deployed in human society. Nevertheless, this thesis showed that by observing two humans demonstrating a collaborative task valuable information about interaction dynamics and spatiotemporal characteristics can be extracted and used for learning interaction skills.

A. Appendix

A.1. Two-Person Motion Capture Data

Table A.1 shows a motion capture frame of a dyadic HRI for a single time step. Each row represents a tracked joint $\mathbf{p}_t^i$ with its Cartesian coordinate $\mathbf{p}_t^{i,x:z}$ and rotation matrix (row wise $m_1, \ldots, m_9$). The recordings for triadic HRI have the same layout for both interactants but feature additional rows for points on the object's surface.

Tab. A.1.: A motion capture frame of a dyadic HRI

Joint	$\mathbf{p}_{t=1}^{i,x}$	$\mathbf{p}_{t=1}^{i,y}$	$\mathbf{p}_{t=1}^{i,z}$	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	m_9
$\mathbf{p}_{t=1}^{i=1}$	541.053	714.656	1744.91	0.007	-0.999	0.030	0.977	0.000	-0.209	0.209	0.031	0.977
$\mathbf{p}_{t=1}^{i=2}$	-503.611	591.681	940.046	-0.396	-0.917	-0.025	-0.911	0.390	0.133	-0.112	0.075	-0.990
$\mathbf{p}_{t=1}^{i=3}$	-424.369	449.784	1503.18	-0.261	0.961	0.079	0.807	0.263	-0.527	-0.528	-0.073	-0.846
$\mathbf{p}_{t=1}^{i=4}$	753.82	822.146	133.857	-0.428	-0.205	0.880	-0.356	-0.856	-0.373	0.830	-0.473	0.293
$\mathbf{p}_{t=1}^{i=5}$	405.666	456.796	1142.93	-0.646	0.390	0.655	0.143	-0.780	0.607	0.749	0.487	0.448
$\mathbf{p}_{t=1}^{i=6}$	463.631	851.441	1095.9	-0.517	-0.207	0.830	-0.355	0.934	0.012	-0.778	-0.288	-0.557
$\mathbf{p}_{t=1}^{i=7}$	-401.239	728.563	110.637	-0.560	0.282	0.778	-0.816	-0.344	-0.462	0.137	-0.895	0.424
$\mathbf{p}_{t=1}^{i=8}$	-271.363	828.821	1220.59	0.183	-0.866	-0.463	-0.472	0.335	-0.814	0.862	0.368	-0.347
$\mathbf{p}_{t=1}^{i=9}$	-554.015	236.418	1260.95	-0.698	-0.364	-0.615	-0.566	-0.241	0.787	-0.435	0.899	-0.037
$\mathbf{p}_{t=1}^{i=10}$	-333.697	615.083	1514.57	0.369	-0.912	-0.173	-0.843	-0.407	0.348	-0.389	0.017	-0.921
$\mathbf{p}_{t=1}^{i=11}$	505.67	581.214	112.215	0.082	0.280	-0.956	-0.049	0.959	0.276	0.995	0.024	0.092
$\mathbf{p}_{t=1}^{i=12}$	501.305	690.086	939.074	-0.135	-0.966	0.219	0.989	-0.121	0.078	-0.048	0.228	0.972
$\mathbf{p}_{t=1}^{i=13}$	-590.77	628.121	1748.1	-0.053	0.989	0.134	-0.949	-0.092	0.300	0.309	-0.111	0.944
$\mathbf{p}_{t=1}^{i=14}$	368.059	828.619	1431.8	-0.249	0.783	-0.568	-0.936	-0.345	-0.064	-0.246	0.516	0.820
$\mathbf{p}_{t=1}^{i=15}$	312.587	569.132	1437.04	0.501	-0.864	0.038	0.810	0.484	0.328	-0.302	-0.133	0.943
$\mathbf{p}_{t=1}^{i=16}$	315.584	549.132	1417.06	0.412	-0.924	0.128	0.841	0.459	0.299	-0.302	-0.101	0.916

Bibliography

[1] David Vogt, Erik Berger, Heni Ben Amor, and Bernhard Jung. Learning Two-Person Interaction Models for Responsive Virtual Characters and Humanoid Robots. In *Fachgruppe Virtuelle und Augmentierte Realität der Gesellschaft für Informatik*, 2012.

[2] David Vogt, Erik Berger, Heni Ben Amor, and Bernhard Jung. A Task-Space Two-Person Interaction Model for Human-Robot Interaction. In *Fachgruppe Virtuelle und Augmentierte Realität der Gesellschaft für Informatik*, 2013.

[3] David Vogt, Heni Ben Amor, Erik Berger, and Bernhard Jung. Learning Two-Person Interaction Models for Responsive Synthetic Humanoids. *Journal of Virtual Reality and Broadcasting*, 11, 2014.

[4] David Vogt, Steve Grehl, Erik Berger, Heni Ben Amor, and Bernhard Jung. A Data-Driven Method for Real-Time Character Animation in Human-Agent Interaction. In Timothy Bickmore, Stacy Marsella, and Candace Sidner, editors, *Intelligent Virtual Agents SE - 57*, volume 8637 of *Lecture Notes in Computer Science*, pages 463–476. Springer International Publishing, 2014.

[5] David Vogt, Ben Lorenz, Steve Grehl, and Bernhard Jung. Behavior generation for interactive virtual humans using context-dependent interaction meshes and automated constraint extraction. *Computer Animation and Virtual Worlds*, 26 (3-4):227–235, may 2015.

[6] David Vogt, Simon Stepputtis, Richard Weinhold, Bernhard Jung, and Heni Ben Amor. Learning human-robot interactions from human-human demonstrations (with applications in Lego rocket assembly). In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 142–143. IEEE, nov 2016.

[7] David Vogt, Simon Stepputtis, Steve Grehl, Bernhard Jung, and Ben Amor Heni. A System for Learning Continuous Human-Robot Interactions from Human-Human Demonstrations. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2882–2889, Singapore, 2017. IEEE.

[8] David Vogt, Simon Stepputtis, Bernhard Jung, and Ben Amor Heni. One-Shot Learning of Human-Robot Handovers with Triadic Interaction Meshes. *Springer Special Issue: Learning for Human-Robot Collaboration*, 2018.

[9] Chuck Palahniuk. *Lullaby*. Doubleday, 2002.

[10] Guy Hoffman. Evaluating Fluency in Human-Robot Collaboration. *Robotics: Science and Systems, Workshop on Human Robot Collaboration*, 381:1–8, 2013.

[11] Heni Ben Amor. *Imitation Learning of Motor Skills for Synthetic Humanoids*. PhD thesis, Technische Universität Bergakademie Freiberg, 2010.

[12] Ivana Konvalinka, Peter Vuust, Andreas Roepstorff, and Chris D Frith. Follow you, follow me: Continuous mutual prediction and adaptation in joint tapping. *The Quarterly Journal of Experimental Psychology*, 63(11):2220–2230, 2010.

[13] Richard C. Schmidt and Beth O'Brien. Evaluating the Dynamics of Unintended Interpersonal Coordination. *Ecological Psychology*, 9(3):189–206, 1997.

[14] Terrence Fong, Illah Nourbakhsh, and Kerstin Dautenhahn. A survey of socially interactive robots. *Robotics and Autonomous Systems*, 42(3-4):143–166, 2003.

[15] Volker Klingspor, John Demiris, and Michael Kaiser. Human Robot Communication and Machine Learning. *Applied Artifical Intelligence*, 11(7):719–746, 1997.

[16] Clare Press. Action observation and robotic agents: Learning and anthropomorphism. *Neuroscience & Biobehavioral Reviews*, 35(6):1410–1418, may 2011.

[17] Emrah Akin Sisbot, Luis F. Marin, and Rachid Alami. Spatial reasoning for human robot interaction. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2281–2287. IEEE, oct 2007.

[18] Anca Dragan, Shira Bauman, Jodi Forlizzi, and Siddhartha Srinivasa. Effects of Robot Motion on Human-Robot Collaboration. *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, 1:51–58, 2015.

[19] John Demiris and Gillian Hayes. Do robots ape. In *Proceedings of the AAAI Fall Symposium*, pages 28–30, 1997.

[20] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE Multimedia*, 19 (2):4–10, 2012.

[21] Organic Motion. OpenStage 2, 2016. URL `http://www.organicmotion.com/`.

[22] Vicon. Bonita - A Optical Motion Capture System, 2017. URL `http://www.vicon.com/`.

[23] K Dautenhahn and C L Nehaniv. *The Agent-Based Perspective on Imitation*, pages 1–40. MIT Press, 2002.

[24] John Demiris and Gillian Hayes. Active Imitation. *AISB'99 Symposium on "Imitation in Animals and Artifacts"*, 1998.

[25] David W. Johnson and Roger T. Johnson. New Developments in Social Interdependence Theory. *Genetic, Social, and General Psychology Monographs*, 131(4): 285–358, nov 2005.

[26] Matthew D. Lieberman. Social Cognitive Neuroscience: A Review of Core Processes. *Annual Review of Psychology*, 58(1):259–289, 2007.

[27] Natalie Sebanz, Harold Bekkering, and Günther Knoblich. Joint action: Bodies and minds moving together. *Trends in Cognitive Sciences*, 10(2):70–76, 2006.

[28] Tao Liu and Matthew Pelowski. Clarifying the interaction types in two-person neuroscience research. *Frontiers in Human Neuroscience*, 8, apr 2014.

[29] Lucas Kovar, Michael Gleicher, and Frédéric Pighin. Motion graphs. *ACM Transactions on Graphics*, 21(3), jul 2002.

[30] Victor Brian Zordan, Anna Majkowska, Bill Chiu, and Matthew Fast. Dynamic response for motion capture animation. *ACM Transactions on Graphics*, 24(3): 697, jul 2005.

[31] Rachel Heck and Michael Gleicher. Parametric motion graphs. *Proceedings of the 2007 symposium on Interactive 3D graphics and games I3D 07*, pages:129, 2007.

[32] T. Pejsa and I. S. Pandzic. State of the art in example-based motion synthesis for virtual characters in interactive applications. *Computer Graphics Forum*, 29 (1):202–226, 2010.

[33] Rami Ali Al-Asqhar, Taku Komura, and Myung Geol Choi. Relationship descriptors for interactive motion adaptation. *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation - SCA '13*, page 45, 2013.

[34] Mathieu Barnachon, Saïda Bouakaz, Boubakeur Boufama, and Erwan Guillou. Ongoing human action recognition with motion capture. *Pattern Recognition*, 47 (1):238–247, jan 2014.

[35] Carlo Camporesi, Yazhou Huang, and Marcelo Kallmann. Interactive Motion Modeling and Parameterization by Direct Demonstration. In *Intelligent Virtual Agents*, pages 77–90. Springer Berlin Heidelberg, 2010.

[36] Yang Xiao, Junsong Yuan, and Daniel Thalmann. Human-virtual human interaction by upper body gesture understanding. In *Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology - VRST '13*, page 133, New York, New York, USA, 2013. ACM Press.

[37] Kilian Q Weinberger and Lawrence K Saul. Distance Metric Learning for Large Margin Nearest Neighbor Classification. *Journal of Machine Learning Research*, 10:207–244, 2009.

[38] Nick Taubert, Martin Löffler, Nicolas Ludolph, Andrea Christensen, Dominik Endres, and Martin a. Giese. A virtual reality setup for controllable, stylized real-time interactions between humans and avatars with sparse Gaussian process dynamical models. *Proceedings of the ACM Symposium on Applied Perception - SAP '13*, page 41, 2013.

[39] Edmond S. L. Ho, Taku Komura, and Chiew-Lan Tai. Spatial relationship preserving character motion adaptation. *ACM Transactions on Graphics*, 29(4):1, jul 2010.

[40] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian Surface Editing. *Eurographics Symposium on Geometry Processing*, pages 175–184, 2004.

[41] Edmond S. L. Ho, Jacky C. P. Chan, Taku Komura, and Howard Leung. Interactive partner control in close interactions for real-time applications. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 9 (3):1–19, jun 2013.

[42] A.J. Ijspeert, J. Nakanishi, and S. Schaal. Trajectory formation for imitation with nonlinear dynamical systems. In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, volume 2, pages 752–757. IEEE, 2001.

[43] Miguel Prada, Anthony Remazeilles, Ansgar Koene, and Satoshi Endo. Dynamic Movement Primitives for Human-Robot interaction: Comparison with human behavioral observation. *IEEE International Conference on Intelligent Robots and Systems*, pages 1168–1175, 2013.

[44] Heni Ben Amor, Gerhard Neumann, Sanket Kamthe, Oliver Kroemer, and Jan Peters. Interaction primitives for human-robot cooperation tasks. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2831–2837. IEEE, may 2014.

[45] Marco Ewerton, Gerhard Neumann, Rudolf Lioutikov, Heni Ben Amor, Jan Peters, and Guilherme Maeda. Learning multiple collaborative tasks with a mixture of Interaction Primitives. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1535–1542. IEEE, may 2015.

[46] Guilherme Maeda, Aayush Maloo, Marco Ewerton, Rudolf Lioutikov, and Jan Peters. Anticipative Interaction Primitives for Human-Robot Collaboration. In *2016 AAAI Fall Symposium Series: Shared Autonomy in Research and Practice*, pages 325–330, 2016.

[47] L Rozo, D Bruno, S Calinon, and D G Caldwell. Learning Optimal Controllers in Human-robot Cooperative Transportation Tasks with Position and Force Constraints. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1024–1030, Hamburg, Germany, 2015.

[48] Yixing Gao, Hyung Jin Chang, and Yiannis Demiris. User Modelling for Personalised Dressing Assistance by Humanoid Robots. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1840–1845, Hamburg, Germany, 2015.

[49] Yasufumi Tanaka, Jun Kinugawa, Yusuke Sugahara, and Kazuhiro Kosuge. Motion planning with worker's trajectory prediction for assembly task partner robot. *IEEE International Conference on Intelligent Robots and Systems*, pages 1525–1532, 2012.

[50] Paul Evrard, Elena Gribovskaya, Sylvain Calinon, Aude Billard, and Abderrahmane Kheddar. Teaching physical collaborative tasks: object-lifting case study with a humanoid. In *2009 9th IEEE-RAS International Conference on Humanoid Robots*, pages 399–404. IEEE, dec 2009.

[51] Paul Evrard and Abderrahmane Kheddar. Homotopy switching model for dyad haptic interaction in physical collaborative tasks. In *World Haptics 2009 - Third Joint EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 45–50. IEEE, 2009.

[52] Tomas Kulvicius, Martin Biehl, Mohamad Javad Aein, Minija Tamosiunaite, and Florentin Wörgötter. Interaction learning for dynamic movement primitives used in cooperative robotic tasks. *Robotics and Autonomous Systems*, 61(12):1450–1459, dec 2013.

[53] D. Lee, C. Ott, and Y. Nakamura. Mimetic Communication Model with Compliant Physical Contact in Human–Humanoid Interaction. *The International Journal of Robotics Research*, 29(13):1684–1704, nov 2010.

[54] Heni Ben Amor, David Vogt, Marco Ewerton, Erik Berger, Bernhard Jung, and Jan Peters. Learning responsive robot behavior by imitation. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3257–3264. IEEE, nov 2013.

[55] Jose Ramon Medina, Martin Lawitzky, Alexander Mortl, Dongheui Lee, and Sandra Hirche. An experience-driven robotic assistant acquiring human knowledge to improve haptic cooperation. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2416–2422. IEEE, sep 2011.

[56] Elena Corina Grigore, Kerstin Eder, Anthony G. Pipe, Chris Melhuish, and Ute Leonards. Joint action understanding improves robot-to-human object handover.

IEEE International Conference on Intelligent Robots and Systems, pages 4622–4629, 2013.

[57] Vladimir Ivan, Dmitry Zarubin, Marc Toussaint, Taku Komura, and Sethu Vijayakumar. Topology-based representations for motion planning and generalization in dynamic environments with interactions. *The International Journal of Robotics Research*, 32(9-10):1151–1163, 2013.

[58] Edmond S. L. Ho and Hubert P. H. Shum. Motion adaptation for humanoid robots in constrained environments. *2013 IEEE International Conference on Robotics and Automation*, pages 3813–3818, may 2013.

[59] Yiming Yang, Vladimir Ivan, and Sethu Vijayakumar. Real-time motion adaptation using relative distance space representation. In *2015 International Conference on Advanced Robotics (ICAR)*, number 17, pages 21–27. IEEE, jul 2015.

[60] Philine Donner, Florian Wirnshofer, and Martin Buss. Controller synthesis for human-robot cooperative swinging of rigid objects based on human-human experiments. *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, pages 586–592, 2014.

[61] Stefanos Nikolaidis and Julie Shah. Human-robot cross-training: Computational formulation, modeling and evaluation of a human team training strategy. *ACM/IEEE International Conference on Human-Robot Interaction*, pages 33–40, 2013.

[62] Vaibhav V. Unhelkar, Ho Chit Siu, and Julie a. Shah. Comparative performance of human and mobile robotic assistants in collaborative fetch-and-deliver tasks. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pages 82–89, New York, New York, USA, 2014. ACM Press.

[63] Kenji Sakita, Kentaro Kawamura, and Koichi Ogawara. Flexible cooperation between human and robot by interpreting human intention from gaze information. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 846–851. IEEE, 2004.

[64] Muhammad Awais and Dominik Henrich. Proactive premature-intention estimation for intuitive human robot collaboration. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4098–4103, 2012.

[65] Maya Cakmak, Siddhartha S. Srinivasa, Min Kyung Lee, Sara Kiesler, and Jodi Forlizzi. Using spatial and temporal contrast for fluent robot-human hand-overs. *Proceedings of the 6th international conference on Human-robot interaction - HRI '11*, page 489, 2011.

[66] Crystal Chao and Andrea L Thomaz. Timing in Multimodal Turn-Taking Interactions: Control and Analysis Using Timed Petri Nets. *Journal of Human-Robot Interaction*, 1(1):1, 2011.

[67] Kelsey Hawkins, Nam Vo, Shray Bansal, and Aaron Bobick. Probabilistic Human Action Prediction and Wait-sensitive Planning for Responsive Human-robot Collaboration. *International Conference on Humanoid Robots*, pages 499 – 506, 2013.

[68] Nicola Maria Ceriani, Andrea Maria Zanchettin, Paolo Rocco, Andreas Stolt, and Anders Robertsson. A constraint-based strategy for task-consistent safe human-robot interaction. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4630–4635. IEEE, nov 2013.

[69] B Lacevic and P Rocco. Kinetostatic danger field - a novel safety assessment for human-robot interaction. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2169–2174. IEEE, oct 2010.

[70] Guilherme J. Maeda, Gerhard Neumann, Marco Ewerton, Rudolf Lioutikov, Oliver Kroemer, and Jan Peters. Probabilistic movement primitives for coordination of multiple human-robot collaborative tasks. *Autonomous Robots*, pages 1–20, 2016.

[71] Elena Gribovskaya, Abderrahmane Kheddar, and Aude Billard. Motion learning and adaptive impedance for robot control during physical interaction with humans. In *2011 IEEE International Conference on Robotics and Automation*, pages 4326–4332. IEEE, may 2011.

[72] Hiraki Goto, Jun Miura, and Junichi Sugiyama. Human-Robot Collaborative Assembly by On-line Human Action Recognition Based on an FSM Task Model. *International Conference on Human-Robot Interaction - Workshop on Collaborative Manipulation*, pages 1–6, 2013.

[73] ART Advanced Realtime Tracking. ARTRACK5: Marker-based Optical Motion Tracking, 2016. URL `http://www.ar-tracking.com/`.

[74] Microsoft. Kinect One Depth Sensor, 2015. URL `https://developer.microsoft.com/en-us/windows/kinect`.

[75] Organic Motion. BioStage: Markerless motion capture, 2016. URL `http://www.organicmotion.com/`.

[76] Bruno Siciliano and Oussama Khatib, editors. *Springer Handbook of Robotics*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[77] Peter Corke. *Robotics, Vision and Control*, volume 73 of *Springer Tracts in Advanced Robotics*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[78] Dumebi Okwechime, Eng-Jon Ong, and Richard Bowden. Real-time motion control using pose space probability density estimation. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 2056–2063. IEEE, sep 2009.

[79] Sebastian Bitzer, Stefan Klanke, and Sethu Vijayakumar. Does Dimensionality Reduction Improve the Quality of Motion Interpolation? *Proc. 17th European Symposium on Artificial Neural Networks (ESANN '09)*, 2009.

[80] Rawichote Chalodhorn and Rajesh P N Rao. Using eigenposes for lossless periodic human motion imitation. *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, pages 2502–2509, 2009.

[81] Mario Botsch and Olga Sorkine. On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics*, 14:213–230, 2007.

[82] Yaron Lipman, Olga Sorkine, Daniel Cohen-Or, David Levin, Christian Rössl, and Hans Peter Seidel. Differential coordinates for interactive mesh editing. *Proceedings - Shape Modeling International SMI 2004*, pages 181–190, 2004.

[83] D Cohen-Or and O Sorkine. Encoding meshes in differential coordinates. *SCCG06) Proceedings of the 22nd Spring ...*, 2006.

[84] Yaron Lipman, Olga Sorkine-Hornung, Marc Alexa, Daniel Cohen-Or, David Levin, Christian Rössl, and Hans-Peter Seidel. Laplacian Framework for Interactive Mesh Editing. *International Journal of Shape Modeling*, 11(01):43–62, jun 2005.

[85] Yanlin Weng, Weiwei Xu, Yanchen Wu, Kun Zhou, and Baining Guo. 2D shape deformation using nonlinear least squares optimization. *The Visual Computer*, 22(9):653–660, 2006.

[86] Anderson Maciel and Suvranu De. Spatiotemporal coupling with the 3D+t motion Laplacian. *Computer Animation and Virtual Worlds*, 24(May):419–428, 2013.

[87] Lawrence R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, 1989.

[88] Gernot A. Fink. *Markov Models for Pattern Recognition.* Springer London, London, 2014.

[89] Lawrence R Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In Alex Waibel and Kai-Fu Lee, editors, *Readings in Speech Recognition*, chapter A Tutorial, pages 267–296. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.

[90] Richard Ernest Bellman. *Dynamic Programming.* Dover Publications, Incorporated, 2003.

[91] Kris Demuynck, Jacques Duchateau, Dirk Van Compernolle, and Patrick Wambacq. Improved Feature Decorrelation for Hmm-Based Speech Recognition. In *International conference on spoken language processing*, pages 2907–2910, Sydney, 1998.

[92] Peter Viszlay, Matúš Pleva, and Jozef Juhár. Comparison of Different Feature Decorrelation Techniques used in HMM-based Acoustic Models. *7th International workshop on digital technologies, circuits, system and signal processing*, Digital Te:1–4, 2010.

[93] Hyekyung Lee and Seungjin Choi. PCA+HMM+SVM for EEG pattern classification. In *Seventh International Symposium on Signal Processing and Its Applications, 2003. Proceedings.*, pages 541–544 vol.1. IEEE, 2003.

[94] Abd Manan Ahmad, Abdullah Bade, and Luqman Al-Hakim Zainal Abidin. Using Principal Component Analysis and Hidden Markov Model for Hand Recognition Systems. In *2009 International Conference on Information and Multimedia Technology*, pages 323–326. IEEE, 2009.

[95] SamerKais Jameel. Face Recognition System Using PCA and DCT in HMM. *IJARCCE*, pages 13–18, jan 2015.

[96] Julien Bloit and Xavier Rodet. Short-time Viterbi for online HMM decoding: Evaluation on a real-time phone recognition task. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2121–2124. IEEE, mar 2008.

[97] Thierry Ravet, Joëlle Tilmanne, and Nicolas D'Alessandro. Hidden Markov Model Based Real-Time Motion Recognition and Following. In *Proceedings of the 2014 International Workshop on Movement and Computing - MOCO '14*, pages 82–87, New York, New York, USA, 2014. ACM Press.

[98] Ulman Lindenberger, Shu-Chen Li, Walter Gruber, and Viktor Müller. Brains swinging in concert: cortical phase synchronization while playing guitar. *BMC Neuroscience*, 10(1):22, 2009.

[99] Nadine Pecenka and Peter E. Keller. The role of temporal prediction abilities in interpersonal sensorimotor synchronization. *Experimental Brain Research*, 211 (3-4):505–515, jun 2011.

[100] Ivana Konvalinka and Andreas Roepstorff. The two-brain approach: how can mutually interacting brains teach us something about social interaction? *Frontiers in Human Neuroscience*, 6, 2012.

[101] Tetsunari Inamura, Iwaki Toshima, Hiroaki Tanie, and Yoshihiko Nakamura. Embodied Symbol Emergence Based on Mimesis Theory. *The International Journal of Robotics Research*, 23(4):363–377, apr 2004.

[102] D. Kulic and Y. Nakamura. Scaffolding on-line segmentation of full body human motion patterns. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2860–2866. IEEE, sep 2008.

[103] Leonel Rozo, Joãn Silvèrio, Sylvain Calinon, and Darwin G Caldwell. Exploiting Interaction Dynamics for Learning Collaborative Robot Behaviors. In *International Joint Conference on Artificial Intelligence (IJCAI), Workshop on Interactive Machine Learning*, pages 1–7, 2016.

[104] Mahvash Jebeli, Alireza Bilesan, and Ahmadreza Arshi. A study on validating KinectV2 in comparison of Vicon system as a motion capture system for using in Health Engineering in industry. *Nonlinear Engineering*, 0(0):1–5, jan 2017.

[105] Reza N. Jazar. *Theory of Applied Robotics*. Springer US, Boston, MA, 2010.

[106] Fi Bashir, A.A. Khokhar, and D. Schonfeld. Object Trajectory-Based Activity Classification and Recognition Using Hidden Markov Models. *IEEE Transactions on Image Processing*, 16(7):1912–1919, jul 2007.

[107] Baris Akgun, Maya Cakmak, Karl Jiang, and Andrea L. Thomaz. Keyframe-based Learning from Demonstration. *International Journal of Social Robotics*, 4 (4):343–355, nov 2012.

[108] Baris Akgun, Maya Cakmak, Jae Yoo, and Andrea Thomaz. Trajectories and Keyframes for Kinesthetic Teaching: A Human-Robot Interaction Perspective. *International Conference on Human-Robot Interaction*, pages 391—-398, 2012.

[109] Matej Kristan, Aleš Leonardis, and Danijel Skočaj. Multivariate online kernel density estimation with Gaussian kernels. *Pattern Recognition*, 44(10-11):2630–2642, oct 2011.

[110] Donald J Berndt and James Clifford. Using Dynamic Time Warping to Find Patterns in Time Series. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, AAAIWS'94, pages 359–370. AAAI Press, 1994.

[111] Chotirat Ann Ratanamahatana and Eamonn Keogh. Everything you know about dynamic time warping is wrong. *Third Workshop on Mining Temporal and Sequential Data*, pages 22–25, 2004.

[112] Liqun Deng, Howard Leung, Naijie Gu, and Yang Yang. Real-time mocap dance recognition for an interactive dancing game. *Computer Animation and Virtual Worlds*, 22(2-3):229–237, apr 2011.

[113] Jongmin Kim, Yeongho Seol, and Jehee Lee. Human motion reconstruction from sparse 3D motion sensors using kernel CCA-based regression. *Computer Animation and Virtual Worlds*, 24(6):565–576, nov 2013.

[114] Xue Weimin and Xia Wenhong. E-Learning Assistant System Based on Virtual Human Interaction Technology. In Yong Shi, GeertDick Albada, Jack Dongarra, and Peter M.A. Sloot, editors, *Computational Science âĂŞ ICCS 2007*, pages 551–554. Springer Berlin Heidelberg, 2007.

[115] Jeffrey Osterlund and Brad Lawrence. Virtual reality: Avatars in human spaceflight training. *Acta Astronautica*, 71:139–150, feb 2012.

[116] Veena Chattaraman, Wi-Suk Kwon, and Juan E. Gilbert. Virtual agents in retail web sites: Benefits of simulated social interaction for older users. *Computers in Human Behavior*, 28(6):2055–2066, nov 2012.

[117] Edmond S. L. Ho, He Wang, and Taku Komura. A Multi-resolution Approach for Adapting Close Character Interaction. In *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology*, VRST '14, pages 97–106, New York, NY, USA, 2014. ACM.

[118] Jeremy N. Bailenson and James J. Blascovich. Avatars. In W. S. Bainbridge, editor, *Berkshire encyclopedia of human-computer interaction*, pages 64–68. Berkshire Publishing Group, 2004.

[119] Thomas Waltemate, Irene Senna, Felix Hülsmann, Marieke Rohde, Stefan Kopp, Marc Ernst, and Mario Botsch. The impact of latency on perceptual judgments and motor performance in closed-loop interaction in virtual reality. *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology - VRST '16*, pages 27–35, 2016.

[120] Edmond S L Ho, Taku Komura, Subramanian Ramamoorthy, and Sethu Vijayakumar. Controlling humanoid robots in topology coordinates. In *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, pages 178–182, 2010.

[121] D. Lee, C. Ott, and Y. Nakamura. Mimetic Communication Model with Compliant Physical Contact in Human–Humanoid Interaction. *The International Journal of Robotics Research*, 29(13):1684–1704, nov 2010.

[122] Felix Duvallet, Murali Karnam, and Aude Billard. A Human-Inspired Controller for Fluid Human-Robot Handovers. In *Humanoids 2016 - 16th IEEE-RAS International Conference on Humanoid Robots*, pages 324–331, 2016.

[123] Henny Admoni, Anca Dragan, Siddhartha Srinivasa, and Brian Scassellati. Deliberate Delays During Robot-to-Human Handovers Improve Compliance With Gaze Communication. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pages 49–56, Bielefeld, Germany, 2014.

[124] Kyle Wayne Strabala, Min Kyung Lee, Anca Diana Dragan, Jodi Lee Forlizzi, Siddhartha Srinivasa, Maya Cakmak, and Vincenzo Micelli. Towards Seamless

Human-Robot Handovers. *Journal of Human-Robot Interaction*, 2(1):112–132, mar 2013.

[125] Minhua Zheng, AJung Moon, Elizabeth A. Croft, and Max Q.-H. Meng. Impacts of Robot Head Gaze on Robot-to-Human Handovers. *International Journal of Social Robotics*, 7(5):783–798, nov 2015.

[126] Anca D. Dragan, Kenton C T Lee, and Siddhartha S. Srinivasa. Legibility and predictability of robot motion. *ACM/IEEE International Conference on Human-Robot Interaction*, 1:301–308, 2013.

[127] Frédéric Dehais, Emrah Akin Sisbot, Rachid Alami, and Mickaël Causse. Physiological and subjective evaluation of a humanâĂŞrobot object hand-over task. *Applied Ergonomics*, 42(6):785–791, nov 2011.

[128] Jim Mainprice, Mamoun Gharbi, Thierry Simeon, and Rachid Alami. Sharing effort in planning human-robot handover tasks. In *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, pages 764–770. IEEE, sep 2012.

[129] Ana Huaman Quispe, Heni Ben Amor, and Mike Stilman. Handover planning for every occasion. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 431–436. IEEE, nov 2014.

[130] Andras Kupcsik, David Hsu, and Wee Sun Lee. Learning Dynamic Robot-to-Human Object Handover from Human Feedback. *Proceedings of the International Symposium on Robotics Research*, pages 1–11, 2015.

[131] Marco Ewerton, Gerhard Neumann, Rudolf Lioutikov, Heni Ben Amor, Jan Peters, and Guilherme Maeda. Learning multiple collaborative tasks with a mixture of Interaction Primitives. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1535–1542. IEEE, may 2015.

[132] Zoltán Bankó and János Abonyi. Correlation based dynamic time warping of multivariate time series. *Expert Systems with Applications*, 39(17):12814–12823, 2012.

[133] Chien-ming Huang, Maya Cakmak, and Bilge Mutlu. Adaptive Coordination Strategies for Human-Robot Handovers. In *Robotics: Science and Systems XI*. Robotics: Science and Systems Foundation, jul 2015.

[134] Maya Cakmak, Siddhartha S Srinivasa, Min Kyung Lee, Jodi Forlizzi, and Sara Kiesler. Human preferences for robot-human hand-over configurations. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1986–1993. IEEE, sep 2011.

[135] John C Mankins. Technology Readiness Levels. *White Paper April*, 6308(July): 5, 1995.

[136] Marco Faber, Jennifer Bützler, and Christopher M. Schlick. Human-robot Cooperation in Future Production Systems: Analysis of Requirements for Designing an Ergonomic Work System. *Procedia Manufacturing*, 3(Ahfe):510–517, 2015.

[137] Bernhard Jung, Heni Ben Amor, Guido Heumer, and Matthias Weber. From motion capture to action capture. In *Proceedings of the ACM symposium on Virtual reality software and technology - VRST '06*, page 145, New York, New York, USA, 2006. ACM Press.